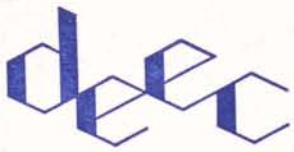


DEPARTAMENTO DE

ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

PESQUISA E IDENTIFICAÇÃO DEFEITOS EM MADEIRA POR VISÃO COMPUTACIONAL

FACULDADE DE ENGENHARIA

UNIVERSIDADE DO PORTO

Rua dos Bragas, 4099 Porto Codex – PORTUGAL

R-8: V. 769
2ª Ex.

FACULDADE DE ENGENHARIA DA
UNIVERSIDADE DO PORTO
DEPARTAMENTO DE ENGENHARIA
ELECTROTÉCNICA E COMPUTADORES

DISSERTAÇÃO DE MESTRADO

PESQUISA E IDENTIFICAÇÃO DEFEITOS EM MADEIRA POR VISÃO COMPUTACIONAL

UNIVERSIDADE DO PORTO
Faculdade de Engenharia
BIBLIOTECA M
Nº 20536-0
CDU 681.3(043)
Data 16 . 1 1994
M.M.34594

ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES
PERFIL DE INFORMÁTICA INDUSTRIAL

RUI ANTÓNIO QUADROS BEBIANO
Março 1994

043M
B352p
ex. 2

PESQUISA E IDENTIFICAÇÃO DE DEFEITOS POR VISÃO COMPUTACIONAL EM MADEIRA DE PINHO PINUS SILVESTRIS

Tese realizada sob Orientação dos professores:

Professor Doutor José Manuel Mendonça[♦]
Co-orientada pelo Doutor José Campos Neves

♦ *Professor Auxiliar do Departamento de Engenharia Electrotécnica e de Computadores, da Faculdade de Engenharia da Universidade do Porto*

À minha mulher Filomena
e à minha filha Marta Carolina

Biografia Sumária

O arguente nasceu em 1955 na aldeia de Alhadas, concelho da Figueira da Foz. Frequentou e completou os estudos do ensino secundário no Liceu Nacional da cidade. Fez parte de uma das suas turmas piloto e foi dispensado do exame de admissão à Universidade.

Licenciou-se em 1977 em Engenharia Electrotécnica pela Faculdade de Ciências e Tecnologia da Universidade de Coimbra (FCTUC), e exerceu funções docentes durante doze anos como professor efectivo do Ensino Secundário. Durante esse período exerceu também actividade liberal como projectista de instalações eléctricas e telefónicas.

Licenciou-se posteriormente em História pela Faculdade de Letras da U.Coimbra e em 1987 concluiu uma pós graduação em Estudos Europeus vertente de Economia, na Faculdade de Direito da U.C.

Presentemente exerce actividade docente, como assistente, na Escola Superior de Tecnologia, integrada no Instituto Politécnico de Viseu, no curso de Engenharia Electrotécnica.

Viseu, Janeiro de 1994

Prefácio

Esta Dissertação de Mestrado integra-se na satisfação de requisitos, do Programa de Mestrado em Engenharia Electrotécnica e de Computadores, na Área de Especialização de Informática Industrial. Versa sobre a *Localização e Reconhecimentos de defeitos em Madeira de Pinho por Visão Computacional,* nas áreas da análise de imagem e do reconhecimento de padrões.

Embora este trabalho não esteja directamente relacionado com o projecto *LASERMADEIRA*, foi motivado em grande medida por ele. A oportunidade de trabalhar nesta área da visão computacional e, nessa medida, na área da inspecção automática de defeitos em madeiras, deve-se ao facto de um grupo de professores e alunos da Faculdade de Engenharia da Universidade do Porto, estar ligado ao referido projecto, projecto que tinha sede de desenvolvimento no INESC-Porto.

O objectivo do trabalho prende-se com a implementação e desenvolvimento de um sistema de procedimentos informáticos que permitam trazer contributos para o desenvolvimento de um sistema de *Localização e Reconhecimentos de defeitos em Madeira de Pinho por Visão Computacional.* Ao descrever em cada capítulo vários tipos de técnicas de processamento ou análise de imagem, não se pretende fazer o mero repositório destas técnicas, mas apoiar caso a caso os processos de segmentação, recolha de características e classificação implementadas, com vista à *identificação e localização dos defeitos em madeira.* Por essa razão, o conjunto de técnicas apresentado não é exaustivo e muitas outras poderiam ter sido apresentadas.

Contudo a exposição reflecte critérios adoptados ao longo do trabalho, tendentes à resolução dos objectivos propostos, critérios esses que não são alheios à experiência gradualmente adquirinda ao longo da sua execução.

Uma das principais dificuldades que surge a qualquer desenhador de um sistema informático que trabalhe com imagem, prende-se com a selecção do algoritmo, ou dos algoritmos, a utilizar na implementação do referido sistema. Isso deve-se ao facto de os algoritmos propostos para o desempenho de cada tarefa a executar; quer se trate de pré-tratamentos, segmentação, morfologia, ou análise de imagem, serem numerosos e o seu desempenho não ser forçosamente independente do problema a tratar. Por outro lado sendo uma área de investigação ainda em processo de desenvolvimento, permite que todos os anos surjam novas propostas e novos processos de tratamento e análise de imagem, com maior ou menor eficácia, quase sempre muito dependentes do problema específico que visa resolver. Nessa escolha, uma resposta eficaz e inteligente não é possível, sem que primeiro se tenha atingido um conhecimento e experiência razoável nesses domínios. Essa atitude determinou que para a implementação do sistema proposto, se tivesse feito préviamente uma abordagem de estudo/aprendizagem da implementação das ferramentas, utilizadas posteriormente no processo desenvolvido e apresentado no capítulo VII da dissertação. Esse facto determinou também que grande parte do labor desenvolvido na preparação desta dissertação tenha sido dedicado a testes e avaliações das diversas técnicas referenciadas ao longo dos seus seis primeiros capítulos.

Os primeiros testes foram efectuados em imagens de pranchas de madeira, previamente gravadas por um sistema de aquisição de imagem DT-IRIS, versão V01.02 numa VAXstation (MicroVAX) - DEC, operando sob Sistema Operativo VMS, do qual foram utilizadas a sua biblioteca de referência(processamento de imagem e de funções gráficas). Razões técnicas impediram o prosseguimento do trabalho neste sistema.

O trabalho evoluiu então gradualmente, através da implementação de funções gráficas e de processamento de imagem, à medida que se iam tornando necessárias para a análise das madeiras, no projecto respectivo, sem o apoio de qualquer biblioteca de imagem. A máquina utilizada era do tipo PC compatível, com 1Mb de RAM e operando com um sistema de programação C++ da Borland. As imagens tinham inicialmente o formato de 320x200 (colunas x linhas), de modo a poderem ocupar um segmento de memória de endereçamento directo. Note-se que neste espaço, e pelo facto do sistema inicialmente desenvolvido apenas utilizar 16 cores diferentes (níveis de cinzento), num segmento de memória poderiam ter sido utilizadas imagens com o dobro da área. O passo seguinte foi dado pela utilização de peças de madeira serrada, adquiridas em 256 níveis de cinzento, e pela adaptação do sistema, a uma forma mais ajustado às necessidades de processamento, através da utilização de três *"buffers"* de imagem declaradas como variáveis globais. Estas três variáveis constituem três vectores de ponteiros, um ponteiro por cada linha da imagem, apontando respectivamente, para uma posição de memória, *alocada(reservada)* na memória para armazenar uma linha de imagem. Este espaço

é *alocado(reservado)* no *heap* da memória, em função dos *"buffers"* de trabalho necessários em cada ponto de execução do algoritmo. A excepção prende-se com a memória endereçada por **buf1[200]*, que é *alocada(reservada)* no inicio do programa e libertada no fim, já que é utilizada para armazenar a imagem alvo do sistema, de que necessita sempre, e que, no mínimo, corresponde a uma imagem carregada.

Os resultados obtidos por processos de visão computacional de objectos naturais, face aos resultados obtidos por identificação visual, são geralmente pobres. Isso deve-se ao facto de que, enquanto os processos de segmentação de uma imagem por via automática se baseiam no valor do *pixel*, gradiente ou probabilidades derivadas de uma área determinada da imagem, os processos de visão humana baseiam-se na experiência e permitem integrar forma, textura, orientação, contornos, informação de cor e intensidade luminosa, além de resolução local ou global. Além disso, a experiência visual pode ser confrontada, enquanto a extrapolação e interpolação são utilizadas na resolução de conflitos.

Saliente-se ainda que a informação contida nas imagens não se transforma em conhecimento, senão por interposição do homem. O conhecimento tem a ver em diversos graus, com um sistema de valores, com um sistema conceptual, com processos racionais e com sistemas semióticos veiculados pelas sociedades, muitas vezes interrelacionados mais por esferas do caótico, do que por mecanismos inteligíveis de causa efeito.

O anexo sétimo faz uma listagem do sistema mínimo de imagem que foi sendo desenvolvido à medida que se construía o projecto. Cresceu em função das variáveis características, ou operações de processamento e análise que iam sendo necessárias para a formalização dos conceitos, ou para a visualização de ideias que iam ocorrendo. Por esse motivo, o sistema envolve funções que foram utilizadas na solução final de *identificação localização e reconhecimento de defeitos em madeira* e outras que se tornaram completamente obsoletas à medida que o trabalho foi evoluindo e portanto desnecessárias. A sua apresentação ou referência tem apenas um interesse meramente histórico, e somente pretende ilustrar o trabalho preliminar efectuado para se chegar à solução final do algoritmo de *localização e reconhecimento de defeitos em madeira de pinho*. O sistema foi assim crescendo gradualmente através de ensaio, falha, repetição, procurando sempre minimizar as insuficiências anteriores.

As versões apresentadas no capítulo VII, apesar de finais, não são definitivas e o trabalho poderia ser ainda aperfeiçoado. Acredita-se que com algum esforço, trabalho e tempo adicionais, poderiam ser conseguidos melhores resultados, do que os conseguidos até hoje. Ou seja, a consistência do sistema de *localização e reconhecimento de defeitos em madeira de pinho* poderia ser melhorada, mas o principal objectivo do trabalho considera-se atingido, já que consistia fundamentalmente na elaboração de um estudo sobre o assunto, na recolha de dados, e na apresentação de propostas.

Os resultados, não sendo os ideais, são apesar de tudo aceitáveis, se se considerar que o erro de classificação cometido para alguns defeitos já é bastante reduzido. Essa conclusão pode ser reforçada, se se atender que esses defeitos (*B_K*, *NÓS*, *BRN* e *BCH*), são efectivamente os mais frequentes no pinho, e os que maiores reflexos negativos trazem para a qualidade da madeira. Para explicar a parcela correspondente aos resultados menos bons, deve-se atender ao facto do reconhecimento de *defeitos* em substâncias ou matérias naturais, apresentar muitas dificuldades, face às dificuldades apresentadas pela identificação de objectos do tipo indústrial. Isso resulta do facto das substâncias naturais, apresentarem variações muito maiores no valor das características, com potencialidades de caracterização das *classes/defeitos* a identificar, face às variações apresentadas pelas correspondentes características encontradas, nos artefactos indústriais.

Finalmente, deseja-se que este trabalho possa dar algum contributo positivo, para o aprofundamento do estudo da *identificação e localização por via automática, dos defeitos em madeira.*

ÍNDICE

Resumo xxxiii

Introdução xl

Notação xlvii

1 - **Tópicos para a Caracterização e Classificação Automática Defeitos em Madeira** 1

1.1 Introdução 1
1.2 Caracterização da madeira como material 3
1.3 Generalidades sobre defeitos em madeira 5
1.4 Defeitos principais nas madeiras 9
1.5 Objectivos e Organização de um Sistema de Visão, para detecção de defeitos e classificação de madeira 15
1.6 Conclusões 17

2- **Processamento de Imagem. Transformações no Domínio Espacial** 21

2.1 Introdução 21
2-2 Fases Envolvidas num Processo de Reconhecimento de Imagem 23
2-3 Transformação dos níveis de cinzento na imagem 26
2.3-1 Noção de histograma 26
2.3-2 Transformações por via determinística 27
2.3-3 Modificações do histograma 30
2.3-4 Restauração, Realce e suavização 38
2.3.4-1 Introdução 38
2.3.4-2 Manipulação do contraste e do brilho da imagem. Remapeamento linear da Imagem 39
2-4 Suavização. Eliminação de ruído 41
2.4-1 Introdução 41
2.4-2 Filtro de Média 43

2.4-3 Filtro de Mediana 45
2.4-4 Método Adaptativo 47
2.5 Convolução Discreta 49
2.6 Realce de orlas na imagem através de técnicas baseadas em gradiente 52
2.6-1 Noção de gradiente 52
2.6-2 Processo de realce utilizando o Gradiente 54
2.6-3 Processo de realce utilizando métodos Estatísticos 54
2.6-4 Filtro Passa Alto ("unsharp masking") 55
2.6-5 Utilizando o Laplaciano 56
2.6-6 Operadores não lineares de orlas 57
2.6-7 Detecção de zero (zero crossing) 59
2-7 Conclusões 61

3. Segmentação 65

3.1 Introdução 65
3.2 Técnicas de segmentação por regiões 67
3.2-1 Introdução 67
3.2-2 Técnicas Globais não Paramétricas 68
3.2.2-1 Divisão do histograma por *"thresold"* - Nivel Segmentação. Binarização 68
3.2.2-2 Método de Forma 70
3.2.2-3 Método de análise de concavidades no histograma 70
3.2.2-4 Método de Nobuyuki Otsu 72
3.2.2-5 Métodos Utilizando Entropia 76
3.2-3 Procura dum Patamar Optimizado por Processos Paramétricos 76
3.2-4 Níveis de segmentação locais 78
3.2.4-1 Introdução 78
3.2.4-2 Níveis de segmentação Múltiplos 80
3.2.4-3 Níveis de Segmentação Variáveis 80
3.2.5 Técnicas Baseadas em Divisão e Conquista ("split and merge"). Crescimento de Regiões 82
3.3 Segmentação baseada na determinação do contorno 86
3.3-1 Introdução 86
3.3-2 Extracção de Fronteiras Através de Pontos do Contorno (*"edges"*) 88
3.4 Segmentação por análise de texturas (região de texturas) 90
3.5 Codificação de imagem binária bidimensional por quadtree 94
3.6 Exemplo de segmentação, utilizando técnicas de divisão de imagem por região, servindo-se de estrutura quaternária do tipo árvore 96
3.7 Conclusões 102

4-Tratamento de imagens binárias. Operações lógicas Contornos e Morfologia 105

4-1 Introdução 105
4.2- Operações com Imagens Binárias. Operações Booleanas Locais 106
4.3 Propriedades das imagens binárias 110
4.3-1 Conectividade 110

4.3-2 Dilatação (*"expand"*) e erosão (*"shrink"*) 113
4.4 Técnicas baseadas em morfologia binária 114
4.4-1 Introdução 114
4.4-2 Operações morfológicas sobre imagens, utilizando operadores de Minkowski 115
4.4.2-1 Generalidades 115
4.4.2-2 Adição de Minkowski e Dilatação 116
4.4.2-3 Subtracção de Minkowski e Erosão binária 118
4.4.2-4 Operações de Reflexão e Diferença de Minkowski 119
4.4.2-5 Operações de abertura e fecho 120
4.5. Descrição de formas 123
4.5-1 Atributos métricos 123
4.5-2 Descritores de contorno 124
4.5.2-1 Cadeia numérica e números de forma 124
4.5.2-2 Código de cadeia - *"Chain code"* 125
4.5.3 Descritores de superfície 125
4.5.3-1 Atributos/descritores topológicos (números de Euler) 125
4.5.3-2 Descritores de geometria elementar 127
4. 5.3-3 Momentos 128
4.5.3-4 Representação duma superfície pelo eixo médio e esqueletização 130
4.6 Extracção da fronteira 132
4.7 Conclusões 134

5 - **Texturas** 139

5-1 Introdução 139
5 2 Textura: Processo Estrutural e Sintáctico 141
5.3-Textura: Processo Estatístico 142
5.3-1 Estatística do 1ºgrau 142
5.3-2 Método *SGLDM* (*"Spacial Gray Level Dependence Matrices"*) 145
5.3-3 Método *GLDM* (*"Gray Level Difference Method"*) 152
5.3-4 *GLRLM* (*"Gray Level Run Length Method"*) 154
5.4 Textura: Métodos Espectrais. Análise do espectro de potência de Fourier 157
5.5 Textura baseada nas propriedades espaciais da geometria 158
5.5-1 Características geométricas locais 158
5.5-2 Medidas de extremos relativos 160
5.5-3 Arestas por unidade de Superfície 161
5-6 Medidas de Energia das Texturas 161
5-7 Autocorrelação 162
5-8 Conclusões 163

6 **Reconhecimento de padrões** 167

6.1 Conceitos básicos 167
6.2 Teoria da decisão por via não paramétrica 173
6.2-1 Funções de decisão linear 173

6.2-2 Classificação de padrões por mínima distância 175
6.2-3 Um exemplo de classificação por *CPDM* 178
6.3. Funções de decisão obtidas com aprendizagem. Técnicas não paramétricas 182
6.3-1 Generalidades 182
6.3-2 Procedimentos de aprendizagem através da correcção do erro 184
6.3.2-1 Regras de Aprendizagem 184
6.3.2-2 Algoritmo de Aprendizagem 187
6.3.3 Técnicas baseadas em gradiente 189
6.3.3-1 Introdução 189
6.3.3-2 Determinação da função critério *"Perceptron"* 190
6.3-4 Exemplo de classificação, através de hiperplanos calculados pelo algoritmo de correcção do erro 191
6.3-5 Considerações finais sobre algoritmos de aprendizagem por correcção do erro 193
6-4 Procedimentos através da Minimização do Erro Médio Quadrático (LMSE) 194
6.4-1 Método da Matriz Pseudo Inversa 194
6.4-2 Método de Ho-Kashyap 196
6.4-3 Exemplo de Classificação pelo algoritmo *LMSE* 197
6.5 Funções de Decisão Estatística 200
6.5-1 Generalidades 200
6.5-2 Função de Decisão de Bayes 201
6.5-3 Decisão por Máxima Probabilidade 202
6.5-4 Funções de Decisão para distribuições Normais de Padrões 205
6.5-5- Função discriminante, para distribuições de multivariável 208
6.5-6 Exemplo de Classificação pelo Método de Bayes 209
6.6 Conclusões 212

7- Localização e identificação de defeitos em madeira de pinho. Implementação e resultados 217

7.1 Introdução 217
7.2 Recursos Informáticos e Estruturas Utilizadas 222
7.3 Processo ad hoc de Reconhecimento de Defeitos 223
7.3-1 Introdução 223
7.3-2 Arquitectura do Sistema rdmp/ad-hoc (Algoritmo de reconhecimento em madeira de pinho) 225
7.3-3 Descrição do sistema rdmp/ad-hoc(programa de reconhecimento em madeira de pinho) 227
7.3-4 Estruturas de dados utilizadas 234
7.4 Selecção e Extracção de Características 237
7.4-1 Introdução 237
7.4-2 Ordenação de características, através das suas propriedades intrínsecas de aglomeração 239
7.4-3 Cálculo do factor de ordenação, em função do valor da dispersão das características 241
7.4-4 Separabilidade entre classes por medidas de distância 243
7.4-5 Separação Numérica de Classes-Método Jeffries-Matusita (JM) 244

7.4.5-1 Descrição 244
7.4.5-2 Separação inter-classes. Cálculo do factor de distancia JM, em defeitos de madeira 246
7.5 Processo de reconhecimento de defeitos em madeira, por decisão singular de etapa única 248
7.6 Processo de decisão binária global 251
7.6-1 Descrição do processo 251
7.6-2 Selecção de características no processo global 253
7.7 Processo de decisão binária hierarquizada 256
7.7-1 Descrição do processo 256
7.7-2 Árvores de decisão 257
7.7-3 Arquitectura do Algoritmo 258
7-8 Processo conjunto de Classificação Global e Hierarquica 262
7.9 Identificação de defeitos previamente marcados 264
7.9-1 Introdução 264
7.9-2 Arquitectura do Algoritmo 265
7.10 Conclusões 269

8- Conclusões 275

Anexos 287

Bibliografia 375

ÍNDICE DE QUADROS E FIGURAS

1.- **Tópicos para a Caracterização e Classificação Automática de Defeitos em Madeira** 1

figura 1.1. Estrutura da madeira 3
figura 1.2. Secção transversal de um tronco 4
figura 1.3. Nós agrupados e suas medições 10
figura 1.4. Defeitos em madeira - Fibras torcidas 12
figura 1.5. Imagem de prancha de pinho, com diversos defeitos. Nó, fissura, background, bolsa de resina e queimado da plaina 18

2 - **Processamento de Imagem. Transformações no Domínio Espacial.** 21

figura 2.1. Esquema de blocos de um processo de reconhecimento 23
figura 2.2. Prancha de Pinho, com nó solto 27
figura 2.3. Densidade e distribuição de probabilidades dos níveis de cinzento da imagem representada na figura 2. 2 27
figura 2.4. Operações pontuais com imagens 28
figura 2.5. Operações locais, para o caso de oito vizinhos 28
figura 2.6. Funções de transferência, para transformação determinística de níveis de cinzento na imagem 29
figura 2.7. Transformação logarítmica da imagem da figura 2.2 30
figura 2.8. Imagem inversa da imagem da figura 2.2 30
figura 2.9. Modificação do histograma - equalização 33
figura 2.10. Função de transferência para um histograma equalizado 34
figura 2.11. Imagem da figura 2.2 equalizada 35
figura 2.12. Densidade de probabilidades para as imagens: equalizada, logaritmizada e invertida da imagem da figura 2.2 35
quadro 2.1. Funções de transferência para modificação de histogramas, segundo formato previamente definido 36
figura 2.13. Imagem de madeira com baixo contraste, marcada a giz 36
figura 2.14. Imagem da prancha da figura 2.13 depois de operação de realce 36
figura 2.15. Imagem da figura 2.13 depois de hiperbolização (raiz cúbica) 37
figura 2.16. Imagem da figura 2.13 depois de hiperbolização (logarítmica) 37
figura 2.17. Processo de transformação de histograma, para uma forma pretendida. 38
figura 2.18. Função de transferência utilizada no realce de imagem 39
figura 2.19. Funções de probabilidades de ocorrência dos níveis de cinzento, nas imagens das figura 2.13, 2.14, 2.15 e 2.16 40
figura 2.20. Várias máscaras usadas em suavização de imagens 44
figura 2.21. Prancha com defeitos: Bolsas de Resina 45
figura 2.22. Imagem depois de aplicado filtro de média 45

figura 2.23. Máscaras usadas em filtros de mediana 46
figura 2.24. Imagem correspondente à da figura 2.22 depois de filtrada por filtro de mediana (máscara quadrada) 46
figura 2.25. Histogramas das imagens2.22, 2.23 e 2.25 46
figura 2.26. Esquema geral de um processo de tratamento adaptativo 47
figura 2.27. Implementação do gradiente 53
figura 2.28. Máscaras para filtros passa alto 55
figura 2.29. Filtro passa alto sobre a imagem da figura 2.21 55
figura 2.30. Filtro de Roberts sobre a imagem da figura 2.21 55
figura 2.31. Implementação do Laplaciano 56
figura 2.32. Detecção de edges por cruzamento de zero da 2ª derivada 59
figura 2.33. Diagrama de blocos de um algoritmo de determinação de edges (calcula o Laplaciano e avalia a variância local, para a tomada de decisão) 60
figura 2.34. Filtro de Sobel a partir da imagem da fig. 2.21 60
figura 2.35. Laplaciano a partir da imagem da fig. 2.21 60

3. Segmentação 65

figura 3.1. Representação da imagem descretizada, e histograma dos níveis de cinzento 68
figura 3.2. Imagem de prancha de madeira 71
figura 3.3. Imagem binária, correspondente à da figura 3.2, depois de segmentação 71
figura 3.4. Histograma da imagem da figura 3-2 72
figura 3.5. Imagem de prancha madeira 75
figura 3.6. Imagem binarizada, por nível de segmentação, determinado pelo método de Otsu 75
figura 3-7. Critério de medida e resultados obtidos, no processo de determinação do *"threshold"* 75
figura 3-8. Curvas de distribuição de probabilidades, agrupadas em dois níveis 77
figura 3-9. Grupos de pixels R1, R2, R3, R4, classificados por crescimento de regiões 83
figura 3-10. Conectividade 83
figura 3.11. Extensão do edge e_0. A seta indica a direcção de e_0 89
figura 3.12. Madeira com nós 90
figura 3.13. Contornos das regiões presentes na figura 3.12, extraídos através de, processo de detecção de orlas 90
figura 3.14. Vizinhanças de um ponto e cálculo do índices de não homogeneidade numa vizinhança 93
figura 3.15. Pirâmide de codificação de imagem binária 94
figura 3.16. Representação de uma imagem binária por árvore quaternária 95
figura 3.17. Processo de divisão da imagem, em a), e arquitectura de divisão da árvore quaternária 97
figura 3.18.Esquema representando a estrutura de um nó da árvore quaternária 97
figura 3.19. Organização simplificada e parcial da estrutura de uma árvore quaternária, utilizada na retenção de informação do algoritmo de segmentação por *"quadtree"* 98

figura 3.20. Diagrama de blocos, mostrando o processo de implementação do algoritmo de segmentação, por divisão da imagem e utilização de *quadtree* 99
figura 3.21. Imagem de madeira com nós 100
figura 3.22. Imagem da figura 3.21 segmentada por *"threshold"* 100
figura 3.23. Imagem da figura 3.21 depois de homogeneizada 101
figura 3.24. Imagem da figura 3.23 segmentada por *"threshold"* 101
figura 3.25. Histogramas das imagens representadas nas figura 3.21 e 3.23 101

4-Tratamento de imagens binárias. Operações lógicas Contornos e Morfologia 105

quadro 4.1. Operações lógicas "AND" "OR" "EOR" 106
quadro 4-2. Operação "NOT" 106
quadro 4-3. Operação "XOR" 106
figura 4.1. Operações booleanas sobre imagem 107
figura 4.2. Representação dos vizinhos de a_0, numa vizinhança 3x3 108
figura 4.3. Vizinhança 3x3 dum pixel isolado numa imagem 109
figura 4.4. a) Prancha de pinho com *background* e brecha de resinagem 109
figura 4.4. b) Prancha depois de binarização 109
figura 4.4. c) Prancha depois de operação de eliminação de pixels isolados 110
figura 4.4. d) Prancha depois de operação morfológica de eliminação de zonas espúrias 110
figura 4-5. Estabelecimento de *conectividade oito* e *conectividade quatro* dum pixel 111
figura 4-6. Fronteiras com conectividade quatro e oito respectivamente 111
figura 4-7. Imagem de prancha de madeira com nós 112
figura 4.8. Imagem depois de eliminados os pixels não conexos na imagem binarizada 112
figura 4.9. Outro modo de ilustração da conectividade 112
figura 4.10. Operação Erosão 113
figura 4.11. Operação Dilatação 114
figura 4.12. Operação dilatação, representada em termos de conjuntos e a sua correspondente representação binária 117
figura 4.13. Imagem binarizada, correspondente à imagem representada na figura 4.7 118
figura 4.14. Dilatação da imagem representada na figura 4.8 118
figura 4.15. Operação Erosão e o correspondente exemplo aplicação binária 119
figura 4.16. Imagem correspondente à prancha da figura 4.8, depois de erosão 121
figura 4.17. Imagem correspondente à prancha da figura 4.8, depois de abertura 121
figura 4.18. Extracção de primitivas da imagem, através de operações de Abertura e Diferença 122
figura 4.19. Imagem binária depois de sujeita a operação de fecho, através de um elemento estruturante quadrado de aresta ρ_0 122
figura 4.20. Códigos de Cadeia e Números de Forma 124
figura 4.21. Codificação de Cadeias - *"Chain Code"* 125

figura 4.22. Diferentes tipos de células quaternárias(*"bit quad"*) 126
figura 4.23. Distância entre pixels numa imagem 127
figura 4.24. Descritores geométricos de um objecto 128
figura 4.25. Extracção do eixo médio 131
figura 4.26. Determinação da profundidade de um pixel 131
figura 4.27. Mapa de profundidades e esqueletização 132
figura 4.28. Procura do contorno seguindo a fronteira 133
figura 4.29. Madeira com afloramentos de casca 133
figura 4.30. Fronteira retirada por pesquisa/correcção do contorno 134
figura 4.31. Fronteira retirada por profundidade 134

5 - **Texturas** 139

figura 5.1. Etapas dum sistema de análise de texturas 140
figura 5.2. Imagem 1 - madeira de pinho com nó e cardido 144
figura 5.3. Imagem 2 - madeira de pinho *SDF* e *"background."* 144
quadro 5-1. Estatística do 1ºgrau, para as imagens 1 e 2, das figura 5.2 e 5.3 144
figura 5.4. Posicionamento de pixels e matriz de co-ocorrência 146
figura 5.5. Construção da matriz de co-ocorrência 148
figura 5.6. Densidade de probabilidades a duas variáveis, para a imagem da figuraura 5.2 com $d=1$ e $\theta=0º$ 148
figura 5-7. Densidade de probabilidade a duas variáveis, para a imagem da figura 5.3 com $d=1$ e $\theta=0º$ 149
quadro 5-2. Medidas estatísticas do 2ºgrau, para as imagens 1 e 2, das figura 5.2 e 5.3 151
quadro 5-3. Medidas *GLDM* retiradas das imagens 1 e 2, representadas nas figura 5.2 e 5.3 154
quadro 5-4. Quadro de medidas características obtidas das imagem, representadas na figura 5.2 e 5.3, obtidas pelo método *GLRLM* 156
quadro 5.5. Exemplo de medidas características de natureza estatísticas, encontradas para os defeitos, casca, bolsa de resina e nós em prancha de madeira de pinho 157
figura 5.8. Texturas de 2ªordem determinadas pela distribuição da orientação dos *edges* na imagem 160

6 **Reconhecimento de padrões** 167

figura 6.1. Representação por blocos, dum processo de reconhecimento de padrões 167
figura 6.2. Classes de padrões, separáveis 168
figura 6.3. Espaço de características a duas dimensões e função de decisão, entre duas classes de padrões 170
figura 6.4. Superfícies de separação entre três classes num espaço bidimensional 170
figura 6.5. Diagrama de blocos dum sistema de classificação 171
figura 6.6. Diagrama blocos dum sistema de reconhecimentos de padrões, por um processo adaptativo 171

figura 6.7. Classes de padrões separáveis - a) Cada classe separada por uma única superfície de decisão b) Classes separadas duas a duas por uma única superfície de decisão 175
figura 6.8. Classificação por proximidade 175
figura 6.9. Fronteira de decisão entre duas classes, caracterizadas por possuírem protótipos com aglomeração simples 177
figura 6.10. Diagrama de blocos dum algoritmo de classificação por mínima distância 178
figura 6.11. Imagens de pranchas de madeira, com um defeito do tipo nó: a)Defeito assinalado na prancha, com objectivos de aprendizagem 179
b)Prancha de madeira, correspondente à da figura a) 179
quadro 6-1. Amostra dos valores das características x_1, x_2 ,x_3 e x_4, a partir das quais se definem as medidas centrais dos protótipos, z_1, z_2 ,z_3 e z_4. Medidas extraídas por processo de ensaio em pranchas de madeira 180
quadro 6-2. Tabelaela de exemplificação, de classificação de defeitos por *CPDM* 180
figura 6.12. Valor das características x_1, x_2 e x_3 por classe de defeito 181
figura 6.13. Valores por característica para as diferentes classes 181
figura 6.14. Dispersão de características por classe 181
figura 6.15. Representação de hiperplanos num problema de duas classes 183
figura 6.16. Hiperplano de decisão, correcção da sua posição na direcção de z_i^* 186
figura 6.17. Procedimentos de ajuste num algoritmo de: • - incremento fixo; ■ - correcção absoluta; ▲ - correcção fraccional 188
quadro 6-3. Protótipos de classificação de amostras de madeira a) Média às diferenças b) Contraste às diferenças a processo c) 2º momento angular às diferenças: (*GLDM*) *sdf* - Madeira sem defeito *crd* - cardido *c_z* - Azulado da madeira 191
quadro 6.4. Tabelaela com alguns valores intermédios, dos vectores de decisão w_1, w_2, w_3, sujeitos a processo de correcção pelo algoritmo descrito em 6.3.2-2 192
quadro 6.5. Classificação de seis amostras através dos hiperplanos de decisão w_1, w_2, w_3 obtidos pelo método de aproximações sucessivas 193
quadro 6.6. Matriz produto [Z'. Z][1] 198
quadro 6.7 Matriz inversa [Z'. Z]-1[1] 198
quadro 6.8. Matriz Z#=[Z'. Z]-1Z'[1] (matriz pseudo-inversa) 198
quadro 6.9. Matriz produto [Z'. Z][2] 198
quadro 6.10. Matriz inversa [Z'. Z]-1[2] 198
quadro 6.11. Matriz Z#=[Z'. Z]-1Z'[2] (matriz pseudo-inversa) 199
quadro 6.12. Matriz produto [Z'. Z][3] 199
quadro 6.13. Matriz inversa [Z'. Z]-1[3] 199
quadro 6.14. Matriz Z#=[Z'.Z]-1Z'[3] (matriz pseudo-inversa) 199
quadro 6.15. Hiperplanos de decisão, obtidos por *LMSE* 199
quadro 6.16. Tabelaela de classificação de uma amostra diversificada de padrões, pertencentes a três classes diferentes 199
figura 6.18. Classificador de Bayes 205
figura 6.19. Formas diferentes de aglomeração das características para a mesma probabilidade de ocorrência 207
quadro 6.17. Vectores de médias das classes 1, 2 e 3 211
quadro 6.18. Matrizes de Covariância para as classes 1, 2 e 3 211

quadro 6.19. Matrizes inversas da Covariância, de cada uma das classes 211
quadro 6.20. Tabelaela de demonstração, duma operação de classificação pelo método de Bayes 211

7- Localização e identificação de defeitos em madeira de pinho implementação e resultados 217

figura 7.1. Imagem sem defeitos com veios bastante marcados 218
figura 7.2. Imagem sem defeitos (*SDF*), com veios muito suaves, identificada pelo RDMP/ad-hoc 218
figura 7.3. Histogramas das imagens das figura 7.1 e 7.2 219
quadro 7-1. Listagem de defeitos na madeira 223
figura 7-4. Esquema de decisão do algoritmo *RDMP*/ad-hoc [Reconhecimento de defeitos em madeira de pinho - ad hoc] 224
figura 7.5. a) Prancha de pinho com nós, b) Segmentada e c) Com os defeitos localizados e reconhecidos 226
quadro 7-2. Quadro de características por nó e tipo de decisão(esquema da fig. 7.4) 227
figura 7-6. Construção da árvore quternária, através do algoritmo de segmentação por subdivisão sucessiva da imagem 229
figura 7.7. Diagrama de fluxos do programa de imagem *RDMP*/ad-hoc 233
figura 7.8. Imagem de pinho com nós 234
figura 7.9. Classificação obtida da fig. 7.8, por rdmp/ad-hoc 234
figura 7.10. Estrutura dos nós das listas ligadas: lstpc1 e lista1 236
figura 7.11. Processos de redução da dimensão dum vector de características, representativo duma classe de padrões 238
quadro 7-3. Factor de ordenação de características em função da aglomeração, por classe 242
quadro 7.4. Coeficientes de distância intercalasse(t_l), para as variáveis 9, 21 e 30 247
quadro 7.5. Coeficientes de distância intercalasse(τ_i), para as variáveis 14, 20 e 32 247
quadro 7.6. Coeficientes de distância intercalasse(τ_i), para as variáveis 5, 10 e 16 248
figura 7.12. Esquema de classificação, por decisão singular, com base nas características e funções de decisão indicadas 249
quadro 7.7. Taxas de sucesso de classificação correcta pelo processo de decisão singular 250
figura 7.13. Diagrama de blocos do processo de decisão binária *sdf/cdf* 251
quadro 7.8. Correlação entre variáveis características definidas para um problema de classificação de duas classes 252
quadro 7.9. Resultados da classificação sdf/cdf 253
quadro 7.10. Quadro de representação dos valores do factor *F* para as quarenta e oito características globais definidas 255
figura 7.14. Árvore de decisão, em processo de classificação hierarquizado, para subregiões da imagem segmentada 256
figura 7.15. Arquitectura de uma árvore de decisão binária 257
figura 7.16. Diagrama de fluxos do sistema de detecção e identificação de defeitos. Processo hierárquico de decisão binária 260

figura 7.17. Imagem classificada, com duas regiões sdf e a presença de backup 261
figura 7.18. Imagem classificada com presença de nó 261
quadro 7.11 Resultados de classificação, obtidos através de processo de decisão binária hierarquizada, do *sistema de localização e identificação de defeitos em madeira de pinho* 261
quadro 7.12 Resultados de classificação, obtidos através de processo de decisão binária global e hierarquizada, do *sistema de localização e identificação de defeitos em madeira de pinho* 262
figura 7.19. Esquema de decisão do sistema final de localização e reconhecimento em madeira de pinho 263
figura 7.20. Imagem original marcada a giz 266
figura 7.21. Imagem depois de operação de realce 266
figura 7.22. Imagem depois de binarizada 266
Figura 7.23. Contorno extraído da imagem 266
figura 7.24. Histogramas das imagens das fig. 7.20 e 7.21 267
figura 7.25. Diagrama de fluxos do algoritmo de detecção de marcações a giz em madeira 267
figura 7.26. Bolsas de resina classificadas por processo hierárquico 271
figura 7.27. Azulado da madeira classificadas por processo hierárquico 271
figura 7.28. Brecha de resinagem classificadas por processo hierárquico 271
figura 7.29. Medula e madeira sem defeito classificadas por processo hierárquico 271
figura 7.30. Nós classificadas. Erro em SDF. Processo de classificação hierárquica 271
figura 7.31. Brechas de resinagem na madeira. Processo de classificação hierárquica 271
figura 7.32. Brechas de resinagem classificadas por processo hierárquico 271
figura 7.33. Nós e b_k na madeira classificados por processo hierárquico 271

LOCALIZAÇÃO E RECONHECIMENTO DE DEFEITOS POR VISÃO COMPUTACIONAL EM MADEIRA DE PINHO PINUS SILVESTRIS

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

Relator chefe: Professor Dr. *Manuel* Mendonça
Co-Relator: Dr.Campos Neves

DISSERTAÇÃO DE MESTRADO

Rui Bebiano

RESUMO

Neste trabalho faz-se um estudo sobre o tema da *localização e reconhecimento de defeitos em madeira* através de visão computacional. Enquadrados nesta problemática, descrevem-se alguns algoritmos, em diversos domínios do tratamento de imagem e implementa-se um sistema informático, capaz de testar as hipóteses teóricas formuladas ao longo desse estudo.

Na primeira parte, constituída pelo primeiro capítulo, caracterizam-se os diversos defeitos que podem inquinar a madeira, e as consequências que a sua presença pode acarretar na degradação das suas qualidades físicas, nomeadamente as mecânicas. Referem-se ainda aspectos genéricos ligados à caracterização e detecção automática de defeitos por via óptica (ou não).

A segunda parte é constituída pelos capítulos II a VI, onde se descrevem alguns algoritmos de tratamento de imagem. Inicia-se o estudo com os pré-tratamentos, nomeadamente transformações de histograma, realce de imagem e operações de filtragem. Seguem-se a apresentação e teste de algoritmos de segmentação e recolha de padrões nas imagens. Esses padrões são recolhidos a partir das propriedades que caracterizam as micro-texturas das superfícies de segmentação ou das suas propriedades geométricas. O capítulo VI versa algumas técnicas de classificação e de reconhecimento de padrões em imagens, e ainda técnicas de aprendizagem não paramétrica dos hiperplanos de decisão (ou funções de decisão) utilizados na classificação.

A terceira e última parte (capítulo VII) descreve um conjunto de algoritmos implementados ao longo do estudo, desenvolvidos com a finalidade de avaliar e testar as potencialidades das características seleccionadas e o processo de decisão envolvido no processo de *localização e identificação dos defeitos na madeira*. Além dos processos de selecção de características descritos, apresentam-se os resultados das classificações efectuadas pelos algoritmos e faz-se a avaliação de cada um dos sistemas de localização e reconhecimento dos defeitos.

Palavras chave: aprendizagem, classificadores, extracção de contornos, funções de decisão, hiperplanos de decisão, propriedades morfológicas, segmentação, selecção de características, texturas.

ABSTRACT

IN this work, a study on the localization and recognition of defects in wood is performed through computer vision. In this framework, some algorithms are described in several domains of the digital image processing.

In the first part, which corresponds to the first chapter, several wood defects are characterized. The consequences of their occurrence in the degradation of the physical properties of the wood, namely the mechanical properties, are referred. Generic aspects connected with the characterization by optical means (or otherwise) are also considered.

The second part goes from chapter II to VI. Here the main techniques of digital image processing are described. The study begins with the pre-processing, namely the histogram modification, enhancement of image and filtering operations. The presentation and testing of segmentation algorithms and pattern extraction, follows. These patterns are obtained, from the properties which characterize the micro textures of the segmented regions or, of their geometrical properties. Chapter VI deals with some techniques of classification and identification of image patterns, as well as, techniques of non parametric training using decision hyperplanes (pattern classification techniques).

The third and the last part, chapter VII, describes a collection of algorithms developed in this study with the purpose of assessing and testing the potentialities of the selected characteristics. Moreover, it also describes the decision process involved in the location and identification of defects in pine wood. Besides the process of feature selection, the results and assessment of each system of location and identification of defects are presented.

Key words: classifiers, edge extraction, decision function, decision hyperplanes, feature selection, morphology, segmentation, pattern classification, texture analysis.

LOCALIZAÇÃO E RECONHECIMENTO DE DEFEITOS POR VISÃO COMPUTACIONAL EM MADEIRA DE PINHO PINUS SILVESTRIS

"A Imagem nunca é só um facto objectivo: é sempre um ponto de vïsta"

Jean Paul Colleyn

Introdução

Esta dissertação aborda fundamentalmente, a temática da *localização e identificação de defeitos em madeira* serrada e aparelhada. Mais precisamente, a *detecção e identificação de defeitos em madeira de Pinho Bravo - pinus silvestris*, por visão computacional. Essa operação, pressupõem a capacidade do algoritmo, para localizar adequadamente o defeito ou os defeitos presentes nas pranchas de madeira, e a capacidade para os poder discriminar, entre vários, num conjunto diverso de *classes/defeitos*, de possível categorização. São esses objectivos que constituem, a principal linha orientadora deste trabalho.

Doutro forma, pretende-se confirmar a tese, de que a *localização e identificação automática de defeitos em madeira de pinho*, é possível, e que o desenvolvimento de um sistema indústrial, suportado por um sistema de detecção deste tipo, pode ser viável se for admitido algum grau de tolerância a falhas e erros do sistema de detecção. Ou seja, se os dersperdícios forem, dentro de uma certa margem, tolerados.

Foram considerados na tipificação das pranchas de madeira vários tipos de *classes/defeitos*, num número total de oito: defeito *Casca(CSC)* que podem ser inclusões completas de casca no lenho, ou vestígios de casca nos seus bordos; defeito *nós(NO)* entendidos genericamente como porções de ramo inclusos na madeira, podendo variar muito no que diz respeito à forma, inserção, dimensão etc.; defeito *bolsas de resina (BRN)* que são inclusões de resina no estado sólido no tecido lenhoso; defeito *brechas de resinagem(BCH)* que representam feridas de resinagem cicatrizadas; defeito *cardidos e ardidos(CRD)* que são formas de podridão branca, numa primeira fase com cor avermelhada; e o defeito *azulados(AZLD)* que são manchas provocadas por fungos na madeira. A estes seis defeitos têm de ser acrescentados mais dois defeitos com características muito especiais: o defeito *backup(B_K)* e o pseudo defeito *madeira sem defeito(SDF)*. O *backup(B_K)* e a

madeira sem defeito(SDF) não são propriamente defeitos da madeira, mas *classes* que, na fase de reconhecimento do algoritmo têm de ser separadas, no caso do *backup*, e precisam de ser ignoradas, no caso do *SDF*. A identificação necessita portando de ser feita nestes casos. Estes defeitos correspondem, no caso do *b_k*, ao fundo negro sob o qual é colocada a madeira, que pode ou não existir, e a *classe SDF* corresponde a porções de madeira escura localizadas erroneamente no processo de segmentação.

O primeiro capítulo aborda o tema genérico dos defeitos em madeira. No desenvolvimento deste capítulo estudam-se aspectos tão importantes como a caracterização da madeira como material, envolvendo referências às suas propriedades físicas e mecânicas e faz-se o levantamento das características que deveria possuir um sistema de laboração de peças de madeira serradas, com detecção de defeitos incorporada.

O capítulo termina com uma abordagem pormenorizada dos defeitos mais frequentes do fuste e das peças serradas de madeira, e com algumas referências às consequências que esses defeitos reflectem na alteração da qualidade e das propriedades físicas dessa madeira e da sua consequente desvalorização económica.

Saliente-se que esta dissertação segue de perto o percurso que foi traçado ao longo da execução do trabalho. Ou seja, aborda primeiro as técnicas de pré tratamento de imagens digitalizadas, ao que se seguem abordagens sobre técnicas de segmentação, de análise morfológica de imagem, análise de imagem através de texturas e processos de reconhecimento de padrões.

O trabalho procura avaliar também as potencialidades das características retiradas da imagem, para *classificar* cada uma das regiões defeituosas delimitadas. Estas regiões são padronizadas como se fossem superfícies com uma determinada textura, a partir das quais se podem efectuar medidas e a partir delas, extrair conclusões específicas acerca da classe a que pertencem. O processo envolve contudo algumas dificuldades, derivadas fundamentalmente de duas ordens de razões:

1- As superfícies de madeira, sendo substâncias naturais, não apresentam propriedades características (variáveis características de imagem) com a estabilidade suficiente para que a caracterização dos defeitos possa ser feita com facilidade.

2- O facto dos testes terem sido efectuados sobre imagens, cuja aquisição foi feita em condições ambientais de iluminação e resolução não controladas, acrescenta um factor de erro adicional à medição das varáveis características.

Com os padrões retirado de um conjunto de imagens, que constituem o conjunto de treino-aprendizagem [protótipos das imagens padrão], foram seleccionadas as características que apresentavam maiores potencialidades de separação inter-classes. Ou seja, as características que permitiam com maior eficiência separar uma determinada classe das restantes.

É com um conjunto de vectores de características (seleccionadas) que se obtêm os classificadores dos defeitos, determinados durante a fase de aprendizagem supervisada. Através destes classificadores, processam-se as operações de atribuição, a cada uma das áreas segmentadas na prancha de madeira, uma classe/defeito adequada, em operação chamada de classificação.

Foram utilizados diversos tipos de classificadores: uns assentaram na metodologia de Bayes, outros em funções de distância, outros ainda em hiperplanos de separação obtidos por processos de aproximação com correcção do erro, em processo iterativo de aprendizagem.

A diversidade de técnicas de análise de imagem, já testadas e implementadas por outrem, é muito grande, quer nos refiramos a operações de pré-tratamento, de segmentação, morfologia ou reconhecimento de padrões. Por outro lado, a sua eficácia é muito dependente da aplicação em causa. Desse modo, o trabalho pode progredir, numa primeira fase, dentro de uma filosofia de aprendizagem passo a passo, tanto mais que se estava a trabalhar sem biblioteca de funções de processamento e transformação gráfica de imagem. As primeiras tarefas do trabalho envolveram portanto a implementação e teste de algumas dessas técnicas de segmentação, decomposição e análise de imagem, bem como pré-tratamento, que constituem o objecto de descrição dos primeiros capítulos desta dissertação.

O capítulo II *'Pré-Tratamento de imagem'* referência algumas operações passíveis de serem efectuadas sobre imagens. Visa a melhoria das condições objectivas de análise da imagem, tendo em vista o seu tratamento posterior. Assim se descrevem neste capítulo, depois de uma referência breve a conceitos fundamentais como o de histograma, os processos de transformação e modificação do histograma, restauração e realce de imagem. O conceito de convolução discreta e a apresentação de vários tipos de filtragens, usando este conceito, fecham o capítulo que faz ainda referência a processos de filtragem adaptativa.

Este capítulo deve a sua inserção à necessidade, que por vezes existe, de realçar ou por vezes atenuar, reconstruir ou eliminar, alguns pormenores presentes na imagem. A existência deste capítulo deve também muito a uma das fases de implementação do algoritmo de *reconhecimento de defeitos em madeira*, em que se procurava a detecção dos limites das regiões homogéneas através do realce das orlas, e da reconstrução das fronteiras a partir delas.

O capítulo III aborda o tema da segmentação de imagens e selecciona num domínio muito vasto de algoritmos propostos pela literatura dedicada a este assunto, entre outras a referenciada na bibliografia, um conjunto de técnicas, que permitam separar regiões homogéneas na imagem global. Geralmente essas regiões apresentam as mesmas propriedades de cor ou textura.

Depois da descrição e teorização de diversos processos e metodologias de segmentação, são explanados com mais pormenor a binarização de imagens através da determinação de limiares de segmentação *("thresholds")*, obtidos de dois modos diferentes: determinação do limiar de segmentação através da correspondência à maior concavidade do histograma e determinação do limiar de segmentação obtido pelo método de Moboyuki Otsu.

Outras técnicas são ainda apresentadas, como aquelas que procuram encontrar a fronteira, através da pesquisa dos segmentos de contorno perdidos, à custa da orientação dos segmentos de orlas conhecidos; ou processos que segmentam a imagem através da pesquisa de regiões homogéneas do ponto de vista da textura; ou processos que combinam técnicas de crescimento ou de partição sucessiva da imagem ou de regiões dessa imagem. O capítulo termina com apresentação de um algoritmo que combina técnicas de partição de imagem com técnicas de segmentação por textura.

O capítulo IV trata de operações que se efectuam sobre imagens binarizadas e de relações entre atributos que é possível estabelecer nessas imagens.

Além de serem feitas referências às operações lógicas elementares, são definidos conceitos fundamentais, como os de *conectividade quatro* e de *conectividade oito* entre pixels. Uma abordagem sobre morfologia e sobre as operações de Minkowski, como erosão, fecho e abertura são apresentadas.

Um estudo sobre topologia binária, através da utilização de descritores de forma da superfície, de descritores de contorno e referências aos números de Euler, fornecem ferramentas suplementares, com indicações importantes sobre a forma e distribuição dos segmentos presentes na imagem. A representação ideal da geometria de um objecto pode ser encontrada através do cálculo dos seus momentos, principalmente quando estes forem independentes de transformações RST(Rotação, Escala, Translação).

Finalmente, faz-se uma abordagem sobre primitivas retiradas das imagens binárias, como eixos médios e fronteiras extraídas por determinação da profundidade dos pixels constituintes duma região. A referência aos códigos de cadeia*("chain code")* e aos processos de reconstituição de fronteiras, completam este capítulo, que tem por objectivo introduzir técnicas, que permitam extrair da imagem, variáveis características ligadas à geometria dos blocos de imagem, obtidos por segmentação.

Num processo de análise de imagem é muito importante conseguir extrair valores característicos, à custa de medidas estatísticas representativas de determinado padrão existente na imagem. Apesar da textura de uma região poder envolver primitivas com macro-dimensões, a natureza do problema em estudo impõe-nos o

cálculo de características baseadas mais frequentemente em medidas estatísticas do primeiro e do segundo grau.

O capítulo V aborda o assunto designado por texturas, em que são referenciados processos de cálculo, de medidas estatísticas do primeiro grau, como: média e variância e medidas estatísticas do segundo grau, que envolvem o emprego de métodos referenciados por: *SGLDM(Spacial Gray Level Dependence Matrices)* donde são retiradas medidas como *entropia, energia, etc.*; o método *GGLDM(Gray Level Difference Method)* donde são retiradas medidas como *contraste* e *segundo momento ângular*; e o método *GLRLM(Gray Level Run Length Method)* donde são retiradas medidas como *SRE(Short Run Enphasis)*, *LRE(Long Run Enphasis)*, etc.

Fazem-se posteriormente referências a outros processos que permitem medir valores característicos da textura numa imagem ou numa região, através de métodos espectrais, ou de medidas estatísticas de segunda ordem baseadas na orientação e distribuição dos *'edges' (arestas)* na imagem.

O objectivo deste capítulo prende-se com a necessidade de conhecer algumas das técnicas, capazes de extrair um conjunto de valores característicos (variáveis características) de uma textura presente numa imagem, que a possa caracterizar.

O capítulo VI trata o problema genérico do reconhecimento de padrões e da aprendizagem supervisada. A necessidade de estudar um classificador capaz de separar duas ou mais classes em dois conjuntos distintos, levou à introdução deste capítulo. A aprendizagem diz-se supervisada, pelo facto do estudo do classificador ser feito através de características padronizadas, previamente extraídas e seleccionadas de um conjunto de protótipos padrão caracterizador de cada uma das classes.

Foram estudados alguns processos de classificação, uns baseados em técnicas paramétricas e outros baseados em técnicas não paramétricas. Dentro das não paramétricas foram estudados classificadores por distância (mínima distância por exemplo), classificadores obtidos por processos iterativos de aproximação à solução óptima através da correcção do erro e classificadores obtidos através de procedimentos obtidos através da minimização do erro quadrático médio[*LMSE*]. No que se refere aos processos paramétricos, foi estudado um método estatístico baseado na função de decisão de Bayes.

O capítulo VII analisa e descreve um conjunto de processos utilizados na *localização e identificação de defeitos em peças serradas e aparelhadas de pinho*. A apresentação segue o caminho que foi utilizado ao longo dos testes efectuados no processo experimental.

O primeiro processo apresentado é um processo *'ad hoc'*, assim designado por basear a escolha das suas características em testes directos sobre a imagem e na comparação empírica dos resultados obtidos. Procura primeiro identificar se a imagem é globalmente defeituosa ou não, e se contém ou não *"background"*. Se a peça de madeira contiver defeitos, a análise prosseguirá *defeito* a *defeito*, através de um processo inicial de localização do defeito potencial, seguido da sua classificação.

O capítulo faz ainda uma abordagem a alguns processos de selecção de características e referências ao conceito de extracção de características. O poder de separação de uma característica é medido de várias formas, entre elas, o da medição da distância Euclidiana inter-classes, para essa(s) característica(s), ou através da medição da distância inter-classes pelo método de Jeffries-Matusita(J-M).

Seguem-se três processos de classificação e teste de defeitos: um através de *decisão singular de etapa única*, outro, um *processo de decisão binária global(SDF/CDF)* e finalmente um *processo de decisão binária hierarquizada*. Estes processos foram avaliados e permitem concluir que o processo de decisão binária hierarquizada apresenta resultados melhores que os anteriores. É possível também concluir pela avaliação feita que existe um conjunto de defeitos que são mais facilmente identificáveis: *background, nós, bolsas de resina e brechas de resinagem.* Estes defeitos apresentam taxas de êxito maiores (maior eficiência) do que os quatro defeitos restantes. A baixa taxa de sucesso dos defeitos *SDF, CRD, AZLD e CSC* compromete a taxa de sucesso global. No que se refere à localização dos defeitos, os algoritmos comportam-se de uma forma robusta.

Um processo de reconhecimento de defeitos pré-marcados a giz é também apresentado e testado com resultados aceitáveis.

Não podendo ser, nem pretendendo ser, um trabalho exaustivo sobre técnicas de classificação de defeitos em madeira, este trabalho mostra que é possível a localização e a identificação de defeitos por via automática em pranchas de madeira. A sua viabilização passa pela definição do grau de precisão e tolerância que um sistema deste tipo possa aceitar.

A identificação do tipo de defeito constitui o passo mais difícil e decisivo, dada a incipiência manifestada na identificação de alguns defeitos. A solução deste tipo de problema poderá passar pela caracterização mais precisa desses defeitos, principalmente daqueles que forçosamente terão de ser identificados pelas propriedades de textura das suas superfícies. Se essas propriedades não se basearem na forma, a caracterização terá de assentar em características de imagem, rigorosamente extraídas nas mesmas condições de controlo externo.

Os defeitos que procurámos identificar tinham todos a característica comum de serem mais escuros do que o fundo de madeira onde se implantavam. A procura duma identificação para outro tipo de defeitos, caracterizados por possuirem uma cor mais clara que o fundo, como no caso de certas podridões brancas, exigiria adaptações nos algoritmos, e por certo nas metodologias, pelo menos nos casos que envolvessem textura e morfologia diferentes.

Esta dissertação sobre o *problema da localização e reconhecimento de defeitos em madeira de pinho por visão computacional* não fornece soluções definitivas sobre o assunto, nem pretende encerrar qualquer discussão sobre a matéria. Antes pelo contrário, procura dar um contributo inicial para um debate, que se admite profícuo e que se pensa possa trazer desenvolvimentos promissores.

Notação

Todas as variáveis ou grandezas referidas no texto, quer se tratem de vectores, matrizes, operadores ou valores estatísticos, estão definidas nos lugares em que são referenciadas. A lista apresentada, visa fornecer um elemento de consulta adicional, para as grandezas que surgem no texto com maior frequência.. Existem índices, como i, j, x, y, n, k, que podem assumir significados diferentes em função do capítulo onde são apresentados. A coerência é mantida dentro do mesmo capítulo.

Refira-se que os vectores são representados no texto em negrito e itálico e as suas componentes apenas em itálico. O expoente ^{T} está utilizado para refereenciar um vector transposto. As matrizes são representadas por maiúsculas, em negrito, e os valores escalares por caracteres minúsculos e normais.

Símbolo	*Descrição*
$f_c(x,y)$	Função continua de imagem; x,y coordenadas de imagem
$f(i,j)$	Função discreta de imagem, de duas variáveis i e j
$g(i,j)$	Função discreta de imagem binarizada, de duas variáveis i e j (transformada depois de operação de convolução ou segmentação)
$\nabla f(i,j)$	Gradiente da função de imagem $f(i,j)$
$\nabla^2 f(i,j)$	Laplaciano da função de imagem $f(i,j)$
$n(i,j)$	ruído branco na imagem
k	Número de bits utilizados na quantificação da imagem digitalizada
l	Número de níveis de cinzento numa imagem
$G=\{g_0,g_1,...,g_{L-1}\}=$ $=\{0,1,...,2^k-1\}$	Conjunto dos níveis de cinzento na imagem
T , τ	Limiares ou níveis de segmentação - *"threshold"*
NxM	Dimensão de uma imagem - N linhas x M colunas
n	Número de pixels numa imagem
$I=f(i,j)$ $\forall i,j$	Imagem I
r	Nível de cinzento, num ponto (i,j) da imagem descretizada
P_{df} , $P(r)$	Probabilidade de ocorrência do nível de cinzento r na imagem - função densidade de probabilidades
$H(r)$	Função $H(r)$ - histograma [número de ocorrências de pixels com valor r na imagem]

CP_{df}, $CP_r(r)$	Valor acumulado da probabilidade de ocorrência do nível de cinzento r na imagem - função acumulada da densidade de probabilidades
$w_{m,n}$	Máscara pesada rectangular, de dimensão m x n
$t(i,j)$	Padrão definido numa imagem digitalizada
$S(m,n)$	Medida de similitude entre duas imagens
$\varphi(m,n)$	Medida de similitude normalizada entre duas imagens
$x^T=(x_1,x_2,...,x_n)$	Vector de características (ou vector de padrões)
$x^T=(x_1,x_2,...,x_n,1)$	Vector aumentado de características (ou de padrões)
x_i	Componente de um vector de características
A,B,C	Matrizes
$\|\mathbf{A}\|$	Determinante da matriz **A**
$w^T=(w_1,w_2,...,w_n)$	Vector pesado para a classe w para um problema definido com n características
$w^T=(w_1,w_2,...,w_n,w_{n+1})$	Vector pesado para a classe w para um problema definido com n características
w_k	Vector pesado da classe de padrões k
w_i	Componente de um vector pesado de uma classes de padrões
$d(\boldsymbol{x})$	Função de decisão ou descriminante
$d_i(\boldsymbol{x})$	Função de decisão ou descriminante, da classe w_i
$d_i(\boldsymbol{x})=w_i^T\boldsymbol{x}$	Função de decisão linear na forma matricial
z_k	Característica k de um protótipo, pertencente à classe w_k
$z_{ki}{}^m$	Característica i de um protótipo de padrões, número m, pertencente à classe w_k
$p(w_i)$	Probabilidade à priori de ocorrência de w_i
$p(\boldsymbol{x})=p(x_1,x_2,...,x_n)$	Função densidade de probabilidades, de multivariável, do vector $\boldsymbol{x}$
$p_i(\boldsymbol{x})=p(\boldsymbol{x}/w_i)$	Probabilidade de ocorrência de $\boldsymbol{x}$, com $\boldsymbol{x}$ provindo de w_i
$p(w_i/\boldsymbol{x})$	Probabilidade condicional da classe w_i
$\boldsymbol{m}_i$	Vector de características médias da classe w_i, dado $\boldsymbol{x}$
C_i	Matriz de covariância para a classe i
$C=E\{(\boldsymbol{x}-\boldsymbol{m})(\boldsymbol{x}-\boldsymbol{m})'\}$	Definição da matriz de covariância
$N(m_i,C_i)$	Notação abreviada para a distribuição normal (ou Gaussiana)
$r_k(\boldsymbol{x})$	Risco médio condicional associado a uma tomada de decisão
l_{ij}	Função custo de penalidade
δ_{ij}	Função delta de Kronecker: $\delta_{ij}=0$ se $i\neq j$, e $\delta_{ij}=1$ se $i=j$
σ	Desvio médio quadrático
σ^2	Variância
μ	Média
$S_{d\theta}[i,j]$	Matriz de co-ocorrência, para d e θ dados(*GLDM*)
$I_{d\theta}\ (i,j)$	Vector que exprime a frequência de ocorrência de pixels com a mesma diferença de cor, para uma determinada distância e direcção, utilizado no método *GLDM*
$\|e(i,j)\|$	Dimensão de um *"edge"*
$\angle e(i,j)$	Argumento de um *"edge"*

PARTE 1

CARACTERIZAÇÃO DE DEFEITOS NA MADEIRA. SISTEMA AUTOMÁTICO DE RECONHECIMENTO.

CAPÍTULO I

TÓPICOS PARA A CARACTERIZAÇÃO E CLASSIFICAÇÃO AUTOMÁTICA

DE

DEFEITOS EM MADEIRA

RUI BEBIANO

1 - Tópicos para a Caracterização e Classificação Automática de Defeitos em Madeira

1.1 - Introdução

As razões que podem justificar o levantamento e estudo dos defeitos em madeira podem ser variadas, mas estão ligados no caso vertente à sua localização e reconhecimento, por via automática, através da inspecção óptica de imagens. É esse o grande objectivo deste trabalho, e é por essa razão que se impõe um estudo mais aprofundado e caracterizado de cada um desses defeitos, pelo menos para os mais vulgares e frequentes na madeira. Importa salientar que esse conhecimento não é de desprezar, já que a presença de defeitos na madeira, tem reflexos nas características físicas e mecânicas das peças de madeira, de primeira e segunda laboração, com reflexos claros no valor económico do produto final. A identificação desses defeitos, quer na árvore quer nos toros, reveste-se também de especial importância no que respeita ao comércio e aproveitamento da madeira, mas a importância que lhe atribuiremos aqui será diminuta, por ultrapassar os objectivos desta dissertação.

A importância do estudo dos defeitos em madeira, principalmente naqueles aspectos que levem a uma melhor caracterização da sua forma e à caracterização do aspecto das suas superfícies, constitui a primeira tarefa a desenvolver para a tornar possível a implementação de um sistema de reconhecimento automático por computador. O estudo visa a identificação de defeitos em madeira serrada, e por isso centrará a sua atenção nesse aspecto, apesar da descrição feita no capítulo ser mais abrangente, uma vez que procura fazer uma abordagem globalizante que possa situar o problema, e fazer compreender melhor a problemática geral. É de salientar que existem defeitos fáceis de identificar(patentes na madeira), mesmo para um não especialista(nós, certas podridões, deformações do fuste,etc), e defeitos só identificáveis por análise minuciosa e especializada, o que acrescenta razões para a necessidade da sua tipificação.

De referir que no que toca às madeiras serradas, os vários tipos de defeitos identificados se encontram nas peças, em diferentes graus de probabilidade de ocorrência. Assim existem defeitos frequentes(casos dos nós e bolsas de resina, nas

resinosas) e outros menos frequentes, como por exemplo alguns tipos de podridões. Em relação à análise automática é também de salientar que defeitos do mesmo tipo, podem não aparecer exactamente com as mesmas características observáveis, e apresentarem aspecto diverso, o que pode dificultar grandemente a sua caracterização e identificação. Por outro lado, enquanto alguns defeitos podem ser facilmente identificáveis através das suas características ou medidas geométricas, outros há que exigem a caracterização da estrutura da superfície através de outros tipos de medidas, sensíveis em muito mais elevado grau, às condições de iluminação e às condições de resolução em que foi feita a aquisição da imagem.

A própria madeira em função do tipo de clima, solo e das condições de exploração em que se desenvolveu, apresenta propriedades estruturais e cromáticas variáveis, o que dificulta a identificação automática, situação agravada pelas condições topológicas e contextuais em que se enquadra o defeito, e que são completamente variáveis em função da amostra em causa.

A inspecção automática, pode fornecer um contributo decisivo para a classificação das madeiras segundo o seu grau de qualidade, não apenas em função dos defeitos contidos, mas em função da relação entre defeitos contidos e características pretendidas; dimensionais, densidade, espessuras dos aneis de crescimento, características mecânicas, húmidade, percentagem do cerne, etc, com vista a uma classificação global, relacionada com uma utilização possível, que envolveria provavelmente também aspectos e considerações de natureza estéticos.

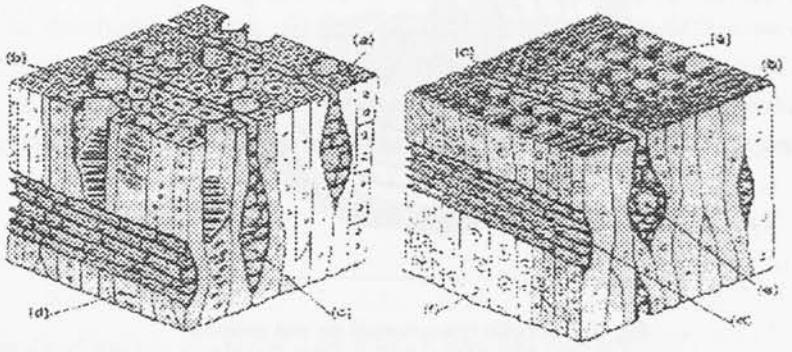
Este trabalho tem objectivos mais modestos, já que apenas se irá debruçar sobre algumas técnicas de análise e classificação de defeitos em madeira, como a localização e identificação dos mesmos nas peças serradas, dentro de um quadro limitado de defeitos a identificar. A dimensão dos problemas impõe limitações ao seu desenvolvimento, quer se olhe pelo lado do tempo que se dispõe para implementar as aplicações, quer pelo lado dos recursos, humanos e materiais, com que têm de se concretizar, e este trabalho não fugiu à regra. Por essa razão, o estudo limitará o seu campo, a um único tipo de madeira, pinho bravo - *pinus silvestris*.

Neste capítulo introdutório, faz-se uma referência à constituição estrutural da madeira e do seu tecido lenhoso, além de uma caracterização genérica da madeira como material.

Além da referência a algumas propriedades físicas e mecânicas da madeira, faz-se referência a diversos defeitos que podem afectar o seu valor como matéria prima. É este aspecto que é abordado com maior detalhe, quer na sua descrição quer na enumeração das suas causas e consequências. É feita também alguma referência à classificação de defeitos e sua normalização. O capítulo termina com uma abordagem sobre as características que deveriam acompanhar um *sistema automatizado de localização e reconhecimento de defeitos em madeira*, nos seus traços genéricos, seguida da descrição faseada de um sistema de aquisição e tratamento de imagens para o reconhecimento de defeitos em madeira.

1.2 CARACTERIZAÇÃO DA MADEIRA COMO MATERIAL

A madeira é a parte sólida situada no interior da casca dos troncos e ramos das árvores. É um tecido vegetal com forte variabilidade entre espécies e mesmo dentro da mesma espécie. As condições de crescimento, dependentes do clima, das condições ambientais e de exploração em que se desenvolve a árvore determinam o seu crescimento celular e consequentemente as suas propriedades físicas. A organização celular de uma folhosa e de uma resinosa está representada na fig. 1.1.



Organização celular numa folhosa
a)fibras
b)vasos
c)raios lenhosos
d)perfurações

Organização celular numa resinosa
a)canais em madeira de primavera
b)canais em madeira de verão
c)canal resinifero
d)raios lenhoso
e)raio lenhoso com canal resinifero
f)perfurações assegurando passagens intercelulares

fig. 1.1 Estrutura celular da madeira

Devido à presença de canais alongados ao longo das fibras, e de raios lenhosos perpendiculares a essas fibras, este material apresenta fortes propriedades anisotrópicas(características mecânicas do comportamento da madeira, em função de esforços aplicados, dependentes fundamentalmente do sentido em que actuam: longitudinal, rotacional, transversal). O tecido lenhoso é muito heterogéneo, como se pode ver na fig. ura1.1, que mostra essa organização celular, e que deixa perceeber que essa heterogeneidade celular terá implicações nas propriedades físicas do mateerial.

O crescimento é feito por aneis circulares, de periodicidade anual, com espessura fortemente influenciada pelo clima. Num corte transversal pode destinguir-se (do centro para a periferia): a medula, o núcleo, formado pelo tecido parenquimatoso situado no eixo do fuste ou do caule das plantas, com forma mais ou menos cilindrica, o cerne(de espessura variável), constituído de membranas espessas e duras, onde circula a seiva ascendente sem substâncias nutritivas, corresponde a uma madeira mais clara e de inferior qualidade. São impregnadas de resina(resinosas) ou goma(folhosas) que asseguram a conservação da árvore.

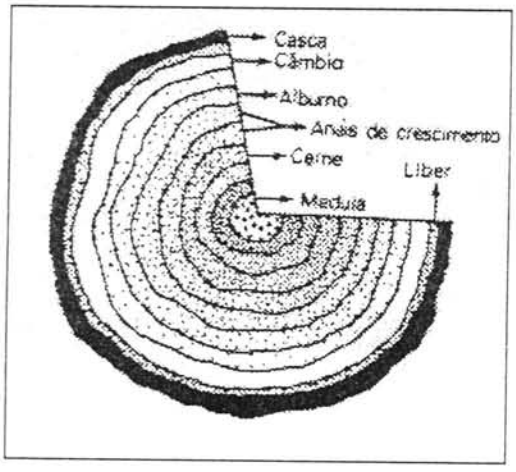


fig. 1.2 Secção transversal de um tronco

O alburno é constituído por células vivas, activas e cheias de matéria nutritiva. A fiada geradora de madeira, responsável pelo crescimento, é constituída pela entrecasca-câmbio, que cria as novas células para o exterior e as liga ao cerne. O liber é a parte onde circula a seiva descendente e dá origem à casca, sendo esta constituída por células mortas, elásticas e impregnadas de substâncias impremeáveis, asseguram a protecção e as trocas gasosas com a atmosfera. Os raios lenhosos(invisíveis nas resinosas) asseguram a circulação de substâncias nutritivas(na entrecasca). As células estão vivas na entrecasca e encontram-se mortas no cerne.

As características mecânicas da madeira (por exemplo; os coeficientes de elasticidade) estão muito dependentes de certos parâmetros físicos, tais como: densidade, teor de húmidade, etc. É de referir que a madeira devido à sua porosidade absorve muito a húmidade, que pode ir desde a saturação-200% a humidades intermédias-30%. A sua utilização está contudo interdita a teores de húmidade superiores a 15%, tendo de ser seca nestes casos.

A húmidade pode ser medida por $H\% = (M_H - M_0) / M_0$ onde M_H é a massa da amostra com húmidade H. Consoante o tipo de utilização, os graus de húmidade podem ser mais ou menos tolerados.

Em consequência das diferenças existentes entre os coeficientes de retracção, transversal(r_T), radial(r_R) e axial (r_L), o processo de secagem é sensível às condições em que é efectuada, sem o que, as probabilidades de empeno serão elevadas.

A massa volúmica da madeira varia de 0,1 a 1,5g/m^3 consoante a espécie e em função do grau de húmidade. A densidade pode ser medida pelo quociente $\rho_{15} = M_{15} / V_{15}$ para a referência de 15º de húmidade. O peso específico da matéria lenhosa é de 1,54g/cm^3 qualquer que seja a essência. Se for comparado com a massa volúmica, a diferença permitirá conhecer a porosidade da madeira.

A madeira contém sob a forma de constituintes elementares, 50% de carbono, 43% de oxigénio, 6% de hidrogénio, 1% de azoto, e uma quantidade variável de cálcio, fosfato, silício e potássio(0,1% a 0,3%).

Além da água que a madeira contém, as paredes das células são em celulose $(C_6H_{10}O_5)_n$, combinadas com lenhina, que constituem os compostos mais frequentes da composição da madeira. Nas células mortas do cerne, encontram-se substâncias azotadas, taninos e resinas. A celulose é a molécula mais abundante(40% a 50%) e apresenta-se na forma de microfibras, com um grau de cristalinidade muito grande. A lenhina é um polimero complexo de natureza fenólica(20% a 30% da constituição).

1. 3 GENERALIDADES SOBRE DEFEITOS EM MADEIRA

O estudo dos defeitos em madeira assume especial importância se isso for visto dentro de um quadro de controlo de qualidade, na esfera da produção de artefactos e peças em madeira.

Os defeitos em madeira resultam de particularidades morfológicas e estruturais das árvores e do deficiente estabelecimento e crescimento do povoamento florestal. Os defeitos ocorrem por isso fundamentalmente durante a fase de formação da madeira, mas também têm a ver com as condições de exploração das matas florestais e com as sequentes operações de conversão da madeira(serragem, serração, laboração)[1].

A outra vertente diz respeito às consequências dessas anomalias, ou defeitos. Segundo as Normas Francesas NFB 50002, CES-84 e CTB - 84, essas acarretam

[1]Entenda-se por serragem, os cortes destinados ao abate da àrvore nas matas, serração o corte em tábuas ou pranchas e laboração todas as operações de transformação da madeira, numa fase e posterior às primeiras.

modificações físicas ou mecânicas na madeira, afectando as condições em que se efectua a laboração das peças. Os nós afectam a dureza da peça de uma forma irregular, bem como o fio torcido, e desse modo as condições com que irão ser trabalhadas. As bolsas de resina travam a laboração das máquinas e prejudicam as qualidades de corte das ferramentas, isto apenas para focar alguns exemplos.

Com este capítulo, procura-se aflorar simplesmente abordar duma forma sucinta, os principais aspectos ligados aos defeitos nas madeiras relativos à sua morfologia (ligada à identificação desses defeitos) mais do que à sua natureza e formação.

Qualquer classificação referente à qualidade da madeira como matéria prima, assenta na quantidade, extensão e tipo de defeitos, que as peças de madeira possam conter.

Nneste trabalho a abordagem que será feita procura apenas com base nas características identificadoras de cada um dos defeitos propostos, estabelecer por via automática, a detecção, localização e identificação dos defeitos presentes nas peças.

O estudo dos defeitos em madeira constitui assim um capítulo vasto e importante, já que não limita o seu interesse apenas à fase da produção, mas envolve todo o ciclo biológico da árvore, e as diversas fases de aproveitamento.

Os defeitos afectam a capacidade de utilização do material, dificultam a sua laboração e aumentam o volume de desperdício. Durante os trabalhos de abate, extracção, transporte, e a consequente conversão primária(na secagem, na conservação e na laboração propriamente dita), o material lenhoso em cada fase de transformação, fica sujeito a depreciações várias, e por isso interessará caracteriza-los.

Segundo Albino de Carvalho [2-A], por *DEFEITO* em madeira entende-se: "Toda e qualquer irregularidade, descontinuidade ou anomalia estrutural, alteração química ou de coloração, modificação morfológica do fuste ou das peças, originada durante a vida da árvore, na exploração e transporte da madeira, na conversão primária, na secagem, na preparação e noutras operações tecnológicas, sempre que qualquer um desses aspectos comprometa o valor intrínseco da madeira".

Albino de Carvalho [2-A] propõe mesmo uma classificação agrupando os defeitos nos seguintes tipos:

A ***Defeito físico*** - Deformidade de estrutura ou particularidade da morfologia da árvore, que deprecia normalmente uma parte do fuste ou da peça de madeira Ex: nós, bolsas de resina, fissura de compressão e de tracção, fenda de geladura, etc]

B Anomalia - Irregularidade estrutural, algumas vezes congénita, que em regra afecta a maior parte do fuste ou da peça de madeira [Ex: fio torcido, fio revesso, medula excêntrica, anel desigual, etc].

C Alteração - Modificação mais ou menos complexa do lenho(física, química, ou estrutural), de vária natureza (fisiológica ou biológica) [Ex: mancha, azulado, podridão, madeira carunchosa, etc.].

D Deformação - Desvio morfológico do fuste ou modificação anormal das peças serradas[Ex: fuste torto, fuste espiralado, fuste escasso, empenos, etc.].

E Defeitos de exploração - Resultante da imperfeita condução das operações de abate, extracção e de transporte[Ex: fenda de abate, fissura de abate, cavidade de abate, etc)

F Defeitos de secagem e de laboração - Provenientes da má realização de operações de conversão e de secagem(Ex: fendas de secagem, assado de estufa, descaio, desvio de dimensões, ressaltos, riscos de serra, etc.).

A norma NP-180(1962)[2], sobre anomalias e defeitos em madeira, fixxa como objectivo estabelecer a terminologia das anomalias e defeitos em madeira de obra e construção. Para isso divide os defeitos em quatro tipos:

A-Anomalias e defeitos relacionados com a estrutura do lenho ou com particularidades da morfologia da árvore.

B-Anomalias e defeitos resultantes de práticas culturais, de acidentes meteorológicos ou de outras influências externas.

C-Anomalias e defeitos devidos ao ataque de fungos ou de animais xilófagos

D-Anomalias e defeitos devidos ao abate, à secagem e à laboração.

A norma aborda também as causas e as consequências ligadas a cada um dos defeitos referenciados.

Os defeitos referenciados por *'A'* estão ligados fundamentalmente às diliferentes formas de distribuição das fibras (lenhina) e à forma de crescimento mais ou menos

[2]NP-180, Direcção Geral da Qualidade(DGQ),"ANOMALIAS E DEFEITOS EM MADEIRA", 1962.

rápida dos aneis (das células), ou seja às condições em que foi formado o lenho. Como defeitos de origem genética, podem ser citados os defeitos de conformação do fuste: Fuste torto, fuste escasso, fuste espiralado. Ainda nos defeitos morfológicos podem ser citados entre outros os nós e o fuste rajudo.

Como defeitos de estrutura também de natureza genética, podem ser citados, entre outros o fio torcido e o fio revesso, que podem ser identificados pela disposição dos veios, mas também da cor e aspecto. Neste grupo ainda se distinguem as *inclusões minerais* (depósitos minerais nas células) e a *soldadura* provocada pela união de dois fustes que se interligam, e que eliminam ou não, a casca intermédia.

Os defeitos classificados em *'B'* revelam a influência da técnica silvícola, que se reflecte imenso na qualidade da madeira obtida. Manifestam-se tanto nos desvios como na proporção ou no tipo de elementos constitutivos do lenho e nas particularidades morfológicas; nós, fuste torto, fuste de medula excêntrica, feridas de resinagem, etc. A madeira proveniente das árvores mortas que se mantiveram de pé sem serem cortadas, também pode ser englobada nesta categoria. Esta madeira caracteriza-se por ser mais leve, apresentar uma menor tenacidade, durabilidade, e retractibilidade, ou seja caracterizada pela sua fragilidade.

As inclusões de resina (nas suas diversas formas, bolsa de resina, fenda de resina, risca de resina) que são acumulações de resina, em consequência de traumatismos externos (vento, sol, peso, curvatura, solicitações mecânicas, etc.) acarretam dificuldades acrescidas de laboração, redução da resistência e aumento dos desperdícios de laboração. As acumulações de goma ou látex, são o equivalente para as folhosas, às acumulações de resina para as resinosas.

As inclusões de casca, inclusões de corpos estranhos (corpo incluso) e cicatrizes(reacção de protecção da árvore para proteger a parte do lenho descoberto) são outros tantos defeitos ocasionados por intervenção e acção externa sobre a floresta.

A categoria *'C'*, trata os defeitos de natureza biótica, que compreende os efeitos originados pela acção de seres vivos (fungos e insectos) que aproveitam a árvore como hospedeiro ou local de refúgio. Neste grupo de depreciações incluem-se as podridões, os ataques de insectos, os tumores e cancros, entre outros.

No que toca aos ataques por microrganismos, há a distinguir diversos tipos de defeitos que depreciam geralmente o valor da madeira, de que se dá exemplo, as manchas, os azulados, e podridões diversas

Na categoria *'D'*, os defeitos de exploração dizem respeito ao abate(fracturas de abate e cavidade de abate) à rechegagem (fendas para extracção de resina), extracção e conservação na mata (que podem acarretar o aparecimento de azulados e fendimento terminal) e ao transporte.

Os defeitos de secagem ocorrem durante a dessecação (enxugo) do lenho. Os defeitos devidos à secagem que mais frequentemente se registam são as fendas de secagem (rachas e fendimentos de secagem), empenos (arco, meia-cana, , aduela, hélice), colapso, assado, queimado de estufa, etc.

Os defeitos provenientes da conversão e laboração da madeira, podem n levar à perda de qualidade das peças devido a falhas, descaio, desvio de dimensões,s, desvio de corte, ressaltos, risco de serra, rugosidades.

1.4 DEFEITOS PRINCIPAIS NAS MADEIRAS

Procura-se neste ponto do capítulo dar uma ideia genérica (de n natureza descritiva) dos principais defeitos das madeiras serradas e das consequências principais desses defeitos na laboração. As causas não serão abordadas, já que não é objectivo do trabalho abordar aspectos relativos à exploração de florestas.

A definição e caracterização de cada um dos defeitos, segue de perto o que está estabelecido na Norma Portuguesa NP-180[2-c] já referida, e na norma NP-987[3] no que se refere à forma de fazer a medição desses defeitos [apenas abordada a de uma forma marginal neste estudo].

> 1 *NÓ* - Os nós resultam das ramificações da árvore, e por essa razão existem sempre em peças de madeira de grande dimensão. Contudo o seu tamanho e número, o seu tipo de inserção, a sua forma de agrupamento, o seu estado sanitário variam, com a espécie, com as características do clima, com o tratamento ministrado à árvore e ao povoamento. Apresenta-se a seguinte definição de nó:
>
> *definição*: "Porções de ramo inclusos nas peças de madeira e que produzem um desvio local das fibras. Bem visíveis exteriormente,..."[4].

Os nós são os defeitos mais graves que a madeira pode apresentar, já que e têm implicações de ordem física, mecânica, estética e de ordem tecnológica, c com influência consequente no valor da peça.

---

[3]NP-987, Direcção Geral da Qualidade(DGQ),"MADEIRAS SERRADAS-Medição de Defeitos", 1973

[4]NP-180, Direcção Geral da Qualidade(DGQ),"ANOMALIAS E DEFEITOS EM MADEIRA", 1962.

Para a madeira onde estão inseridos, interessa considerar as suas implicações físicas, já que são responsáveis por grande parte dos empenos da madeira. Os nós sendo mais densos e tendo orientação perpendicular aos tecidos circundantes fazem ainda, devido às tensões que suportam, que apareçam fendas nos nós fixos e se desprendam nos nós saltadiços.

As consequências de ordem mecânica, referem-se à eliminação da resistência das peças, para o que contribui grandemente, não só a existência dos nós mas também a sua forma, tamanho, localização e inserção do mesmo. É também de salientar que o tamanho e o tipo de peça em que está presente, tem igualmente importância recíproca na resistência da mesma.

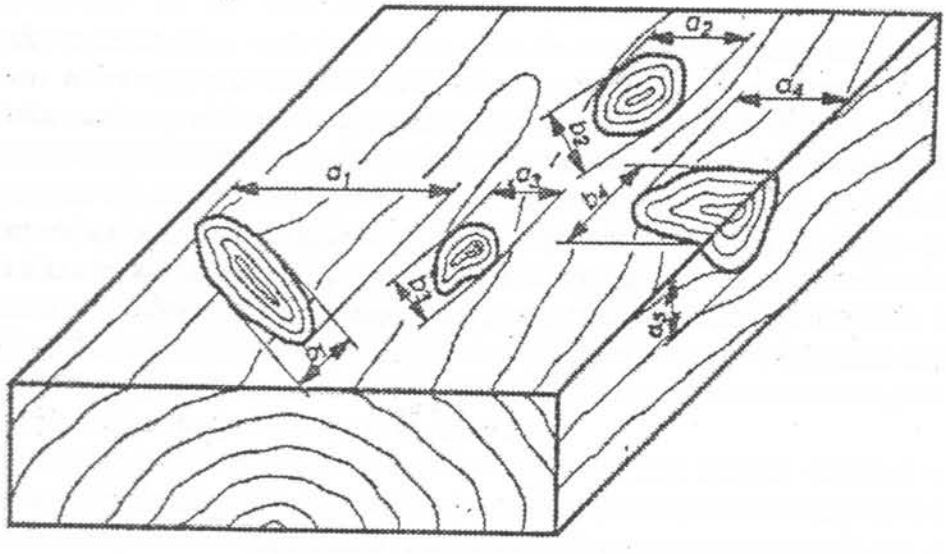


fig. 1.3 Nós agrupados e suas medições

A influência maior faz-se sentir em relação aos nós redondos (que atravessam geralmente as pranchas de madeira) e menos nos nós de forma elíptica ou oval, e ainda menos acentuada nos nós deitados. Os nós alteram a direcção do fio da madeira na sua vizinhança e com frequência fazem aparecer fendas e rachas nos mesmos, além de dificultarem os trabalhos de acabamento das peças. A localização dos nós é também determinante, já que as implicações são graves nos nós de aresta, nós repassados e nós deitados. Os nós firmes e de pequena dimensão podem contudo dar valor ornamental à peça. O estado sanitário do nó pode também retirar valor às peças de madeira com esse defeito. A gravidade deste defeitos nas peças serradas, dependem do tipo, da dimensão, do número, agrupamento, localização nas peças e estado sanitário das mesmas.

Os nós podem ser classificados quanto:

à inserção; em nó firme(ou nó aderente) ou em nó saltadiço, conforme está ou não perfeitamente ligado ao tecido lenhoso.
à forma; em nó circular, elíptico, ou deitado(comprido) em que este último apresenta na superfície serrada contorno alongado pelo facto de a secção se fazer longitudinalmente.

à sanidade do lenho; em nó são ou nó podre, conforme está ou não atacado por agentes biológicos(fungos, ou outros), nó fendido (rachado) com fendas, e nó lascado, incompleto em resultado de laboração das peças.

à localização nas peças; em nó de aresta, nó de face, nó de canto, nó repassado(nó que atravessa uma peça e aparece nas duas superfícies de corte opostas) e nó incluso(nó que não aparece à superfície da peça).

à dimensão; nó muito grande, nó médio, nó pequeno, nó muito pequeno. Para determinar a dimensão dos nós (representada na fig. 1.3) é necessário atender às seguintes condições[5]:

a) Medir a distância entre tangentes ao contorno do nó, paralelas às arestas longitudinais da peça.

b) Medir o eixo menor da secção do nó.

2 - ***Fibras torcidas***; As fibras dispõem-se normalmente paralelas ao eixo da árvore. Contudo, por vezes devido à acção do vento ou de outros agentes, encontram-se inclinados ou enrolados em hélice. A madeira com as fibras torcidas é difícil de trabalhar, tem menor resistência mecânica e não apresenta desenho uniforme. São identificadas pelo fio diagonal e por fendas superficiais que acompanham o alinhamento dos elementos fibrosos.

3 - ***Fendas*** A madeira fendilha com facilidade, fendilhamento que se dá paralelamente às fibras, sendo as mais vulgares as do tipo

[5]No texto apenas se faz uma referência sucinta à forma de medir os diversos tipos de nós. Para um estudo mais aprofundado recomenda-se a consulta da norma NP-987.

radial. Podem ser provocadas como já foi dito, por abaixamento brusco de temperatura, deslocamento dos aneis de crescimento, por choque e esforços mecânicos, vento ou na queda da árvore, ou na fase de secagem. Segundo NP-180 fenda é definida como "soluções continuidade no lenho segundo superfícies longitudinais", e distingue vários tipos de fendas como: fenda radial(segundo plano radial do lenho), periféricas(externa), central(que parte do interior e não atinge o exterior do tronco), em pé de galinha, em quadrante(um agrupamento de várias), anelar(em forma de meia lua)

As fendas de secagem partem da periferia para o centro, e são devidas a fenómenos de ruptura segundo planos longitudinais, e surgem em consequência de tensões internas atribuídas a diferenças de retracção durante a secagem. Patente por deslocamento total ou parcial das camadas de crescimento.

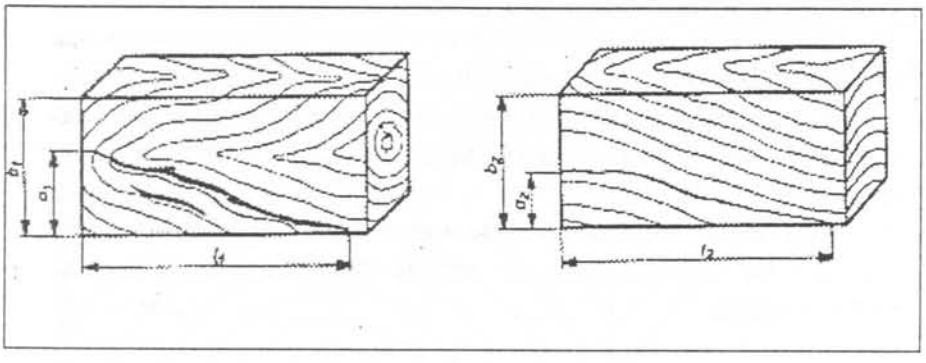


fig. 1.4 Defeitos em madeira-Fibras torcidas

4 - ***Racha*** Fenda de secagem profunda. Quando as fendas se localizam nos topos e se prolongam de uma face à outra da peça, tomam o nome de racha.

5 - ***Fissuras*** "São soluções de continuidade nos tecidos lenhosos segundo superfícies mais ou menos transversais". Resultam geralmente de traumatismos violentos que provocam a brusca encurvadura dos fustes. As fissuras de compressão, é uma "marca transversal causada por esmagamento, resultante de esforços de compressão axial, sem descontinuidade aparente dos tecidos", geralmente atribuído este defeito ao abate mal feito.

Nas peças serradas, as fissuras de compressão aparecem como linhas quebradas de cor clara, dispostas mais ou menos perpendicularmente ao fio, rodeada de tecido ofendido de cor

mais escura, devida a uma extensa mancha de resina que acompanha a lesão. Patente nas peças serradas por linha mais ou menos perpendicular ao fio, algumas vezes rodeada de manchas escuras de resina. As fissuras de tracção são patentes nas peças serradas, por fendas tangenciais e cicatrizes.

6 - ***Manchas*** Alterações mais ou menos extensas da cor do lenho. Os fungos que provocam apenas alterações na cor do lenho são chamados fungos cromogéneos. São defeitos, patentes nas peças serradas por alteração cromática, definida por zonas de contorno irregular e assimétrico. O azulado é uma mancha deste tipo, caracterizada pela sua cor cinzento-azulado, ou mesmo quase negras.

7 - ***Podridões*** Os fungos que provocam podridões são chamados xilófogos. Antes da fase de podridão estes fungos fazem sempre aparecer uma mancha. Podridão pode ser definida como "decomposição dos tecidos lenhosos pela acção dos fungos, com maior ou menor alteração da cor e aspecto, e diminuição das qualidades físicas e mecânicas"[2].

As podridões podem ser classificadas quanto à cor (castanho - ataca a celulose ou branca-ataca a lenhina); quanto ao grau de ataque (incipiente, avançada, completa); quanto ao aspecto (cúbica, alveolar, fibrosa, lamelar) e quanto à húmidade(seca e húmida). Estes defeitos são patentes nas peças serradas pela alteração da cor, do cheiro, da consistência, da dureza, da densidade e da estrutura(infelizmente por meios ópticos só a alteração da estrutura e da cor pode ser testada).

8 - ***Ardido e cardido*** "Fase incipiente da podridão branca devida ao ataque da *Trametes Pini (ou Xanthocrous pini)* no cerne que toma uma cor avermelhada", para o ardido, enquanto que o cardido "é uma forma de podridão branca, alveolar, seca e avançada do cerne do pinheiro". O cardido não progride depois do abate da madeira. O cardido é muitas vezes utilizado para designar outras formas de podridão. O ardido é visível pela alteração da cor, de vermelho tijolo, enquanto no cardido além da forte alteração da cor, nota-se alteração profunda da estrutura da madeira, com aparecimento de alvéolos revestidos de feltro branco. Em estados mais avançados podem aparecer cavernas provenientes da desordem do lenho.

9 - ***Madeira carunchosa*** madeira com galerias causadas por insectos xilófagos e geralmente cheias de serrim.

10 - ***Feridas de resinagem*** Este defeito é uma consequência da incisão que é necessário fazer na casca do pinheiro para provocar a exsudação da gema. Quando as feridas são extensas, constituem defeito grave, que favorece o cardido e a perda de resistência da madeira.

11 - ***Inclusões de resina*** Acumulações anormais de resina no estado sólido ou liquido nos tecidos lenhosos. Podemos distinguir vários tipos - Bolsa de resina (espaço lenticular formada na zona inicial duma camada de crescimento cheia de resina) - fenda de resina (fenda radial cheia de resina), riscas de resina, e mancha de resina ("acumulação localizada de resina nos tecidos lenhosos, com forma de nódoa irregular mais ou menos extensa").

As inclusões de goma, são para as folhosas, o defeito eqivalente às bolsas de resina, nas resinosas. Assemelham-se a pequenas fendas tangenciais, ou "facadas" mais ou menos cheias de goma ou gema, frequentemente exsuda.

12 - ***Inclusão de casca*** *(casca inclusa)* Porção de casca completamente envolvida pelo tecido lenhoso.

13 - ***Inclusão de corpos estranhos*** *(corpo estranho incluso)* Caracterizado pela presença de corpos estranhos nos tecidos lenhosos e por eles recobertos, como por exemplo pregos, parafusos, projecteis, etc.).

14 - ***Cicatriz*** Formação que se desenvolve para proteger uma parte do lenho que foi descoberto devido a um traumatismo qualquer.

15 - ***Falha*** "Defeito de laboração caracterizado pela falha da madeira, nas arestas, faces ou extremidades das peças".

16 - ***Descaio*** *(madeira falheira)* "Parte remanescente da superfície de um toro numa peça de madeira serrada. "

17 - ***Falhas de plaina*** "Manchas ásperas na superfície aplainada, pelo facto de os ferros da plaina não terem actuado nesssas zonas".

18 - ***Empenos*** Encurvamentos nas peças de madeira devido a retracção. Podem ser distinguidos vários tipos de empenos, como em meia cana em aduela, e em hélice.

19 - ***Medula*** A medula não constitui um defeito propriamente dito mas comporta-se como tal, já que constitui motivo de depreciação ou de desvalorização das peças. A medula é o núcleo mais ou menos

cilindrico do tecido parenquimatoso situado no eixo do fuste ou do caule das plantas. Contribui para o enfraquecimento das características mecânicas das peças que o contêm, já que é formada por um tecido de fraca coesão.

A acção enfraquecedora da medula tem a ver com a proporção do cilindro central relativamente ao lenho que forma a peça. O lenho juvenil que o envolve(lenho de crescimento longo) contribui para a pouca resistência dessas peças.

1.5 Objectivos e Organização de um Sistema de Visão, para Detecção de Defeitos e Classificação de Madeira

Segundo Ciccotelli[4-B] e Bourdier[3-B], os objectivos de um sistema de detecção automática prendem-se com o aumento da velocidade de produção e com a diminuição dos custos de produção(menos desperdícios, menos mão de obra, maiores velocidades de laboração, melhor apresentação da matéria prima).

As peças de madeira devem satisfazer as normas de classificação e respeitar as regras de comercialização. Os principais parâmetros a ter em conta são a geometria das peças, húmidade, características mecânicas, defeitos e alterações. Para essa classificação, além do aspecto, contam portanto as características físicas e mecânicas(como densidade, módulo de ruptura, elasticidade, etc.) além das singularidades da estrutura(nós, bolsas de resina, entrecasca, fendas, . . .), estado sanitário, bem como as suas dimensões e acabamento.

Um sistema que permita este desempenho deverá ser automatizado e operar on-line, em tempo real, nas condições de velocidade adaptada às características de laboração.

Deverá ter as características de laboração de um sistema automatizado que, segundo Patrick Martin[24-A], deve obedecer entre outras, às seguintes condições:

1 - Funcionamento fiável e uniforme
2 - Integração fácil numa cadeia de fabrico

3 - Adaptar-se à exploração em tempo real
4 - Adaptação rápida às mudanças das características externas e internas da madeira
5 - Não interferir nas características físicas da madeira
6 - Independência em relação ao meio envolvente
7 - Não introduza riscos para os operadores (ex: radiação)
8 - Custos reduzidos e facilidade na sua manutenção.

Os sensores utilizados num sistema de detecção automático, podem ser de vários tipos, mas os mais usados são os ultra-sónicos (a velocidade de propagação dos ultra sons é afectada pelas características físicas da madeira) que medem o tempo que o som demora a propagar-se através da madeira, desde o instante em que é emitido e o momento em que é recebido. Outros baseiam-se na propagação de ondas de natureza electromagnética através da madeira, desde as bandas visíveis aos infravermelhos, dos ultravioletas aos raios γ, dos raios χ às microondas. É que as características físicas da madeira, afectam também as condições de propagação (coeficientes de transmissão) das ondas electromagnéticas(constante dieléctrica por exemplo), valores esses que podem ser testados e medidos em cada uma das condições. A comparação dos resultados com um protótipo permite inferir da existência ou não de defeito e num segundo momento, ser capaz de especificar o tipo.

Um sistema global de classificação poderia portanto, integrar no processo de classificação sensores de vários tipos, de modo a obter a máxima informação necessária, de forma a ser possível integra-la num sistema de conhecimento globalizante e geral.

Os sistemas ópticos também são utilizados, mas estão limitados à medição dos defeitos visíveis. A identificação destes defeitos por meios ópticos vai depender do material em observação, da luminância medida, da câmara de captação e do tratamento dado aos sinais recebidos. Da caracterização conveniente dos defeitos, irá depender em grande parte, o êxito da operação de reconhecimento. Essa tarefa não é porém das mais fáceis mesmo que se trabalhe apenas com uma única espécie vegetal, dado que na madeira, defeitos do mesmo tipo em amostras diferentes, podem apresentar níveis cromáticos, forma e textura das suas superfícies de segmentação com valores muito dispares, tornando difícil organizar um conjunto de características consistente, que possa constituir a base de um modelo de classificação de madeira perfeitamente fiável na identificação de cada defeito.

Podemos ver mais de perto como se organiza um sistema de visão óptica por computador, dado ser esse o meio que vamos utilizar na captação de imagens,

utilizadas no sistema de análise e classificação de defeitos em madeira de pinho-"*pinus silvestris*".

A variação da quantidade de luz reflectida na superfície de uma prancha de madeira, é o ponto de partida das medições ópticas a efectuar. A imagem obtida por reflexão depende da iluminação da superfície e das características de reflexão da superfície e da câmara utilizada. A energia duma radiação determina a luminância emitida.

No caso da madeira, a análise espectral, permite ver que a reflexão aumenta com o comprimento de onda. O estado da superfície tem também grande influência sobre as características de reflexão e por essa razão a madeira aplainada permite condições de reflexão muito mais estáveis do que a madeira não aplainada. A iluminação dos objectos deve ser conseguida de uma forma uniforme e constante, e a luz ambiente não deve influenciar a iluminação do objecto. O captor pode ser uma câmara de vídeo de varrimento de linhas(reticon) ou mais usualmente uma a câmara matricial de diodos fotosensiveis(CCD), sensível a 256 níveis de cinzento, ou a cores(RGB).

Depois de efectuada a captação da imagem é feito o tratamento da imagem, constituído por filtragens, segmentação, fecho dos contornos, extracção de características, e partição em categorias. Como em geral os defeitos são mais escuros que a madeira sã, estes podem ser localizados e analisados pela brilho das superfícies, pela forma/dimensão e textura da superfície. A utilização de análise de texturas, desde que em condições de iluminação controlada, pode melhorar os resultados da classificação. Na posse de um vector de características procede-se à classificação num dos grupos de defeitos em que foi dividido o problema.

A rapidez dos cálculos, num tratamento de imagem em tempo real, é fundamental, o que nem sempre é fácil, já que mesmo para imagens pequenas, isso implica ter de tratar dezenas ou centenas de milhar de pixels, o que prefaz sempre milhões de operações em cada processo.

1.6 Conclusões

Procurou fazer-se neste capítulo, uma síntese do que poderiam ser os objectivos e as principais dificuldades encontradas, num *sistema de localização e identificação de defeitos* em pranchas de madeira de pinho serradas.

Para tal fez-se a caracterização da madeira como material e a algumas referências à sua estrutura e composição. Aflorou-se também a influência de determinados parâmetros físicos nas características mecânicas do material.

Uma descrição e caracterização mais aprofundada sobre defeitos na madeira, suas causas e suas implicações nas características físicas e mecânicas do material,

preenche boa parte do capítulo, já que o objectivo central da dissertação se prende com o *reconhecimento automático de defeitos em madeira.*

O capítulo termina com uma descrição genérica das características e organização funcional, que um sistemas de visão automatizado por computador deveria possuir. Esse sistema teria como finalidade última, além da produção automatizada, a melhoria da qualidade do produto final.

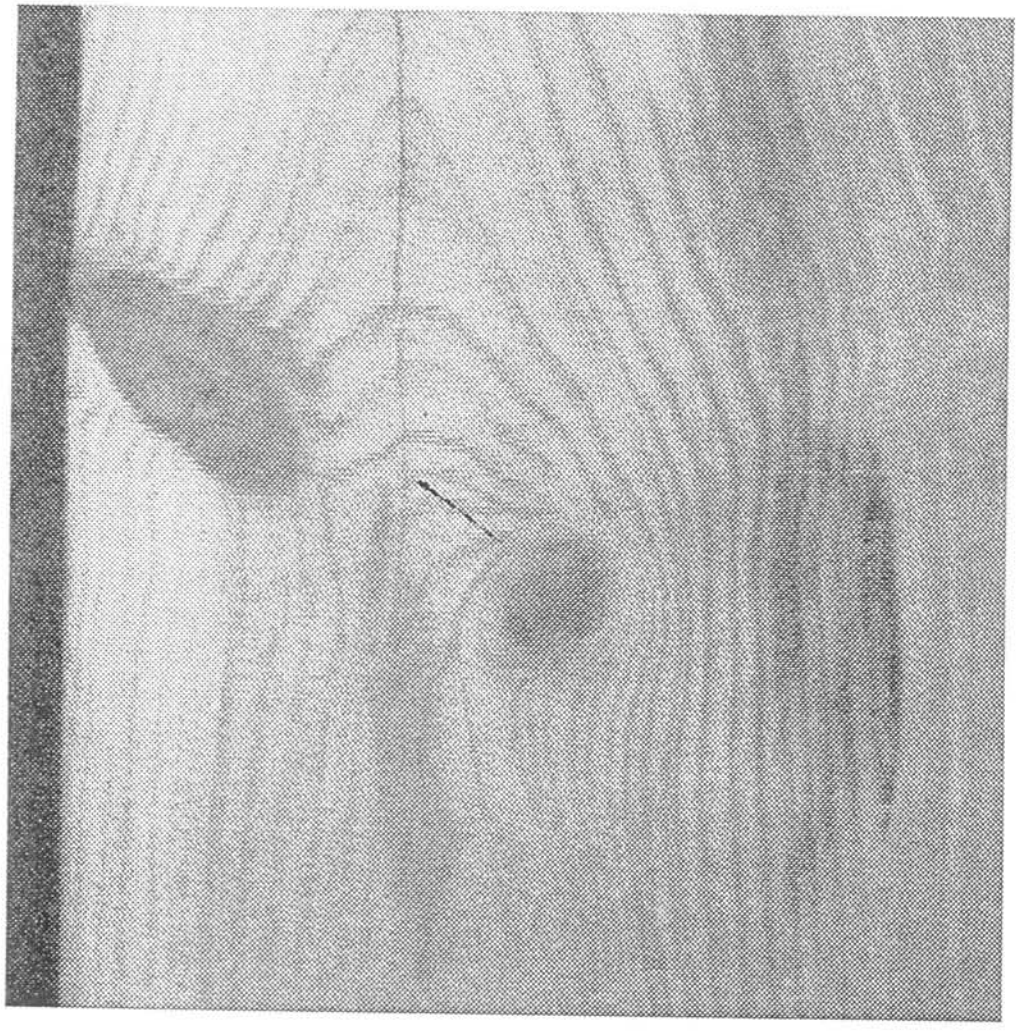


fig. 1.5 Imagem de prancha de pinho com diversos defeitos.
Nó, fissura, background bolsa de resina e queimado da plaina

PARTE 2

TRATAMENTO DE IMAGEM

CAPÍTULO II

PROCESSAMENTO DE IMAGEM

TRANSFORMAÇÕES NO DOMÍNIO ESPACIAL

2 - Processamento de Imagem. Transformações no Domínio Espacial.

2.1 Introdução

Neste capítulo procura-se fazer uma abordagem genérica e não exaustiva, de diversas técnicas utilizadas na transformação e tratamento de imagem, geralmente utilizadas com a finalidade de deixar as imagens em condições mais favoráveis aos objectivos das manipulações sequentes. Esses objectivos podem ser relativos a análises a efectuar sobre a imagem, ou prender-se com a sua classificação total ou parcial.

Não sendo sempre possível organizar, todos os assuntos, em subcapítulos estanques, já que muitas das técnicas envolvendo processos de melhoria de imagem se interpenetram, organizámos este capítulo segundo tópicos descritos da forma que se segue:

1 - Transformações determinísticas dos níveis de cinzento
2 - Modificações do histograma
3 - Realce de imagens (acentuação de '*edges*')
4 - Eliminação do ruído
5 - Filtragens

Com as técnicas de melhoria de imagem, procura-se através de prorocessos diversos, obter outras imagens que apresentem melhores características para a máquina com vista à sua análise, ou para o utilizador através de uma melhor aparência e definição. Nessa última perspectiva, uma imagem destorcida pode-se

apresentar mais interessante do que a imagem ao natural. Um exemplo disso seria o caso duma imagem com orlas realçadas, com distorção de cor ou contraste exagerado o que pode permitir em certas circunstâncias que algumas características sejam postas em evidência na imagem, realçando desse modo aspectos não visiveis da imagem normal.

O processamento digital de imagem visa assim a transformação da imagem, através de operações geométricas, matemáticas, estatísticas, ou outras, com vista à obtenção de certos objectivos, que podem ser: o aumento do contraste, a transformação da cor na imagem, alteração do realce das orlas, etc. Entendida duma forma mais ampla, pode ser vista como a descrição e interpretação automática de cenas/ou imagens (Pratt[26-A]).

Em resultado das operações efectuadas sobre as imagens, podem ser obtidas outras imagens, ou descrições quantitativas ou interpretativas dessas imagens. No primeiro caso, estamos perante operações de processamento digital de imagem (ex: realce de imagens, manipulação do histograma, etc.) e no segundo caso no domínio do reconhecimento de padrões em imagem e na computação visual [Computer Vision]. Se neste segundo caso o objectivo for mesmo a interpretação de imagens, estaremos no domínio interpretativo de visão através da máquina [Machine Vision].

Em resumo, o capítulo II descreve alguns processos utilizados na melhoria de imagem, sempre com o objectivo de colocar em evidência determinadas características úteis à sua análise posterior. As operações de realce, transformações de histograma e filtragens são as mais interessantes do ponto de vista do objectivo da dissertação, e por esse motivo as mais citadas. Essas operações são utilizadas no processo global de *localização e identificação de defeitos em madeira*. Este capítulo faz ainda uma introdução à forma como pode ser estruturado um processo de reconhecimento e às etapas nisso envolvidas.

Finalmente uma referência ao domínio em que serão operados os processamentos de imagem. As operações de processamento de imagem podem ser divididas em duas categorias: processamento espacial e processamento através de transformada. No processamento por transformada as imagens são primeiro convertidas para o domínio da transformada e só depois é que é feito o processamento. Processos de transformação inversa são posteriormente utilizados para obter novamente a imagem no domínio espacial.

Contudo por razões de natureza diversa, mas principalmente pelo tempo elevado que os algoritmos de processamento desses programas no domínio da frequência espacial levavam, a abordagem feita no capítulo e duma forma geral a orientação em toda a dissertação, apenas se fará baseada em técnicas do domínio espacial, técnicas em que o *pixel* na imagem é manipulado directamente.

2.2 FASES ENVOLVIDAS NUM PROCESSO DE RECONHECIMENTO DE IMAGEM

A descrição de um processo de reconhecimento poderá envolver as seguintes etapas:

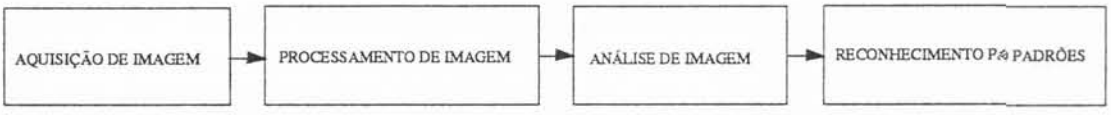


fig. 2.1 Esquema de blocos de um processo de reconhecimento

A imagem digitalizada por um processo de aquisição, é uma função $f(i,j)$ da função continua da imagem $f_C(x,y)$ onde x e y são coordenadas espaciais. Esta digitalização envolve uma amostragem espacial da imagem e uma quantificação da intensidade luminosa em cada ponto.

O conhecimento da distribuição dos níveis de cinzentos na imagem, dá-nos informação relevante sobre diversas grandezas, tais como: contraste, predominâncias de cor, separabilidade entre regiões, etc.

Depois de ter sido feita uma amostragem digital da imagem e de tererem sido quantificados os seus brilhos, é possível sobre esses valores quantificados, definir operações de vários tipos, tais como: Operações Pontuais (ex: inverter escalas de cinzento, redução do número de bits, etc.); Operações Locais (operações de convolução da imagem original, através de uma janela de coeficientes - ex: Filtros de mediana); Operações Globais (exemplo: histogramas, transformações geométricas, transformadas, etc.).

O histograma fornece-nos informação de imagem através da distribuição das frequências de ocorrência dos diferentes níveis de cinzento r, em que r é a variável que indica o nível de cinzento dum ponto na imagem (*pixel*), com valores compreendidos entre *"0"* e *"2^k-1"* (k - número de bits de codificação).

Operar sobre o histograma pode permitir o realce de algumas características da imagem. Através de operações e operadores locais (filtros, gradiente, Laplaciano, etc.), globais (operações geométricas, . . .), ou pontuais (equalização, multiplicações, etc.), é possível intervir sobre as amostras de modo a que essa operação se possa fazer a extracção de características de imagem duma forma mais adequada e mais facilitada.

Os objectivos do processamento (e do pré-processamento) são melhorar a imagem através de vários processos, como: restauração, realce, filtragens, etc. Com a análise, procura-se extrair um conjunto de variáveis de imagem, que a possam

caracterizar o melhor possível. Essa análise pode traduzir-se na extracção dum conjunto de dados numéricos, ou numa estrutura de dados que representem relações entre partes significativas (segmentos) da imagem, ou relações proprietárias entre essas partes.

Com a análise de imagem estudam-se um conjunto de processos que permitem obter da imagem ou dos seus segmentos[1], propriedades de cada um desses segmentos e relações que se possam estabelecer entre eles. Os segmentos tanto podem ser, cursos de água numa imagem de satélite, como peças numa imagem industrial. As propriedades extraídas podem ser: características de forma, geometria ou dimensão das regiões, posições absolutas ou relativas dos objectos na imagem, texturas das superfícies, ou outras.

Os resultados da análise não são uma imagem, mas uma descrição sua de natureza quantitativa. A informação pode referir-se aos seus elementos constituintes, a uma forma simbólica, a uma lista de objectos presentes e suas propriedades, a mudanças ocorridas na cena de imagem, à classificação da imagem ou dos seus segmentos por categorias, etc. Consiste na transformação de uma grande matriz de pontos, que é a imagem, numa sua descrição compacta em termos de unidades visualmente distintas, e das características que a enformam. A informação deverá ser agregada e etiquetada com nomes simbólicos e atributos.

Segundo Pratt[26-A] a análise de imagem está ligada à extracção de medidas, dados ou informação da imagem, por métodos automáticos ou semiautomáticos, e ainda segundo o mesmo autor, a análise de imagem distingue-se de outros tipos de processamento de imagem, tais como: codificação, restauração, realce, pelo facto de o resultado da análise já não ser uma imagem.

A análise da imagem pode exigir a *segmentação* prévia da Imagem, que pode ser definida como uma partição em unidades uniformes e homogéneas, com respeito a alguma característica (brilho, textura,. etc.). A fronteira das regiões segmentadas não deve ser demasiado irregular e o interior da região deve ser compacto. Podem-se definir vários níveis de segmentação. Nos de baixo nível, descrevem-se as orlas e as zonas de brilho homogéneo, nos de nível intermédio fazem-se indicações sobre a superfície e as suas fronteiras, e nos de alto nível descrevem-se as inter-relações entre os segmentos.

Análise de imagem também diverge da clássica definição de reconhecimento de padrões, já que não se limita à classificação das regiões duma cena por categorias previamente definidas, mas em vez disso é desenhada de forma a permitir a descrição de cenas complexas, sem o conhecimento de categorias à priori.

O processo de reconhecimento de objectos numa imagem através do estabelecimento de correspondências (*"matching"*), pode também ser encarada como uma forma de analisar essa mesma imagem. O estabelecimento de correspondências (*"matching"*) assume grande importância, se for feita entre duas imagens ou entre um padrão e uma imagem. Pode ser utilizada na comparação entre duas imagens em

[1]Entenda-se por segmento todo o agrupamento de pixels com um significado específico num determinado contexto. Assim pode entender-se um segmento, segundo uma óptica de primitiva da região, do contorno, da textura ou outra.

momentos diferentes, na comparação entre imagens obtidas por sensores diferentes, ou duas imagens através de pontos de vista diferentes (triangulação)[2], ou na pesquisa de um padrão na imagem (*"template matching"*).

A correspondência raramente é exacta devido à presença de ruído e aos efeitos da quantificação espacial e do brilho. São necessárias portanto medidas de semelhança, entre o padrão $t(i,j)$ e a imagem $f(i,j)$, sendo usadas para tal medidas de deslocamento sobre a imagem, do tipo:

$$D(m,n)=\Sigma\Sigma\,[f(i,j) - t(i-m,j-n)]^2 \qquad (2.1)$$

que traduz uma correspondência entre o padrão $t(i,j)$ e a imagem representada.

Uma medida de similitude mais segura é a correlação cruzada normalizada, definida por Rosenfeld[27-A]:

$$S(m,n)=\Sigma\Sigma f(i,j)\ .\ t(i-m,j-n)$$

$$\varphi(m,n) = \frac{S(m,n)}{\sqrt{\sum_{ij}[f(i,j)]^2}} \qquad (2.2)$$

Um dos problemas ligados ao estabelecimento de correspondências é a lentidão de processamento. Para aumentar a velocidade de tratamento, identificam-se primeiro as posições candidatas às correspondências, usando para isso versões de reduzida resolução da imagem e do padrão e numa segunda fase usando toda a resolução.

Com o processo de reconhecimento de padrões, pretende-se atribuir, a um conjunto de variáveis características da imagem, ou a um vector de características de imagem, uma classe ou categoria adequada, tão adequada que o erro de atribuição se aproxime de zero.

Esta tarefa também chamada de classificação, pode ser efectuada de um modo supervisado, ou não supervisado, conforme é conhecido um conjunto de amostras de treino padronizadas, ou não, relativas a cada uma das classes em que foi dividido o problema. O capítulo VI, abordará este assunto com maior detalhe.

[2]Processo largamente utilizado em visão 3D estereoscópica, à custa da observação de singularidades na imagem, através de pontos de observação distintos.

2.3 TRANSFORMAÇÃO DOS NÍVEIS DE CINZENTO NA IMAGEM

2.3-1 NOÇÃO DE HISTOGRAMA

Considerando a imagem continua *f(x,y)* representada através da função discreta *f(i,j)*, com valores de cinzento estabelecidos no conjunto {0, 1, 2, . . . , 2^k-1}, onde *k* representa o número de *bits* de quantificação da imagem digitalizada. Podemos definir o histograma $H(r)$ duma imagem através de uma função discreta, em que o valor da função corresponde ao número de *pixels* na imagem com o mesmo nível de cinzento, e o domínio corresponde aos níveis de cinzento em que foi quantificada a imagem.

$$H(r)=\sum_{i=0}^{M}\sum_{j=0}^{N}[f(i,j)=r] \qquad (2.3)$$

Contudo a sua representação pode ser feita através de uma função continua, que constitui uma aproximação à representação discreta, já que isso pode melhorar bastante a representação gráfica do histograma, dada a proximidade entre os pontos coordenados quando representados na totalidade no espaço de écran. Neste caso o histograma passa a ser uma função continua mostrando a probabilidade de ocorrência de cada nível de cinzento na imagem. O histograma representa assim a frequência relativa de ocorrência de cada nível de cinzento na imagem ou a densidade de probabilidade $P_r(r)$ dos níveis de cinzento.

$$P_r(r)=H(r)/n \qquad (2.4)$$

onde $P_r(r)$ é a probabilidade de ocorrência de um *pixel* com valor de cinzento r na imagem e n é o número total de *pixels* na imagem

A função de distribuição $CP_r(r)$ representa a função acumulada da densidade de probabilidades $P_r(r)$ dos níveis de cinzento r presentes na imagem.

$$CP_r(r) = \sum_{j=0}^{r} P_r(j) \qquad (2.5)$$

As fig. 2.2 e fig. 2.3 representam respectivamente a imagem de uma prancha de madeira de pinho onde é visível um nó solto, e as funções, densidade de probabilidades dos níveis de cinzento na imagem e a sua distribuição de probabilidades.

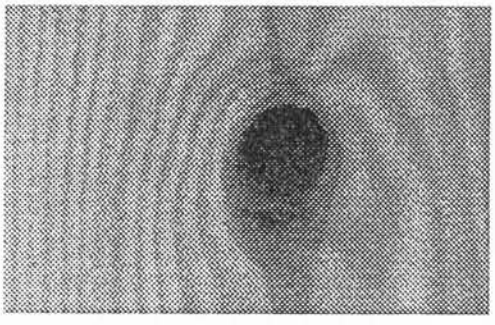


fig. 2.2 Prancha de Pinho, com nó solto

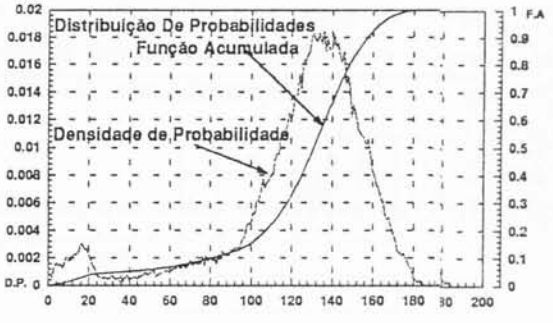


fig. 2. 3 Densidade e distribuição de probabilidades dos níveis de cinzento da imagem representada na fig. 2.2

O histograma representa a forma como se distribuem as intensidades luminosas dos *pixels* sobre a imagem, sem fazer referência à sua localização, indicando apenas a frequência $H(r)$ com que ocorrem. É uma operação que faz corresponder a uma imagem, um vector [H(0), H(1), . . . , H(l)] com $l=2^k-1$.

2.3-2 TRANSFORMAÇÕES POR VIA DETERMINÍSTICA

As transformações determinísticas efectuadas numa imagem, permitem a alteração dos seus *pixels* à custa da aplicação de operadores pré estabelecidos. Essas operações podem ser de três tipos: pontual, se na imagem o processo de transformação for feito *pixel* a *pixel*, local se a operação envolver *pixels* situados numa sua vizinhança, e global se for envolvida toda a imagem na operação.

Nas operações locais, uma vizinhança pode conter *pixels* da imagem que envolvam janelas 3 x 3, 5 x 5 ou maiores, em volta do *pixel* em causa. Quanto maior

for a vizinhança, maiores serão os custos, em termos de tempo de processamento dispendido com a operação. Apesar das virtualidades que possa apresentar, as operações locais têm desvantagens em termos de tempo de processamento, que podem ser consideráveis, dadas as quantidades de informação a processar (imagens de 512x512, contêm 256.000 *pixels* (*bytes*) e vizinhanças 5 x 5 obrigam a operações sobre 6.400.000 *pixels*(*bytes*)), principalmente em operações de tempo real.

Se num processamento for necessária considerar a informação contextual disponível na imagem, para se poder dispor de maiores quantidades de informação, a operação necessitará de utilizar toda a imagem e será considerada, por essa razão, global.

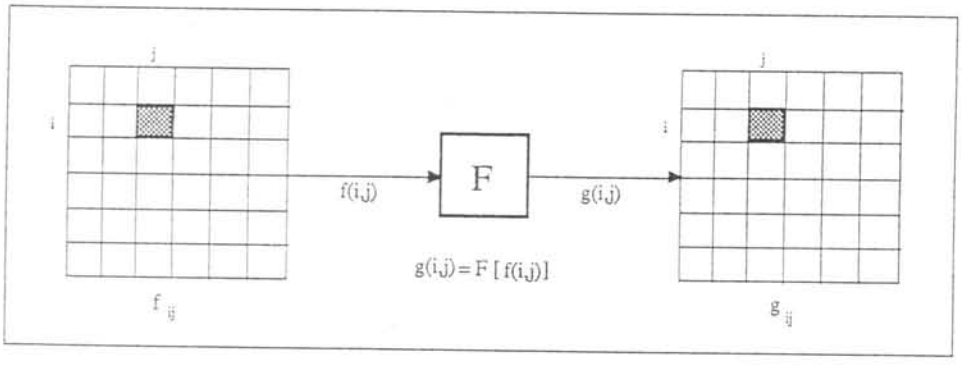


fig. 2.4 Operações pontuais com imagens

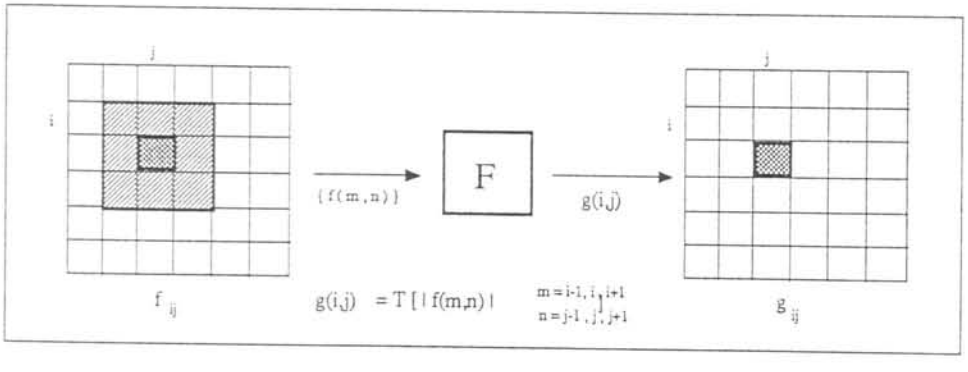


fig. 2.5 Operações locais, para o caso de oito vizinhos

No processamento ponto a ponto, a transformação depende exclusivamente da informação contida nesse ponto, sem tomar em consideração a sua vizinhança. A conversão pode ser conseguida através de uma tabela, ou de uma expressão algébrica de conversão, como se mostra no esquema da fig. 2.4, em que *g(i,j)* é a imagem após a transformação dos *pixels* originais da imagem *f(i,j)*.

A fig. 2.6 mostra algumas funções de transferência que podem ser aplicadas às imagens, resultando nas seguintes transformações:

a - A função representada por uma linha recta (1) permite obter imagens mais claras que a original, enquanto(2) permite obter imagens mais escuras. Todos os *pixels* são multiplicados nestas transformações por um factor positivo.

b - Concentra o brilho da imagem nas cores extremas do histograma, atenuando as cores intermédias. Permite uma melhor discriminação visual dos níveis médios de cinzento à custa da concentração nos extremos da escala(manipulação do contraste).

c - É uma transformação binária da imagem, que reduz a imagem *g(i,j)* a apenas dois niveis: *'0' e '255'*.

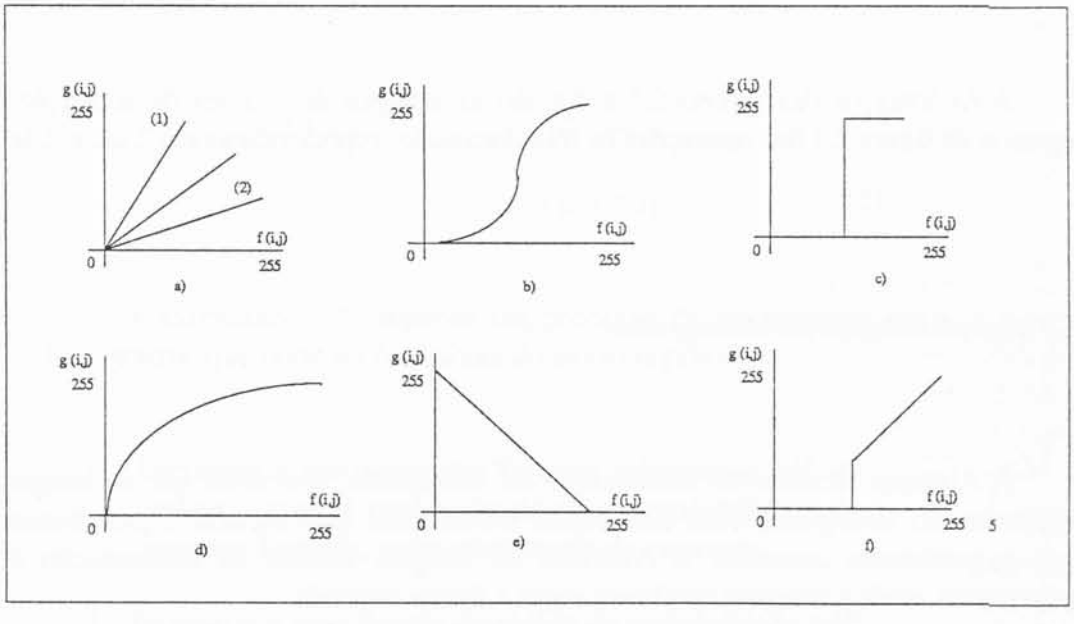


fig. 2.6 Funções de transferência, para transformação determinística de níveis de cinzento na imagem

d - É uma transformação logarítmica, pode ser usada por exemplo para compensar as não linearidades dos sensores.

e - Inverte a escala de cinzentos, permitindo a obtenção de uma imagem em *"negativo"*.

f - Mostra uma função de transferência que permite alterar o histograma de uma imagem, ajustando a gama dinâmica das cores na imagem.

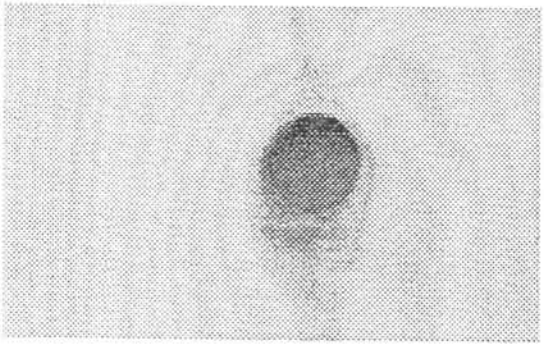


fig. 2.7 Transformação logarítmica da imagem da fig. 2.2

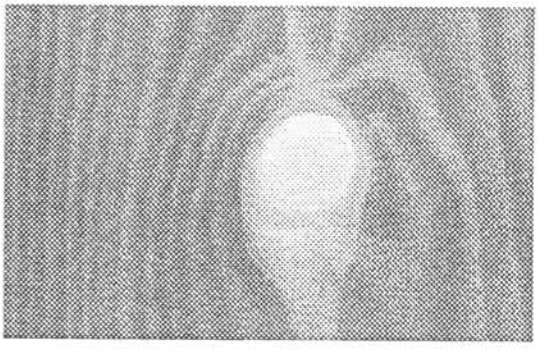


fig. 2.8 Imagem inversa da imagem da fig. 2.2

As imagens das figuras 2.7 e 2.8 são as imagens resultantes da aplicação à imagem da figura 2.2 das operações de transformação[3] representadas em 2.6d) e 2.6e).

2.3-3 Modificações do Histograma

Algumas técnicas de modificação do histograma, procuram que as imagens possuam um histograma com uma forma préviamente determinada. Essas técnicas visam geralmente aumentar o contraste da imagem através da manipulação do histograma, onde a imagem resultante segue a forma desejada

Consideremos uma imagem com a dimensão M x N, em que o número total de *pixels* na imagem é igual a $n=\sum_{i=r_1}^{r_k} n_i = MxN$ onde $r_1, \ldots, r_k$ são os níveis de

[3]As operações de tranformação logarítmica e de inversão a que foi sujeita a imagem da fig. 2.2, foram implementadas por iVngtlog(...), em TSEGMT1.C, e correspondem às imagens representadas nas fig. 2.7 e 2.8.

cinzento da imagem e n_i é o número de *pixels* com o nível de cinzento r_i. A função acumulada da densidade de probabilidades (*"cumulative probability density function"*), é igual a

$$CPdf = \sum_{r_0}^{r_k} (Pdf) = 1 \qquad (2.6)$$

e onde *Pdf* é a função densidade de probabilidades (*"probability density function"*).

A fig. 2.3 mostra dois gráficos que representam, o histograma da imagem da fig. 2.2 e a distribuição de probabilidades (função acumulada da densidade de probabilidade) *CPdf* em relação aos níveis de cinzento *r*, da mesma figura.

Considere-se a variável *r*, correspondente à intensidade luminosa da imagem num ponto, representada pelo valor da função $r=f(i,j)$ com $r_0 \leq r \leq r_j$. Considere-se que $f(i,j)$ representa, para todo o domínio, a imagem original, e é mapeada na imagem de saída $s=g(i,j)$ com $s_0 \leq s \leq s_j$, de tal modo que a distribuição de probabilidades $P_S(s_k)$ sege alguma forma desejada de distribuição de probabilidades que queremos fazer corresponder a $P_r(r_j)$. Com esse objectivo podemos relaciona-las por:

$$\begin{aligned} s &= T(r) \qquad & r_0 \leq r \leq r_j \\ r &= T^{-1}(s) \qquad & s_0 \leq s \leq s_k \end{aligned} \qquad (2.7)$$

A expressão (2.7) sintetisa um processo de modificação de histograma, que pode ser formulada do modo seguinte :

1 - Encontrar a transformação $T(r)$ que relacione o *Pdf* da imagem original com a densidade de probabilidade da imagem desejada, numa nova escala de cinzentos.

2 - Encontrar a nova função densidade de probabilidade *Pdf** da nova imagem, na nova escala de cinzentos, a partir do *Pdf* da imagem original, e da transformação são conhecida.

Na transformação, o número de cinzentos da imagem final pode ser diferente da imagem original, mas a quantidade de número de *pixels* na imagem é mantido constante. Podemos escrever que,

$$\sum_{j=0}^{J} p_r(r_j) = 1 \qquad \sum_{k=0}^{K} p_s(s_k) = 1 \qquad (2.8)$$

As expressões indicadas em (2.8) estabelecem que a soma das densidades de probabilidades têm de ter valor igual a um, podendo ser igualadas. Num processo de equalização essas densidades de probabilidades acumuladas da imagem de entrada e saída, têm de ser iguais qualquer qualquer que seja o índice j

$$\sum_{j=0}^{J} p_r(r_j) = \sum_{k=0}^{K} p_s(s_k) \qquad (2.9)$$

A equalização do histograma, é principalmente útil no realce de imagens de baixo contraste. Procura encontrar-se por este processo a imagem *g(i,j)*, através do rescalonamento do brilho dos seus pontos, tal que os níveis de cinzento do histograma normalizado, sejam iguais a $H_S(k) = 1/k$ com $k = 1, 2, \ldots, K$ em que *k* é o número de cinzentos da imagem equalizada.

O algoritmo é calculado do seguinte modo:

1 - Calcula-se o número médio de *pixels* com a mesma probabilidade de ocorrência, em função do número de níveis de cinzento pretendido para a nova imagem equalizada.

2 - Começa-se no *pixel* de mais baixo valor, acumula-se o número de *pixels* sufíciente até perfazer o número de *pixels* mais próximo da média. Atribui-se a este conjunto de *pixels* um novo valor, em função do rescalonamento dos níveis de cinzento efectuado.

A fig. 2.9 apresenta um exemplo de equalização, obtida por processo equivalente ao descrito no algoritmo anterior, onde o número de níveis de cinzento da imagem equalizada é igual a metade do número de níveis de cinzento na imagem de entrada.

Geralmente o histograma dos níveis de cinzento de uma imagem natural que tenha sido quantificada linearmente, apresenta um desvio na direcção dos *pixels* de

luminância inferior ao valor médio. Os detalhes nas zonas escuras da imagem geralmente não são perceptíveis, o que torna frequentemente necessário fazer a transformação do histograma dessas imagens. Este exemplo serve para ilustrar a importância destas operações de transformação do histograma.

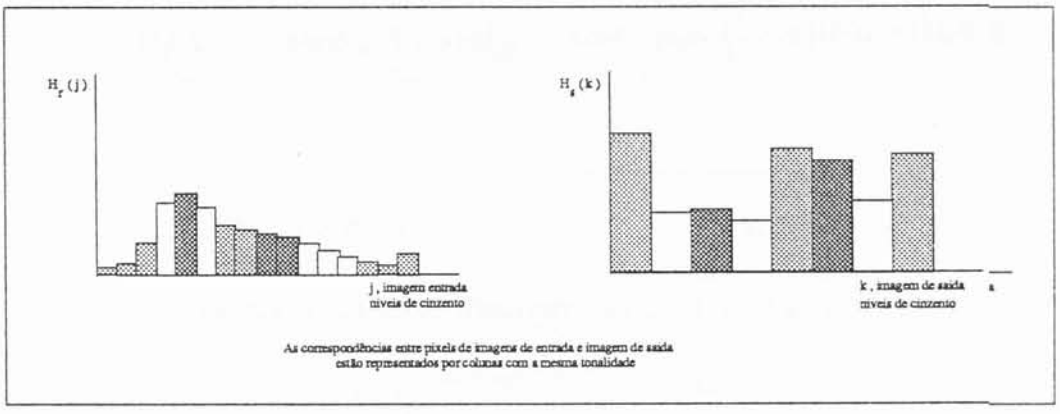


fig. 2.9 Modificação do histograma - Equalização

A técnica usada para aumentar o realce destas imagens, envolve transformações do histograma, em que a imagem original é reescalonada, e em que o histograma da imagem realçada segue uma forma desejada.

Assim vários autores, Andrews[2-B], Prat[26-A] e outros, sugerem a produção de imagens realçadas através de processos de equalização de histogramas, ou de imagens possuindo formas de histogramas exponenciais ou hiperbólicas. .

Generalizando o processo de equalização, podemos dizer que a probabilidade que existe numa imagem de entrada que um *pixel* tenha um nível de cinzento menor ou igual a r_j, deve ser igual à probabilidade que existe na imagem de saída que o nível de cinzento seja menor ou igual a s_k, com $s_k = T(r_j)$. A função acumulada de densidade de probabilidades $\sum_j p_r(j)$ pode ser substituida pela do histograma acumulado $\sum_j H_r(j)$.

A resolução da equação apresentada em (2.9) para s_k, em função de r_j, consiste na inversão dessa equação. Apesar de não ser impossível a inversão desta equação, analiticamente é muito difícil de fazer, excepto para os casos de histogramas de saída uniformes.

Para esse caso, substituindo o somatório pelo integral, obtemos

$$\int_{s_{min}}^{s} p_s(r)ds = \int_{r_{min}}^{r} p_r(r)dr \qquad (2.10)$$

em que $\int p_r \, dr$ fornece o valor acumulado das probabilidades para a imagem de entrada. Para uma imagem com histograma uniforme na saída(equalizado), obtém-se a seguinte expressão:

$$p_S(s) = \text{const} = 1 / (s_{max} - s_{min}) \qquad s_{min} \leq s \leq s_{max} \qquad (2.11)$$

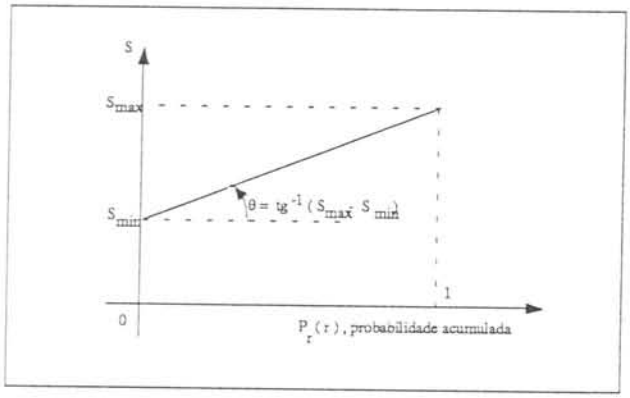


fig. 2.10 Função de transferência para um histograma equalizado

substituindo (2.11) em (2.10)

$$\frac{s - s_{min}}{s_{max} - s_{min}} = P_r(r) \qquad (2.12)$$

a partir da qual temos

$$s = [s_{max} - s_{min}] \, P_r(r) + s_{min} \qquad (2.13)$$

Que é a função de transferência do histograma equalizado[4]. A distribuição acumulada de probabilidades $P_r(r)$ pode ser aproximada por $\sum_j H_r(j)$.

[4]A imagem equalizada da fig. 2.11 foi obtida através da função eqlz(...), do arquivo TSEGMT1.C.

Para o caso de uma imagem de saída com um histograma exponencial, o processo de resolução da equação (2.9) seguia os seguintes passos:

$$p_s(s) = \alpha\, e^{-\alpha[s-s_{min}]} \qquad s \geq s_{min} \qquad (2.14)$$

substituindo (2. 14) em (2. 9), obtém-se

$$\int_{s_{min}}^{s} \alpha\, e^{-\alpha[s-s_{min}]} ds = \int_{r_{min}}^{r} p_r(r) dr \qquad (2.15)$$

ou

$$1 - e^{-\alpha[s-s_{min}]} = P_r(r) \qquad (2.16)$$

resultando a função de transferência (2.17) igual a,

$$s = s_{min} - \frac{1}{\alpha} \ln[1 - P_r(r)] \qquad (2.17)$$

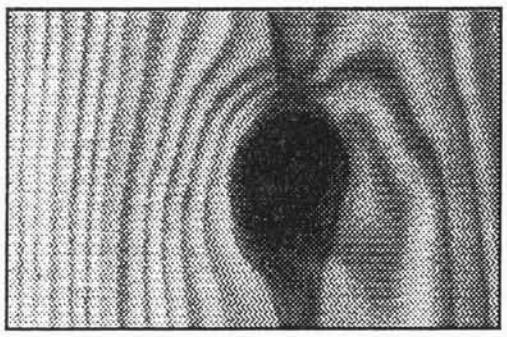


fig. 2.11 Imagem da fig. 2.2 equalizada

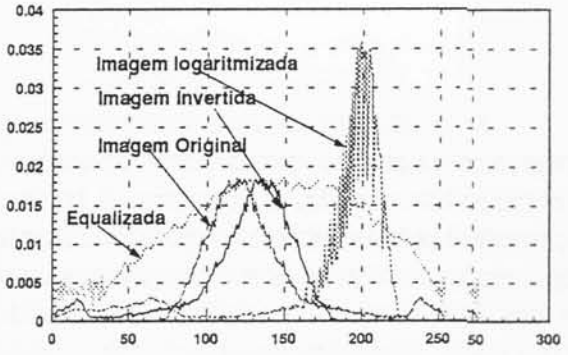


fig. 2.12 Densidade de probabilidades para as imagens: Equalizada, logaritmizada e invertida da imagem da fig. 2.2

Neste processo de transformação de histogramas de imagens, s, foram propostos diversos métodos e formas de hoistograma para as transformações respectivas. Frei[12-B] propôs procedimentos de hiperbolização de histograma, conforme indicado na tab. 2-1. Com este processo o histograma da imagem de entrada é modificado através de funções de transferência indicadas na tabela 2-1, de tal modo que a imagem de saída apresenta um histograma de forma hiperbólica, conforme expressões quatro e cinco do quadro(tab. 2-1).

Densidade de Probabilidade para o Modelo de Saída			Função Transferência
Uniforme	$P_s(s)=\frac{1}{s_{max}-s_{min}}$	$s_{min}\le s\le s_{max}$	$s=[s_{max}-s_{min}]P_r(r)+s_{min}$
Exponencial	$P_s(s)=\alpha\exp\{-\alpha(s-s_{min})\}$	$s\ge s_{min}$	$s=s_{min}-\frac{1}{\alpha}\ln[1-P_r(r)]$
"Rayleigh"	$P_s(s)=\frac{s-s_{min}}{\alpha^2}\exp\left\{-\frac{(s-s_{min})^2}{2\alpha^2}\right\}$	$s\ge s_{min}$	$s=s_{min}+\left[2\alpha^2\ln\left(\frac{1}{1-P_r(r)}\right)\right]^{1/2}$
Hyperbólica (*raiz cúbica*)	$P_s(s)=\frac{1}{3}\frac{s^{-2/3}}{s_{max}^{1/3}-s_{min}^{1/3}}$		$s=\left(\left[s_{max}^{1/3}-s_{min}^{1/3}\right]\left[P_r(r)\right]+s_{max}^{1/3}\right)^3$
Hyperbólica (*logaritmica*)	$P_s(s)=\frac{1}{s[\ln(s_{max})-\ln(s_{min})]}$		$s=s_{min}\left[\frac{g_{max}}{g_{min}}\right]^{P_r(r)}$

tab. 2.1 Funções de transferência para modificação de histogramas, segundo formato previamente definido

As imagens seguintes mostram em que medida as transformações do realce da imagem ocorrem, quando são aplicadas transformações de histograma do tipo hiperbólico. As fig. 2.15 e fig. 2.16, mostram respectivamente a transformação hiperbólica (raiz cúbica) e hiperbólica (logarítmica), aplicadas a uma imagem marcada a giz, em condições de muito baixo contraste. A imagem representada na figura 2.13, foi obtida com interposição de filtro azul na objectiva da câmara, razão pela qual apresenta tão baixo contraste.

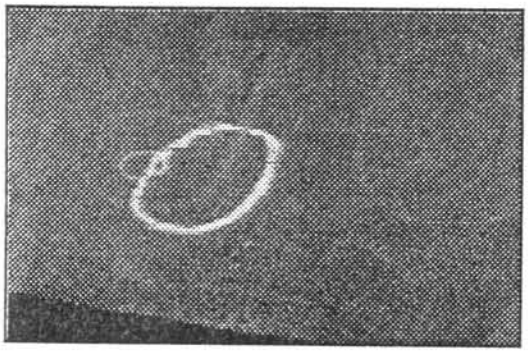


fig. 2.13 Imagem de madeira com baixo contraste e marcações de giz,

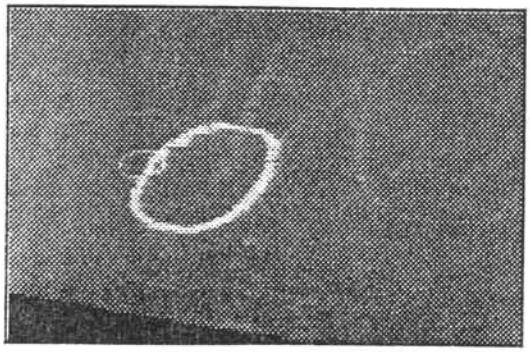


fig. 2.14 Imagem da prancha da fig. 2.13 depois de operação de realce

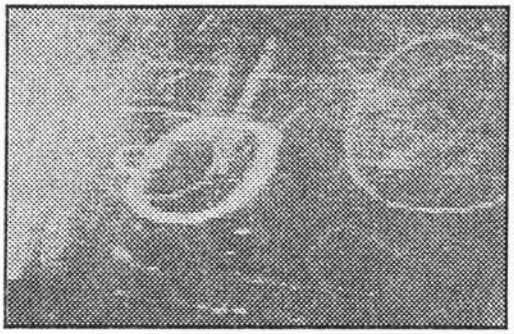


fig. 2.15 Imagem depois de hiperbolização (raiz cúbica)

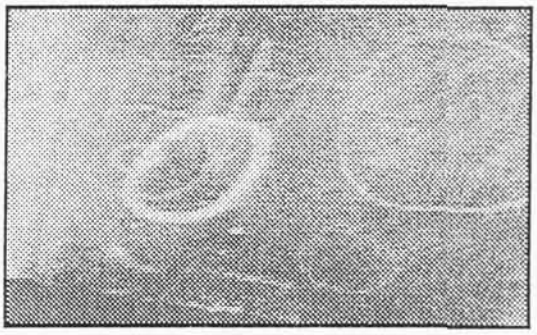


fig. 2.16 Imagem depois de hiperbolização (logarítmica)

Se se pretender aplicar uma transformação, sobre uma imagem, em que o histograma resultante dessa imagem, corresponda a um histograma especificado, ou dito de outro modo, para dois histogramas dados, como pode ser encontrada a transformação T, de tal modo que um desses histogramas seja convertido no outro. O procedimento encontra-se esquematizado na fig. 2.17 e deverá seguir os seguintes passos:

1 - Encontrar a transformação T_1, que transforme a imagem inicial numa imagem com o histograma equalizado (histograma uniforme)

2 - Encontrar a transformação T_2, que transforme o histograma previamente definido para a imagem de saída, no histograma da imagem equalizado. Isto determina uma segunda transformação $S=T_2(r)$.

3 - Calcular a transformação inversa, da que está descrita em 2.

4 - Combinar a transformação T_1 e a transformação inversa T_2^{-1}, para que se encontre a transformação composta, tal que:

$$S = T_2^{-1}(T_1(r)) \qquad (2.\ 18)$$

Esta transformação pode ser útil por exemplo na comparação entre imagens, obtidas em condições de iluminação diferentes.

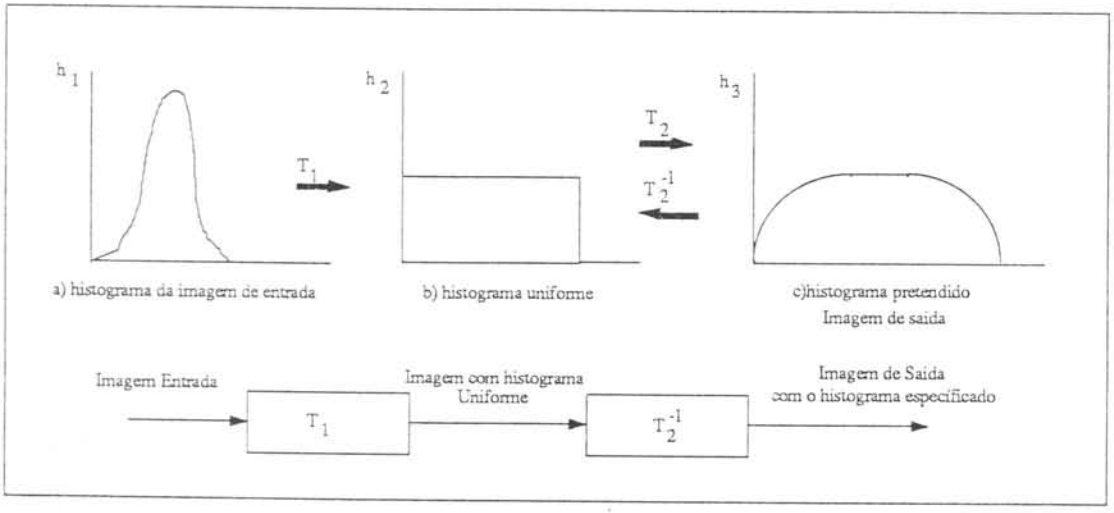


fig. 2.17 Processo de transformação de histograma para uma forma pretendida

2.3.4 RESTAURAÇÃO, REALCE E SUAVIZAÇÃO

2.3.4.1 INTRODUÇÃO

A restauração de uma imagem, visa acima de tudo a recuperação o mais fiel possível da fonte original. A restauração de imagens que foram obtidas na presença de uma ou mais fontes de degradação ou ruído, desfocagens, movimentos de sensores, etc, visa estimar $f^*(x,y)$ tão perto quanto possível de $f(x,y)$.

O realce melhora a qualidade da imagem, procurando aproveitar mais eficientemente as propriedades de visão do olho humano. Assim alterações do escala de brilhos, modificações no histograma, suavização da imagem, acentuação de orlas, representação da imagem em falsa cor, são tudo operações possíveis de efectuar para obter maior realce na imagem.

A suavização tem por objectivo atenuar o ruído, em regiões onde a sua presença seja menos tolerada (por exemplo, a presença de ruído em regiões de brilho constante penaliza a observação). A acentuação de orlas, melhora a imagem e permite a detecção automática e extracção de contornos de objectos, com muito mais facilidade. A acentuação de orlas pode ser feita por meio de operadores diferenciais.

2.3.4-2 MANIPULAÇÃO DO CONTRASTE E DO BRILHO DA IMAGEM. REMAPEAMENTO LINEAR DA IMAGEM

O contraste de uma imagem pode ser melhorada por rescalonamento dos níveis de cinzento de cada *pixel* da imagem. Através de uma transformação linear é possível afastar os níveis de cinzento na imagem de saída, em relação aos níveis de cinzento que possuíam na imagem de entrada.

A manipulação do brilho e do contraste da imagem é também conhecido por remapeamento dos níveis de cinzento da imagem. Isto pode ser feito para compensar deficiências na captação da imagem pelos sistemas sensores. Imagens com pouco contraste podem ser melhorados por este processo.

Há dois modos simples de alterar o contraste na imagem. Uma é adicionar uma constante a cada um dos *pixels* da imagem, a que se chama constante de desvio, outra é multiplicar cada um dos *pixels* da imagem por uma constante (ganho). Estas operações além de afectarem o contraste alteram também o brilho global da imagem. O remapeamento linear dos níveis de cinzento numa imagem, pode ser representado pela expressão mostrada na figura

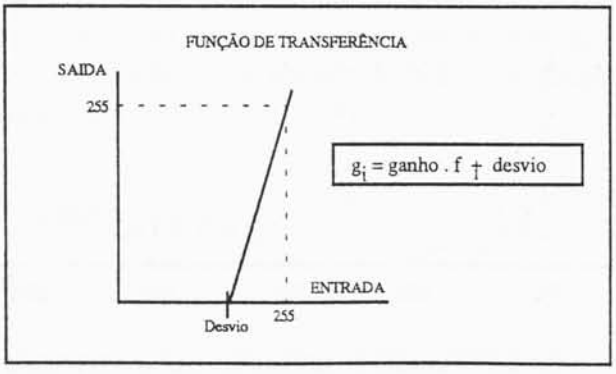


fig. 2.18 Função de transferência utilizada no realce de imagem

f_i representa o *pixel i* da imagem de entrada e g_i o *pixel* do resultante. Há uma relação linear entre f_i e g_i que determina que: se o ganho for menor que '1' o brilho da imagem resultante, será menor do que o brilho da imagem original, o mesmo se verificando se o desvio for negativo. No caso contrário, o brilho da imagem resultante será maior.

Uma forma mais usual de remapeamento linear da imagem é através da especificação dos valores designados para a imagem final [22-A]. Comparando os valores desejados com os valores actuais da imagem, pode-se chegar aos novos valores dos *pixels* de saída do modo representado na expressão (2.19).

$$g_i = (f_i - f_{min})\left[\frac{g_{max} - g_{min}}{f_{max} - f_{min}}\right] + g_{min} \qquad (2.\ 19)$$

O realce de imagem é uma etapa importante no processamento de dados de imagens digitalizadas, com o objectivo de obter dados com melhores aptidões para análises posteriores.

Esta operação pode acentuar características de imagem, como arestas ('*edges*'), fronteiras ou bordas ('*bondaries*') bem como o contraste global. Estas operações não aumentam o conteúdo da informação residente na imagem, mas aumenta-lhe a gama dinâmica das características do histograma.

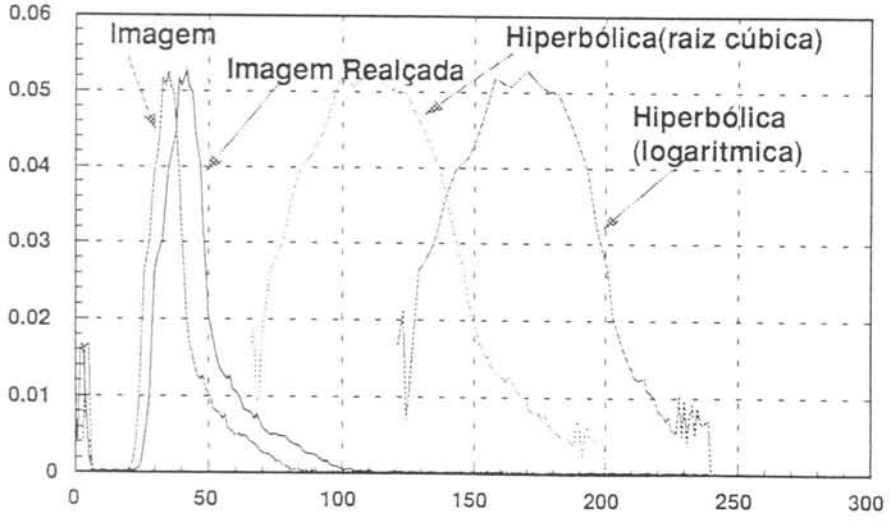


fig. 2.19 Funções de probabilidades de ocorrência dos níveis de cinzento nas imagens das figuras: fig. 2.13, fig. 2.14, fig. 2 15 e fig. 2.16

Os gráficos da fig. 2.19 mostram os histogramas das imagens represesentadas nas figuras; 2.13 a 2.16, obtidas por processos de realce e hiperbolizaçãação[5] do histograma.

2.4 SUAVIZAÇÃO. ELIMINAÇÃO DE RUÍDO

2.4-1 INTRODUÇÃO

Um ruído pode provir de várias fontes geradoras. Pode provir por exemplo de perturbações introduzidas na transmissão através dum canal. Esse ruído geradodo desse modo, não tem qualquer relação com a imagem e o seu valor é geralralmente independente do valor que a imagem possa assumir em qualquer ponto. Este re ruído é por esse motivo de natureza aditiva e pode ser expresso na forma:

$$f(i,j) = p(i,j) + b(i,j) \qquad (2.20)$$

onde $p(i,j)$ representa a hipotética imagem sem ruído e $b(i,j)$ represesenta o ruído na imagem. O ruído pode ser estimado por análise estatística, por exemplo à custa de uma região onde se conheça o brilho que deve possuir natureza consonstante. Este ruído pode ser eliminado por filtros estatísticos clássicos ou por meio de técnicas *"ad hoc"*.

Outra fonte de introdução de ruído nas imagens resulta dos dispositivos que fazem a sua captação. Este tipo de ruído é diferente do anterior já que apresenta natureza multiplicativa, apresentando-se o valor do brilho na imagem dependente do sinal da imagem captada.

$$f(i,j) = p(i,j) \times n(i,j) \qquad (2.21)$$

[5]Estas imagens foram obtidas de 2.13, através de operação de realce com a função rctt(...), em TSEGMT1.C, e para o caso da imagem da fig. 2.14, e as restantes através de hiperbolização, ão, com as funções hypbl(...) e hypblCR(...) em TSEGM1.C, respectivamente.

Na prática é frequente que exista uma diferença entre a imagem real e a imagem quantificada. Essa diferença pode ser frequentemente estimada e quantificada.

A passagem de um ruído multiplicativo a aditivo pode ser efectuada através da logaritmização (logaritmos neperianos) da expressão (2.21). O ruído $n(i,j)$ é um ruído branco de natureza gaussiana de média(μ) e desvio padrão (σ) nulo.

$$\begin{aligned} ln\, f(i,j) &= ln\,[p(i,j) \times n(i,j)] \\ &= ln\,[p(i,j)] + ln[n(i,j)] \\ f'(i,j) &= p'(i,j) + b'(i,j) \end{aligned} \qquad (2.22)$$

A compensação do operador logarítmico é realizada através da aplicação do operador exponencial aos valores resultantes, processo que é explorado nos filtros homomorfico. A aplicação de operadores logarítmicos pressupõe que $f(i,j)$, $p(i,j)$ e $n(i,j)$ sejam positivos.

Em muitas aplicações, as imagens antes de serem analisadas, são primeiro melhoradas em relação ao ruído que contêm. Os processos que consideram este tipo de perturbação na imagem podem seguir abordagens de natureza adaptativa, ou alternativamente abordagens não adaptativas.

A detecção de ruído pontual pode ser feita por comparação do nível de cinzento num ponto da imagem com os seus vizinhos. Se o seu valor for substancialmente maior ou menor que praticamente todos os vizinhos o ponto é classificado como ruído pontual. Se o ponto for classificado como um ponto de ruído o seu nível de cinzento poderá ser substituido pela média pesada dos seus vizinhos, senão o ponto manterá o seu valor.O processo poderá ser sistematizado de um dos seguintes modo:

1 - Comparar os níveis de cinzento de cada ponto com os seus vizinhos.

$$\begin{aligned} &\text{Se} \quad |f - f_i| \geq \tau \\ &\text{o ponto é de ruído} \end{aligned} \qquad (2.23)$$

f é um ponto de ruído e f_i é o nível de cinzento dos vizinhos i, com $i=0, \ldots, 7$ se a vizinhança considerada for a de uma janela de dimensão 3x3 *pixels* e τ o *threshold* considerado.

2 - Comparar o nível de cinzento do ponto com a média dos vizinhos. s.

$$\text{Se} \quad |f - f_{avg}| \geq \tau \quad \text{o ponto é de ruído} \qquad (2.24)$$

f é um ponto de ruído e f_{avg} é a média dos níveis de cinzento dos os vizinhos i, com i=0, ... , 7 se a vizinhança for a de uma janela de 3x3 x3 *pixels* e τ o threshold considerado. A vantagem deste processo so pode ser apenas computacional enquanto as desvantagens podem:m advir de problemas nas fronteiras.

3 - Outro processo[6] chamado de *"processo de decisão de Fuzzy"*, faz`az depender o valor de $g(i,j)$ resultante da operação de eliminação de de ruído, do valor da probabilidade 'p' do ponto '$f(i,j)$' na imagem ser;er ponto de ruído, e da probabilidade (1-p) de não o ser. Umana combinação linear de f com f_{avg} ($f_{avg} = \Sigma f_i/k$) dará o nível de de cinzento g do ponto considerado, através de:

$$g=(1-p)f + p\Sigma f_i/k \qquad (2.25)$$

2.4-2 FILTRO DE MÉDIA

Seja $f(i,j)$ uma funçãode imagem portadora de ruído. A função de imimagem $g(i,j)$ é a resultante de $f(i,j)$ depois ter sido sujeita à filtragem pela média, que ue pode ser obtida do seguinte modo:

$$g(i,j) = \frac{1}{N} \sum_{r=-m}^{m} \sum_{q=-n}^{n} w(r,q) f(i-r, j-q) \qquad (2.26)$$

com i,j=0,1, ... , M , imagem com o mesmo número de linhas e colunasnas

[6]Processo citado por Bekkhoucha, em [6-A].

onde $w_{m,n}$ é uma máscara rectangular pesada com a dimensão m x n, e N o número total de pixels da matriz $w_{m,n}$.

O critério é definido de modo que

$$|f(i,j) - g(i,j)| \geq \tau \qquad (2.27)$$

Se esta condição se verificar, $f(i,j)$ é substituído por $g(i,j)$, senão nenhuma alteração é efectuada.

Várias máscaras são frequentemente usadas em processos de suavização. A figura 2.20 apresenta alguns exemplos

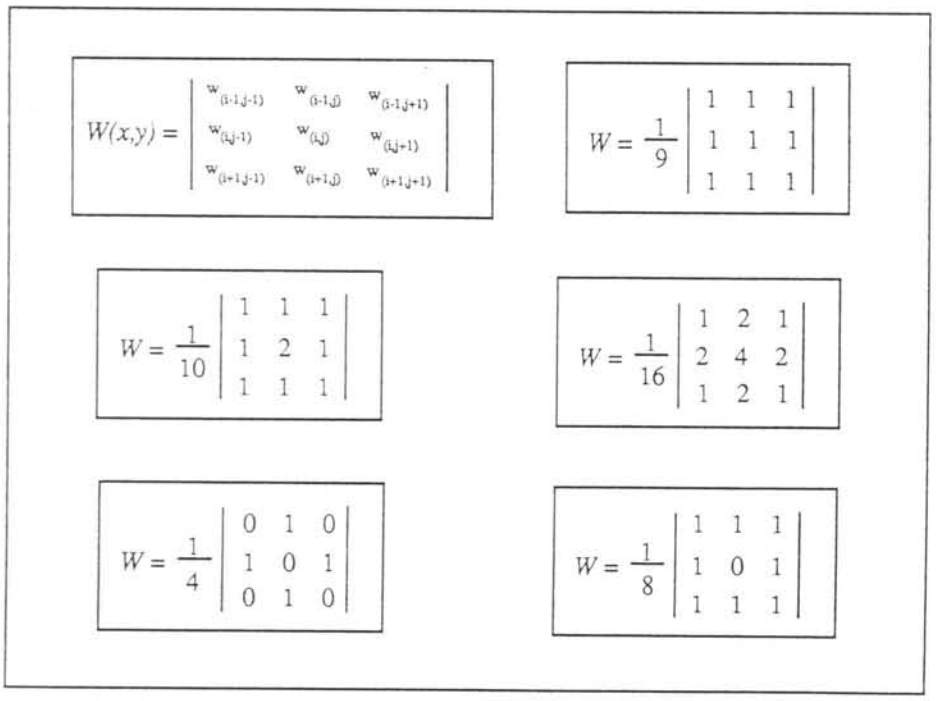


fig. 2.20 Várias máscaras usadas em suavização de imagens

As figuras 2.21 e 2.22 mostram a imagem de uma prancha de madeira, onde são visíveis bolsas de resina, e a imagem correspondente depois da aplicação de um filtro de média[7].

[7]Imagem filtrada pela média através de SPVIZmd(...), função desenvolvida em TFILTR1.C.

O processo podia ser generalizado, e apresentado do seguinte modo:

$$g(i,j)=\begin{cases}\dfrac{1}{N}\displaystyle\sum_{r=-m}^{m}\sum_{q=-n}^{n}w(r,q)f(i+r,j+q) & \text{se } \left|f(i,j)-\dfrac{1}{N}\displaystyle\sum_{r=-m}^{m}\sum_{q=-n}^{n}w(r,q)f(i+r,j+q)\right|>\tau \\ f(i,j) & \text{em caso contrário}\end{cases} \quad (2.28)$$

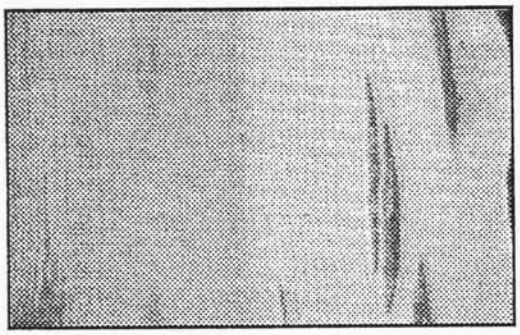


fig. 2.21 Prancha com defeitos: bolsas de resina

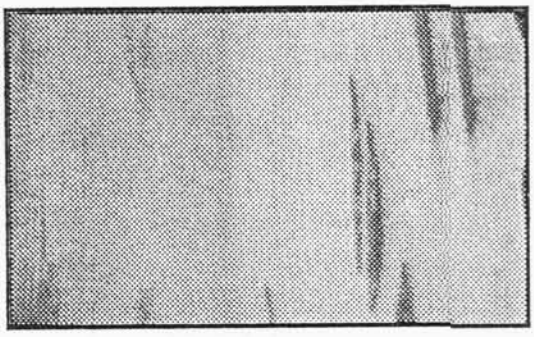


fig. 2.22 Imagem depois de aplicado o filtro de média

Neste caso o processo consistiria na aplicação sucessiva de uma filtragem passa baixo à imagem, seguida de um critério de decisão. A aplicação de um filtro passa baixo à imagem, faz com que as zonas homogéneas apareçam com luminusidade constante, enquanto que as zonas com componentes de frequência espacial elevada, ruído por exemplo, mas também as componentes com diferenças elevadas de luminusidade, sejam bastante atenuadas, acarretando ou podendo acarretar atenuações indesejáveis com a consequente perda de pormenor na imagem. Os critérios de decisão definidos (*threshold* τ) visam atenuar estas consequências.

2.4-3 FILTRO DE MEDIANA

O filtro de mediana é um processo não linear utilizado na redução do ruído. O processo é aplicado a toda a imagem e em cada posição utiliza uma janela centrada sobre o pixel respectivo. O critério baseia-se na determinação da mediana nessa vizinhança, entendendo-se por mediana duma sequência de vizinhos $A_1, A_2, ..., ..., A_N$

de um pixel $f(i,j)$, o valor do pixel que tiver (N-1)/2 valores menores ou iguais ao seu e (N-1)/2 valores maiores ou iguais ao seu. Em termos de probabilidades, seria

$$\text{Se} \quad p(A_n \leq A_m) \geq 1/2 \quad \text{e} \quad p(A_n \geq A_m) \geq 1/2, \quad \text{com } m=\{1, \ldots, N\}$$

$$\Rightarrow A_n, \text{ seja o pixel da mediana nessa vizinhança.} \qquad (2.29)$$

O filtro de mediana promove a eliminação de ruído na imagem, sobretudo do tipo *"salt and pepper"*, e ao contrário do filtro de média preserva as descontinuidades.

São utilizadas na implementação deste filtro[8] vários tipos de máscaras (relacionadas com a tomada de vizinhança), mas as mais utilizadas são as representadas na fig. 2.23.

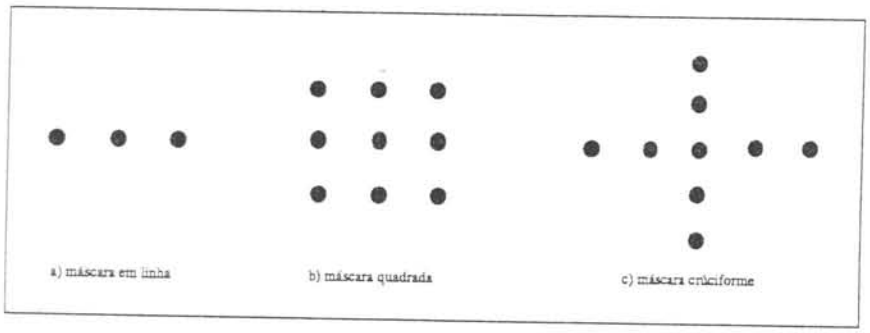


fig. 2.23.Máscaras usadas em filtros de mediana

fig. 2.24 Imagem correspondente à da fig. 2.22 depois de filtrada por filtro de mediana (máscara quadrada)

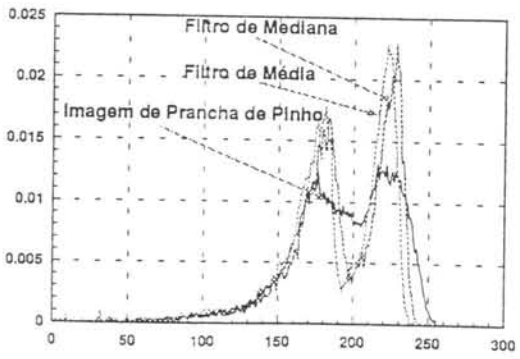


fig. 2.25 Histogramas das imagens das fig. 2.22, 2.23 e 2.25

[8]Imagem da fig. 2.24, obtida a partir da função fmdna(...) em TFILTR0.C.

2.4.4 MÉTODO ADAPTATIVO

Enquanto um método não adaptativo aplica o operador a todos os pontos da imagem, um mecanismo adaptativo utiliza um processo de decisão que permita ajuizar da necessidade de aplicar a correcção ou não a um ponto. O tratamento é assim aplicado em função das características locais da imagem. Um processo adaptativo pode ser esquematizado da seguinte forma:

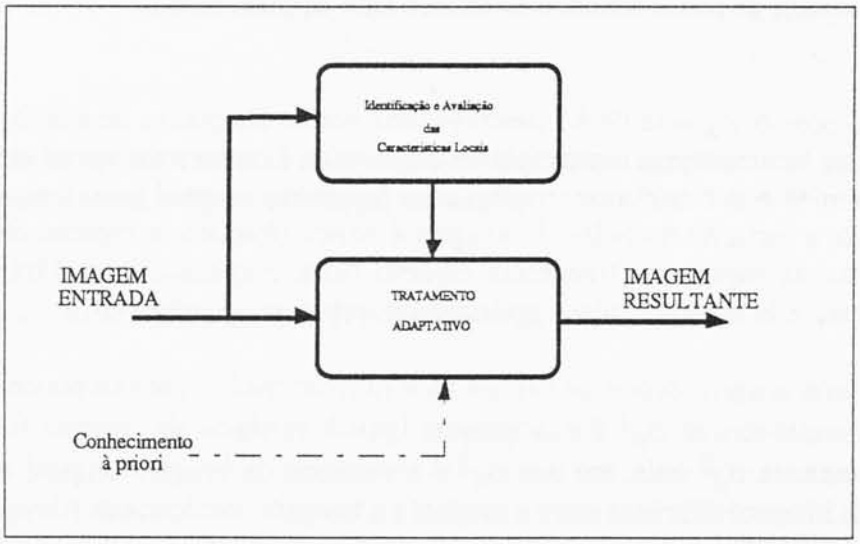


fig. 2.26 Esquema geral de um processo de tratamento adaptativo

A ideia geral de um tratamento adaptativo, em particular para a compensação do ruído, baseia-se na aplicação de um operador que se ajuste às características locais e/ou globais da imagem. O tratamento a efectuar a cada ponto da imagem é função dos dados considerados para o tratamento, que podem envolver conhecimento à priori (como o tipo de degradação e o agente de degradação).

O filtro de média adaptativo, consiste na avaliação da média dos níveis de cinzento dos pixels seus vizinhos, através de:

$$g(i,j)=\frac{1}{N}\sum_{r=-m}^{m}\sum_{q=-n}^{n}w(r,q)f(i-r,j-q) \qquad (2.30)$$

e na aplicação do critério seguinte, para os pesos de *w(r,q)*

$$w(r,q)=\begin{cases}1 & \text{se } |f(i,j)-f(i-r,j-q)|\le\tau \\ 0 & \text{em caso contrário}\end{cases} \qquad (2.31)$$

$$N=\sum_{r=-m}^{m}\sum_{q=-n}^{n}w(r,q)$$

Este filtro tem a vantagem de conservar as transições, já que toma em devida conta unicamente os pixels situados do mesmo lado do pixel tratado.

O processo seguinte [9-A], descreve uma técnica adaptativa de suavização da imagem. Esta técnica utiliza o princípio de que o ruído é muito mais visível em locais da imagem onde esta é uniforme (regiões com frequência espacial baixa), do que em regiões onde a variação do brilho da imagem é brusca (frequência espacial elevada). Dessa forma, às regiões de frequência espacial baixa é aplicada uma filtragem de média normal, e às outras regiões é aplicado um processso ponderado.

Se uma imagem depois de filtrada (por filtro de média) for exactamente igual à original, segue-se que σ_n^2 é exactamente igual à variância da imagem filtrada e consequentemente σ_d^2 nula, em que σ_n^2 é a variância da imagem original e σ_d^2 a variância da imagem diferença entre a original e a imagem resultante da filtragem.

Se se verificar que $\sigma_d^2 \le \sigma_n^2$ é verdadeira, então a imagem filtrada é uma representação adequada da imagem original, se se verificar que é $\sigma_d^2 \ge \sigma_n^2$ então a imagem não pode ser representada pela imagem de mais baixa ordem (imagem filtrada). Deste modo σ_n^2 pode funcionar como *"threshold"* local para testar σ_d^2.

Com base nestes pressupostos o algoritmo seguinte pode ser desenhado:

Divide-se a imagem num conjunto determinado de àreas. Nas áreas de baixa actividade aplica-se o filtro de média, enquanto nas áreas de alta actividade calcula-se uma nova imagem, através da média pesada de *B* e I, em que *I* é a imagem original, *B* a imagem suavizada e *N* a nova imagem calculada.

$$N = (W_B).B + (W_I).I \qquad (2.32)$$

A execução do algoritmo segue as seguintes etapas:

1 - Calcular B por filtragem de I

2 - Calcular a diferença D, onde $D=I-B$

3 - Partir a imagem em p x q secções, denotadas por B_i , I_i , e D_i .

4 - Para cada secção definir θ_i , como sendo igual a

$$\theta_i = min\left\{1.0, \frac{\sigma_n^2}{\sigma_d^2}\right\} \qquad (2.33)$$

onde σ_{n^2} é a variância da imagem com ruído, na secção i, e σ_{d^2} é a variância da imagem diferença, na secção i.

5 - Calcular a nova imagem N por secções da forma seguinte

$$N_i = \theta_i B_i + (1 - \theta_i) I_i \qquad (2.34)$$

2.5 CONVOLUÇÃO DISCRETA

A convolução é uma operação local que opera sobre um conjunto de vizinhos de um pixel, sendo o seu valor igual a uma soma de produtos. Pode ser representada pela expressão:

$$R_{ij} = \sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} I_{k,l}\, h_{i-k,j-l} \qquad (2.35)$$

Cada pixel na operação de convolução é tomada em volta de uma vizinhança do pixel central, e multiplicado pelo elemento correspondente da máscara.

A convolução é uma técnica muito usada na filtragem de imagens (no domínio espacial), mas tem contudo a desvantagem de necessitar de efectuar um conjunto de operações muito elevado, afectando deste modo o tempo de computação dos algoritmos, principalmente em máscaras maiores do que as que têm dimensão 3x3. No entanto para máscaras específicas é possível utilizar "firmware" (uma arquitectura dedicada) que implementa estes filtros de uma forma muito mais rápida.

Na definição de um sistema linear importa considerar a função Ψ , que transforma a imagem I na imagem R .

$$R = \Psi\ I \tag{2.36}$$

O sistema diz-se linear, se para uma combinação de sinais de entrada (ou imagens) produz uma combinação de sinais de saída de tal modo que:

$$\Psi\ (aI + bj) = a\Psi\ [I] + b\Psi\ [J] \tag{2.37}$$

Em que I e J são duas imagens e a e b são duas constantes. Esta expressão mostra que as imagens não agem uma sobre a outra, mas apenas são afectadas pelo sistema duma forma linear (e independente).

Uma propriedade importante dos sistemas lineares são as suas transformações pontuais (*"point spread function"*).Ou seja a imagem que resulta da aplicação ao sistema de uma fonte pontual ideal, é chamada resposta impulsional do sistema, e pode ser escrita na forma seguinte:

$$h(i-k, j-l) = \Psi\ [\delta\ (i-k, j-l)] \tag{2.38}$$

Esta equação relaciona a resposta do sistema a uma fonte pontual localizada no ponto (k,l) e δ denota a função delta de Kronecker, a duas dimensões, expressa por:

$$\delta\ (m,n) = \begin{cases} 0 & m \neq 0, n \neq 0 \\ 1 & m = 0, n = 0 \end{cases} \tag{2.39}$$

A resposta impulsional de um sistema linear, determina a forma como esse sistema afecta as imagens de entrada. A propriedade de sobreposição dos sistemas lineares permite afirmar, que da aplicação de múltiplas imagens ao sistema, daí não resulta qualquer influência de umas sobre as outras. Isto permite considerar uma imagem de entrada como um conjunto discreto de imagens pontuais uma em cada ponto de localização particular da imagem. A imagem pode assim ser descretizada e representada pela expressão(2.40) com a ajuda de impulsos unitários δ, na definição das entradas individuais da imagem *I*. O produto dentro do somatório só não é zero quando $i=k$ e $l=j$.

$$I_{ij} = \sum_k \sum_l I_{k,l}\, \delta\ (i-k, j-l) \qquad (2.40)$$

Aplicando a imagem a um sistema linear, resulta

$$\Psi\ \left[I_{ij}\right] = \Psi \left[\sum_k \sum_l I_{k,l}\, \delta\ (i-k, j-l) \right] \qquad (2.41)$$

Por causa da sobreposição linear, a resposta do sistema a um conjunto de entradas é idêntica à soma das respostas tomadas individualmente.

$$\Psi\ \left[I_{ij}\right] = \left[\sum_k \sum_l I_{k,l}\, \Psi\, \delta\ (i-k, j-l) \right] \qquad (2.42)$$

Por (2.41) e (2.42)

$$\Psi\ \left[I_{ij}\right] = \left[\sum_k \sum_l I_{k,l}\, h\ (i-k, j-l) \right] \qquad (2.43)$$

que é a expressão da convolução discreta. Pode-se constatar que (2.43) é igual a (2.35), que foi a definição inicialmente apresentada para a convolução. A máscara utilizada na filtragem define efectivamente a resposta impulsional do sistema.

2.6 REALCE DE ORLAS NA IMAGEM ATRAVÉS DE TÉCNICAS BASEADAS NO GRADIENTE

2.6-1 NOÇÃO DE GRADIENTE

O realce duma imagem ou dos pixels que constituam orlas nessa imagem, pode ser conseguido por um processo de diferenciação[9] da imagem. Para uma imagem $f(i,j)$ bidimensional a diferenciação da imagem pode ser obtida pelo cálculo do gradiente $\nabla[f(i,j)]$ dessa mesma imagem. Pode ser representado do seguinte modo:

$$\nabla[f(i,j)] = \begin{vmatrix} \frac{\partial f}{\partial i} \\ \frac{\partial f}{\partial j} \end{vmatrix} \qquad (2.44)$$

ste vector aponta na direcção da máxima variação dos níveis de cinzento na imagem, enquanto que a amplitude deste vector fornece a medida dessa variação. O valor da amplitude ou módulo do vector é dado por:

$$\nabla[f(i,j)] = \left[\left(\frac{\partial f}{\partial i} \right)^2 + \left(\frac{\partial f}{\partial j} \right)^2 \right]^{\frac{1}{2}} \qquad (2.45)$$

considerando a derivada ($\delta f / \delta i$) a variação na direcção horizontal e a derivada ($\delta f / \delta j$) a variação na direcção vertical, a implementação do gradiente pode ser representada como mostra a fig. 2.27 alínea a).

[9]A imagem filtrada da fig. 2.30, foi obtida através da operação de um filtro de Roberts, que não é mais do que a implementação de uma operação por realce de orlas através de processos de cálculo do gradiente. A função que executa essas operações chama-se rbert_ft(...) e foi implementada em TFILTR1.C.

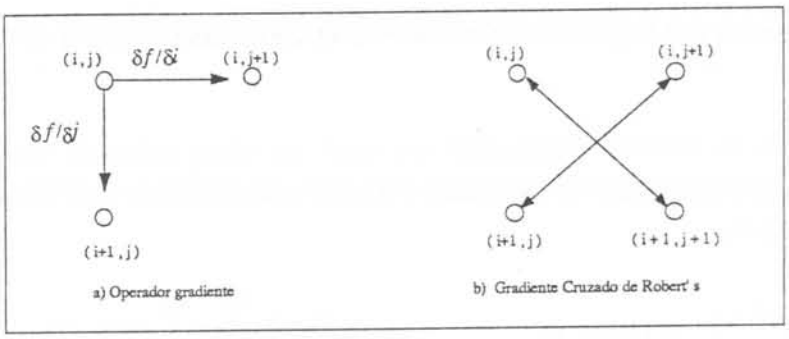


fig. 2.27 Implementação do gradiente

A expressão (2.45) pode ser calculada por:

$$| \nabla f | = [(f(i,j) - f(i+1, j))^2 + (f(i,j) - f(i,j+1))^2]^{1/2} \qquad (2.46)$$

ou para um menor custo computacional

$$| \nabla f | = | f(i,j) - f(i+1, j) | + | f(i,j) - f(i,j+1) | \qquad (2.47)$$

Esta forma de cálculo do gradiente (2.46) e (2.47) diz respeito ao *"gradiente a três pontos"* enquanto a que está representada na fig. 2.27-b) diz respeito a quatro pomntos, gradiente de Robert's, em que o valor do módulo é expresso por (2.48)

$$| \nabla f | = | f(i,j) - f(i+1, j+1) | + | f(i+1,j) - f(i,j+1) | \qquad (2.48)$$

a amplitude do gradiente será grande em locais de "edges" proeminentes e pequeno em zonas homogéneas.

Existem várias técnicas utilizadas no realce de orlas em imagem. Entre elas descrevem-se as que estão referenciadas nos pontos 2.5.2 a 2.5.3.

2.6-2 PROCESSO DE REALCE UTILIZANDO O GRADIENTE

Calcula-se a amplitude do gradiente $|\nabla f(i,j)|$, e determina-se $g(i,j)$ de acordo com expressão 2.41.

$$g(i,j)=\begin{cases} |\nabla f(i,j)| & \text{se } |\nabla f(i,j)| \geq \tau \\ f(i,j) & \text{em caso contrário} \end{cases} \qquad (2.49)$$

em que τ é um *"threshold"* determinado

2.6-3 PROCESSO DE REALCE UTILIZANDO MÉTODOS ESTATÍSTICOS

Pode ser implementado através do algoritmo seguinte

1 - Específicar uma janela 3x3, 5x5 ou 7x7 para o pixel central (i,j)

2 - Calcular o desvio padrão σ ou a variância σ^2 para os elementos situados dentro de cada janela considerada, com σ^2 dada por

$$\sigma^2(i,j)=\frac{1}{N}\sum_{i,}^{\sqrt{N}}\sum_{j\in janela}^{\sqrt{N}}|f(i,j)-\overline{f}(i,j)|^2 \qquad (2.50)$$

onde N é o número de pixels na janela e $\overline{f}(i,j)$ é a média dos pixels dentro da janela.

3 - Substituir o valor do pixel, pelo valor de $g(i,j)$, igual ao quociente dado por $g(i,j) = f(i,j) / \sigma\ (i,j)$. O valor dos pixels, será aumentado em amplitude em relação ao valor original $f(i,j)$, se se tratar de uma orla, e decrementado em caso contrário.

2.6-4 FILTRO PASSA ALTO ("UNSHARP MASKING")

O realce de orlas pode ser feito por filtragem[10], através da convolução da imagem original com determinados tipos de máscaras, representadas na fig. 2.28.

$$g(i,j) = \sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} f_{k,l}\, H_{i-k,j-l} \qquad (2.51)$$

1 - todos os pixels $g(i,j)$ são obtidos pela convolução de $f(i,j)$ com H .

2 - Escolher para novo valor da imagem, o valor obtido $g(i,j)$ ou $f(i,j)$, conforme $|f(i,j) - g(i,j)| \leq \tau$ é ou não maior que τ respectivamente, sendo τ um *"threshold"* escolhido.

$$H = \begin{vmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{vmatrix} \qquad H = \begin{vmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{vmatrix} \qquad H = \begin{vmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{vmatrix}$$

fig. 2.28 Máscaras para filtros passa alto

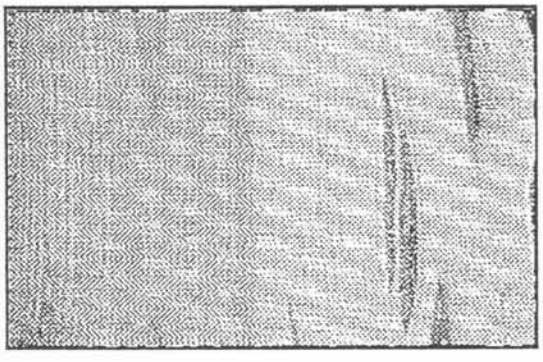


fig. 2.29 Filtro passa alto sobre imagem da fig. 2.21

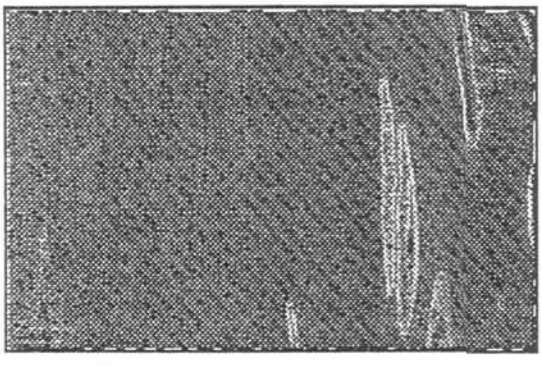


fig. 2.30 Filtro de Roberts sobre imagem da fig. 2.21

[10]Um filtro passa alto é implementado pela função fpalt(...), em TFILTR2.C, através da segunda máscara representada na fig.2.28. A fig. 2.29, mostra o resultado da filtragem.

2.6-5 UTILIZANDO O LAPLACIANO

Na determinação do Laplaciano[29-A] é necessário considerar a vizinhança representada na fig. 2.31.

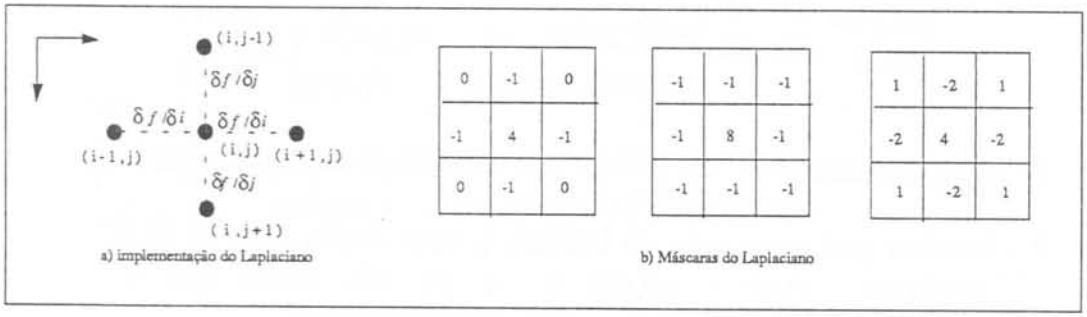


fig. 2.31 Implementação do laplaciano

O Laplaciano é definido por:

$$\nabla^2 f(i,j) = \frac{\partial^2 f}{\partial i^2} + \frac{\partial^2 f}{\partial j^2} \tag{2.44}$$

o que é equivalente a

$$\nabla^2 f(i,j) = \frac{(f_{i+1,j} - f_{i,j}) - (f_{i,j} - f_{i-1,j})}{(\Delta i)^2} + \frac{(f_{i,j+1} - f_{i,j}) - (f_{i,j} - f_{i,j-1})}{(\Delta j)^2} \tag{2.52}$$

sendo Δi e Δj iguais à unidade

$$\nabla^2 f(i,j) = 4(f_{avg} - f_{i,j}) \tag{2.53}$$

onde f_{avg} é a média dos quatro vizinhos mais próximos

$$f_{avg} = (1/4)(f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}) \tag{2.54}$$

Laplaciano pode ser assim obtido por convolução, através duma das máscaras, indicadas na fig. 2.31

$$\nabla^2 f(i,j) = \begin{vmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{vmatrix} \bullet \begin{vmatrix} 0 & f_{i,j-1} & 0 \\ f_{i-1,j} & f_{i,j} & f_{i+1,j} \\ 0 & f_{i,j+1} & 0 \end{vmatrix} \qquad (2.55)$$

Desta forma obtém-se uma imagem realçada nas suas orlas, mas também dos pontos de ruído da imagem. O ruído pode apesar disso ser distinguido das orlas verdadeiras, pelo facto de apresentar um valor maior, já que enquanto as orlas são direccionais o ruído não o é (ambas as derivadas de 2ª ordem .$\nabla_i^2 f$ e $\nabla_j^2 f$ apresentam valores elevados).

Através do gradiente de imagem posso também obter uma imagem despida de ruídos, através duma operação do tipo

$$(i,j) = f(i,j) - k\tau \ \nabla^2 f \qquad (2.56)$$

onde $k\tau$ é uma constante previamente escolhida.

2.6-6 OPERADORES NÃO LINEARES DE ORLAS

Existem vários operadores não lineares de orlas, apenas aqui focaremos os operadores de Sobel[29-A] e de kirch[9-A].

Sobel

Este operador utiliza janelas de dimensão 3x3, centrada sobre um pixel *(i,j)*. A amplitude do gradiente é definida por (2.57) e S_i e S_j são obtidos por convolução segundo (2.35), e as máscaras estão definidas em (2.58)

$$S = (S_i^2 + S_j^2)^{1/2} \quad \text{ou} \quad S = |S_i| + |S_j| \qquad (2.57)$$

S_i é obtido por convolução com a máscara w_i e S_j é obtido por convolução com a máscara w_j, iguais a:

$$w_i = \begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{vmatrix} \qquad w_j = \begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix} \tag{2.58}$$

Kirsch

Kirsch propôs um algoritmo de reforço (ou realce) de orlas utilizando também janelas de dimensão 3x3.

O algoritmo utiliza o seguinte critério

$$g(i,j) = max\begin{pmatrix} 7 \\ 1, max\left[|5S_i - 3T_i|\right] \\ i=0 \end{pmatrix} \tag{2.59}$$

em que S_i e T_i são calculados do seguinte modo

$$S_i = A_i + A_{i+1} + A_{i+2}$$

e

$$T_i = A_{i+3} + A_{i+4} + A_{i+5} + A_{i+6} + A_{i+7} \tag{2.60}$$

com os A_K definidos numa vizinhança de $f(i,j)$, do seguinte modo

$$\begin{matrix} A_0 & A_1 & A_2 \\ A_7 & f(i,j) & A_3 \\ A_6 & A_5 & A_4 \end{matrix} \tag{2.61}$$

2.6-7 DETECÇÃO DE ZERO (ZERO CROSSING)

As orlas numa imagem podem ser detectadas pelo cruzamento de zero da 2ª derivada da imagem. Este processo está esquematizado na fig. 2.32, para o caso de uma dimensão

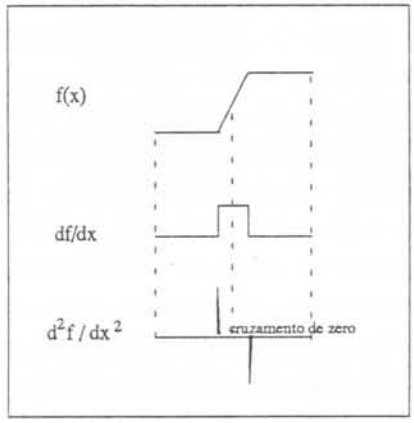


fig. 2.32 Detecção de edges por cruzamento de zero da 2ª derivada

O Laplaciano é muito sensível ao ruído e qualquer *"riple"* em *f(x)* faz aparecer um edge nesse ponto. Um edge pode ser distinguido de um ponto de ruído numa imagem, porque no caso de um edge a variância local é elevada. Essa constatação permitiu a Sing-Tze-Bow[9-A], propor um processo de detecção de edges através do Laplaciano, e eliminação dos edges falsos provocados por ruídos através do cálculo da variância local. Os passos seguintes descrevem o algoritmo:

1 - Cálculo da variância

$$\sigma_f^2(i,j) = \frac{1}{(2M+1)^2} \sum_{k=i-M}^{i+M} \sum_{l=j-M}^{j+M} \left[f(k,l) - \overline{f}(i,j) \right]^2 \qquad (2.62)$$

onde

$$\overline{f}(i,j) = \frac{1}{(2M+1)^2} \sum_{k=i-M}^{i+M} \sum_{l=j-M}^{j+M} f(k,l) \qquad (2.63)$$

2 - Comparação do valor do Laplaciano $\nabla^2 f(i,j)$, correspondente ao pixel $f(i,j)$, com $i,j = 1, 2, \ldots, L\text{-}1$ com a variância na vizinhança desse pixel. Deste modo é possível eliminar a ocorrência de falsos edges, de acordo com o esquema

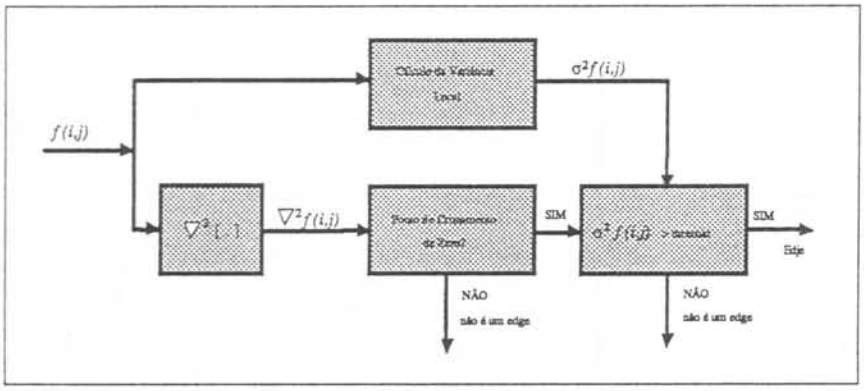


fig. 2.33 Diagrama de blocos de um algoritmo de determinação de edges (calcula o Laplaciano e avalia a variância local, para a tomada de decisão)

As figuras[11] seguintes representam respectivamente a filtragem da imagem da fig. 2.21 através de um filtro de Sobel e a filtragem da mesma figura através da utilização do Laplaciano, pelo processo descrito neste ponto 2.5-7. De referir que apesar de se ter procurado com este processo eliminar os falsos *'edges'* na utilização do Laplaciano, o processo não é totalmente eficaz, como se pode ver na figura 2.35, dado a grande sensibilidade do algoritmo ao ruído.

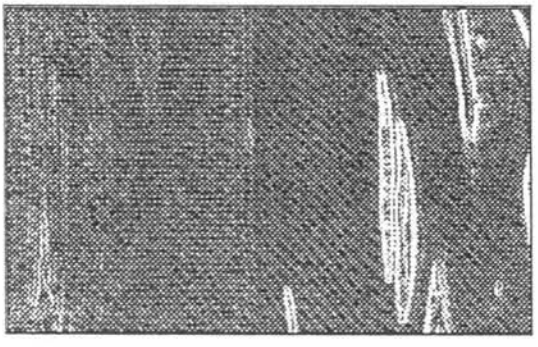


fig. 2.34 Filtro de Sobel a partir da imagem da fig. 2.21

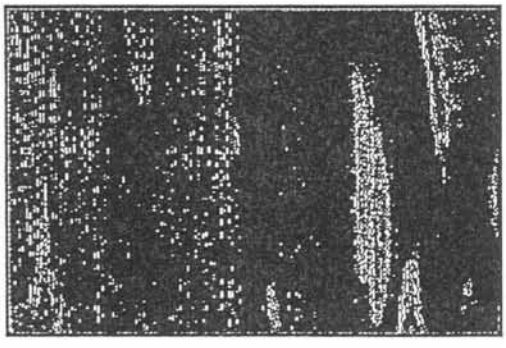


fig. 2.35 Laplaciano a partir da imagem da fig. 2.21

[11] As imagens das fig. 2.34 e fig. 2.35, foram obtidas através de filtros de sobel, sobel_ft(...) em TFILTR1.C e lp_ft2(...) em TFILTR0.C, para uma operação envolvendo o laplaciano e avaliação da variância local, para a tomada de decisão.

2.7 CONCLUSÕES

Através deste capítulo é feito um estudo, sobre o assunto usualmente chamado, de processamento digital de imagem. O estudo é feito com o objectivo central de preparação do terreno, através do treino e apresentação de um conjunto de ferramentas, tendentes a facilitar a análise das imagens de madeira, e a preparar a recolha de dados globais e nos segmentos das imagens. Esses dados são a condição essencial, para que se possam classificar os defeitos na madeira, dados esses que podem não ser facilmente extraíveis, havendo nesse caso que recorrer em primeiro lugar, por exemplo, a técnicas de elimição do ruído, ou de realce de imagem.

Com essa finalidade são abordados fundamentalmente, técnicas de transformação de histogramas para formas desejadas, realce de imagem, eliminação de ruídos e filtragens. Alguns desses assuntos são apresentados a título únicamente indicativo, enquanto outros são posteriormente utilizados no processo global de *localização e reconhecimento de defeitos em madeira de pinho*.

CAPÍTULO III

SEGMENTAÇÃO

3. SEGMENTAÇÃO

3.1 INTRODUÇÃO

A segmentação é uma operação ou conjunto de operações, que fixam como objectivo a divisão da imagem em conjuntos de regiões com significado, não sobreponíveis, e cuja reunião das partes resulte na imagem original. Poderá ser dito também de outro modo, que a segmentação é a operação que procura identificar superfícies e contornos significativos na imagem, necessários a uma aplicação particular, e que constituam as primitivas dos objectos representados nessa imagem.

□ A segmentação tende geralmente a obedecer às regras seguintes:

- As regiões de uma imagem segmentada devem ser uniformes e homogéneas no que respeita a algumas características, tais como nível de cinzento ou textura.
- As regiões interiores a uma região segmentada, devem ser simples e ausentes de pequenos buracos
- Regiões segmentadas adjacentes, devem ter valores significativamente diferentes em relação à característica à qual foram consideradas uniformes.
- As fronteiras de cada segmento devem ser simples, continuas e suficientemente precisas.

Estes objectivos nem sempre são fáceis de conseguir, já que a uniformidade das regiões e a continuidade dos contornos, são muitas vezes difíceis de obter.

Poderão assim surgir problemas de vária ordem, desde os relacionados com regiões de fronteiras comuns, caracterizadas por valores diferentes das características, mas apesar disso pertencentes à mesma região e necessitando por isso de serem fundidas. Os problemas podem também estar relacionados com segmentos de fronteira perdidos, ou a dificuldades ligadas à presença na imagem de ruídos, reflexos, sombras, etc, ou ainda a dificuldades inerentes a processos de segmentação que exijam tomadas de decisão relacionados com o nível de segmentação a utilizar num processo.

Existem muitas técnicas que visam segmentar as imagens. Algumas baseiam-se na procura de regiões da imagem cujos *pixels* tenham de algum modo, semelhanças com os seus vizinhos. Isso pode acontecer, quer porque os *pixels* apresentem valores de cinzento semelhantes, quer porque façam parte de um grupo de *pixels* com a mesma textura repetitiva, quer porque façam parte de um grupo de *pixels* aglutinados por uma mesma propriedade comum. Outras técnicas de segmentação baseiam os seus mecanismos na identificação de fronteiras consistentes, entre objectos ou regiões. Se uma orla/delimitação completa e fechada é encontrada, ela passa a constituir o contorno dessa região, tomada como distinta das demais.

A segmentação quer se baseie em regiões, em contornos, em texturas ou em propriedades morfológicas, visa sempre a divisão da imagem em segmentos com significado, que dependem obviamente da imagem, dos objectivos de utilização a dar a essa informação e da sua relação com o utilizador. Por esse motivo não pode haver técnicas de segmentação universais, porque elas não dão sempre resultados equivalentes, em circunstâncias diferentes.

A segmentação pode ter natureza contextual ou não contextual. Na segmentação não contextual a relação entre as características dos *pixels* ou das regiões são ignoradas. Na segmentação contextual o processo de segmentação emprega relações proprietárias de características, entre *pixels* ou entre regiões vizinhas. A informação local é tida em consideração, apesar de muitas vezes a informação qualitativa não ser muito fácil de quantificar.

Exemplos de aplicações não contextuais, podem ser as segmentações que usam padrões estatísticos baseados na intensidade luminosa dos *pixels* (níveis de cinzento), chamadas segmentações por *nível/patamar de segmentação* ou *"thresshold"* τ . A detecção de fronteiras (*"edges"*) e segmentação por crescimento de regiões são aplicações tipicamente contextuais, onde as decisões de pertença/exclusão advêm das características da região, em vez de se basearem nas características dos *pixels* tomadas individualmente.

Procura-se fazer neste capítulo, uma descrição sintética de algumas técnicas de segmentação testadas e usadas ao longo do trabalho sobre *localização e reconhecimento de defeitos em madeira*. Simultâneamente faz-se o enquadramento do assunto (*reconhecimento de defeitos em madeira*) no tema geral do capítulo, a que chamámos Segmentação. Nem todas as técnicas estudas mostraram iguais potencialidades, pelo que as mais utilizadas no processo de *localização e reconhecimento de defeitos em madeira* foram as baseadas em *"threshold"*, extraído por análise de concavidades no histograma, extraídos pelo método de Otsu, e extraídos por. técnica mista baseada na divisão de regiões, com utilização de árvore quaternária como suporte. Esta técnica de extracção do *threshold* revelou-se útil nalgumas situações particulares.

A razão principal do capítulo, deve-se resumidamente, ao facto da divisão da imagem em regiões ser a primeira operação (ou das primeiras) a efectuar-se num processo de reconhecimento de padrões/objectos. Uusuamente a cada região segmentada corresponde um objecto/defeito diferente.

3.2 TÉCNICAS DE SEGMENTAÇÃO POR REGIÕES

3.2-1 INTRODUÇÃO

Nas técnicas de segmentação baseadas em regiões são exploradas as propriedades locais ou globais na imagem. O processo de divisão e conquista *"split and merge"*, usualmente estudado como uma categoria à parte, não é mais do que uma técnica local de segmentação por regiões.

Na partição das imagens, se o *nível de semgmentação* τ for determinado apenas à custa do nível de cinzento de cada *pixel*, diz-se que estamos a utilizar uma técnica dependente do ponto. Se τ (*nível de semgmentação*) for determinado à custa de uma propriedade local, na vizinhança de cada *pixel*, diz-se que estamos a utilizar uma técnica de segmentação dependente da região.

Para binarizar uma imagem (técnica de divisão da imagem em dois níveis, à custa da determinação do *patamar de segmentação-*'τ ' *"threshold"*) é necessário em primeiro lugar determinar o *nível de segmentação-*'τ '. Existem técnicas de determinação do *nível de segmentação* 'τ ' ditas paramétrica como a que está referida em 3.2-3, onde o se faz uma apróximação do histograma no sentido do desvio quadrático, através de uma soma de distribuições gaussianas. A esta apróximação são aplicadas decisões estatisticas. Nas técnicas ditas não paramétricas faz-se a distinção entre binarização global e binarização local. As técnicas de binarização global procuram um *patamar de segmentação* (*"threshold"*) para toda a

imagem, enquanto que as técnicas de binarização local, partem a imagem em subimagens e determinam os patamares de segmentação para cada uma dessas subimagens.

3.2.2 TÉCNICAS GLOBAIS NÃO PARAMÉTRICAS

3.2.2-1 DIVISÃO DO HISTOGRAMA POR *"THRESOLD"* - NIVEL SEGMENTAÇÃO. BINARIZAÇÃO

Seja definido um conjunto de números naturais N e um conjunto de coordenadas espaciais *(i,j)* para uma imagem digitalizada. Seja G={0,1, ... , *l*-1} um conjunto de números inteiros representando os níveis de cinzento na imagem. Então uma imagem pode ser representada como uma função que faça corresponder ao domínio NxN um valor no conjunto G, tal que $f : N \times N \rightarrow G$. A intensidade (ou nível de cinzento) de um *pixel* com coordenadas *(i,j)* é representado por *f(i,j)*.

A segmentação de imagens à custa da análise do histograma dos níveis de cinzento *r=f(i,j)*, baseia-se na procura de um vale, no qual possa ser estabelecido o patamar de segmentação 'τ ', que divida o histograma em dois blocos distintos B_1 e B_2. Se o histograma for bimodal, este vale pode ser encontrada com facilidade, mas o mesmo não acontecerá em casos de histogramas uni ou multimodais.

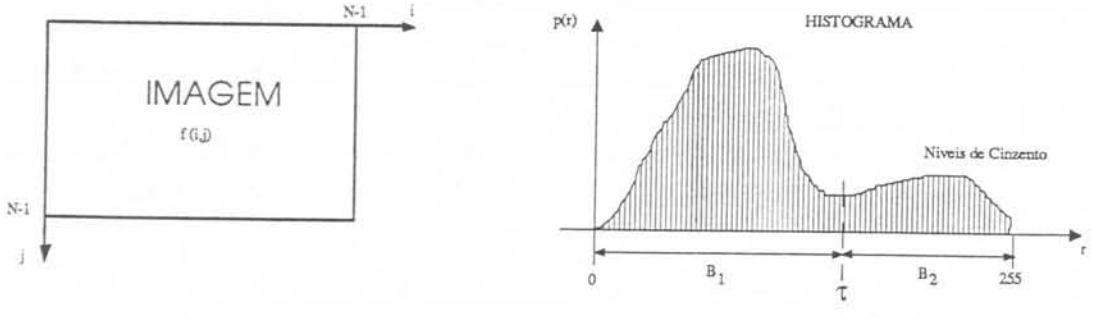


fig. 3.1 Representação da imagem discretizada e histograma dos níveis de cinzento

Na figura 3.1 são reconhecidos um conjunto de *pixels* com nível de cinzento *r* e n_r *pixels* com esse valor *'r'*. O número total de *pixels* presentes na imagem será dado por:

$$n = \sum_{r=0}^{l-1} n_r \qquad (3.1)$$

e a probabilidade de ocorrência de um nível de cinzento *'r'* na imagem, dado por

$$p_r = \frac{n_r}{n} \quad \text{com } p_r \geq 0 \text{ e } \sum_{r=0}^{l-1} p_r = 1 \qquad (3.2)$$

Considera-se que *'0'* corresponde ao nível de cinzento mais escuro na imagem e *'l-1'* ao nível de cinzento mais claro na imagem. Numa imagem de 2^8 níveis de cinzento $r_0 = 2^0 - 1$ e $r_{max} = 2^8 - 1$.

A segmentação da imagem por este processo, implica uma binarização que consiste na transformação da imagem de cinzentos *f(i,j)*, numa imagem de dois níveis de cinzento *g(i,j)*, à custa da condição (3.3) apresentada do modo seguinte

$$g(i,j) = \begin{cases} 1 \text{ se } r > \tau & \text{(pontos('pixels') do objecto)} \\ 0 \text{ se } r < \tau & \text{(pontos fora do objecto)} \end{cases} \qquad (3.3)$$

Se o valor do *patamar de segmentação* 'τ' depender exclusivamente do valor de *Pf(i,j)* a segmentação define-se como global. Se o valor do *patamar* 'τ' depender do valor de *f(i,j)* e de *(i,j)* a segmentação define-se como local. Se o valor do *patamar* 'τ' depender do valor de *(i,j)*, de *f(i,j)* e de *Pf*(i,j) a segmentação diz-se dinâmica.

No caso da binarização não assentar nos níveis binários '0' e '1', mas noutros valores da escala de cinzentos, para um *patamar de segmentação* da imagem '$\tau \in G$' e para um par binário de níveis de cinzento $B=\{ b_0 , b_1 \}$ com $b_0, b_1 \in G$, o resultado da

binarização da imagem *f(i,j)* pelo *nível de segmentação* 'τ ', é dado pela função binária (3.4) enunciada de modo seguinte

$$g(i,j) = \begin{cases} b_0 \text{ se } f(i,j) < \tau \\ b_1 \text{ se } f(i,j) \geq \tau \end{cases} \qquad (3.4)$$

3.2.2-2 Método de Forma

Supondo uma imagem constituída por vários objectos distintos, sobre um fundo (*background*) de cor diferente, o histograma para essa imagem deverá ser bimodal. Neste caso, o valor do *patamar da segmentação* deverá ser escolhido de modo que o seu valor corresponda ao vale do histograma. Este processo sendo simples não pode ser aplicado a imagens com picos muito desiguais e com vales longos e planos, dada a imprecisão que se observa ao fazer a sua escolha.

Um aperfeiçoamento deste algoritmo pode consistir na determinação do *"threshold"*(*nível de segmentação*), através do histograma de imagem, depois de excluidos todos os *pixels* com gradiente elevado. Ao excluir todos os pontos de gradiente elevado está-se a manter todos os restantes, que se encontram em áreas uniformes da imagem. Isto introduz melhorias na forma do histograma de imagem, obtendo-se níveis de separação entre picos do histograma muito maiores e portanto muito maior facilidade em os separar. O *"threshold"* é depois definido como anteriormente, nos vales situados entre os picos do histograma.

3.2.2-3 Método de análise de concavidades no histograma

Para imagens constituidas por objectos sobre fundo uniforme (*background*) é possível seleccionar o *nível de segmentação* 'τ ' pelo método das concavidades.

Considerando $(g_0, g_1, \ldots, g_{l-1})$ o conjunto de níveis de cinzento da imagem e $H(g_r)$ a altura do histograma, onde $H(g_r) \neq 0$ para todos os r presentes na imagem.

Para determinar as concavidades do histograma, constrói-se o menor polígono convexo possível $\overline{HS}$, ligando os picos e vales do histograma. As concavidades de H são determinadas através das diferenças entre o histograma H e os

limites do polígono convexo *HS(H*=HS-H)*. Os potenciais níveis de segmentação são os máximos locais $H^*(g_r)$, onde $H^*(g_r)$ é a altura de $\overline{HS}$ menos H para o nível cinzento g_r respectivo.

Nem todos os máximos são bons candidatos a níveis de segmentação, já que pode haver ruídos introduzidos por concavidades alargadas, pelo que têm de ser tomadas medidas adicionais para anular estes máximos espúrios.

As imagens das fig. 3.2 e das fig. 3.3 representam respectivamente, uma imagem de ensaio com madeira de pinho, e a imagem correspondente obtida por segmentação[1]. O nivel de segmentação *("threshold")* foi obtido pelo método das concavidades.

De salientar que para este caso particular e com este método das concavidades, o nível de segmentação por *"threshold"* é demasiado elevado, tornando a imagem binária muito escura, e portanto apresentando deficiências de descriminação em relação ao que se pretendia distinguir na imagem. Isso deve-se particularmente à forma do histograma, representado na fig. 3.4, que tem uma componente muito elevada junto dos valores extremos da escala de cinzentos da imagem, em consequência da sobreiluminação da mesma. Isto acarreta a existência de uma grande concavidade no histograma, próximo do nível de cinzento 255 (valor máximo das abcissas do histograma). A profundidade deste vale é portanto determinante para a fixação do nível de segmentação (*threshold*) em 249 pelo método das concavidades, comprometendo o resultado que se procurava obter.

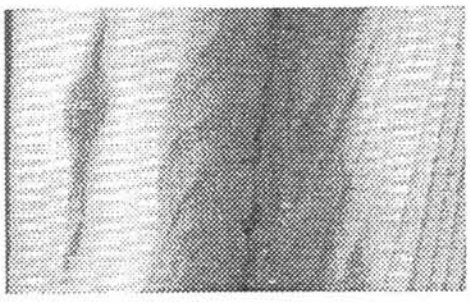


fig. 3.2 Imagem de prancha de madeira

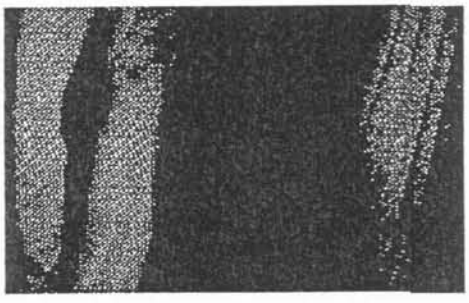


fig. 3.3 Imagem binária correspondente à da fig 3.2 depois de segmentação

[1]O algoritmo utilizado implementa o cálculo do threshold por análise de concavidades do histograma.As funções utilizadas nesse processo são: thistogm(...) - que calcula a distribuição de pixels, em TFUNC.C ; trphist(...) que calcula as concavidades e o threshold;trshold(...) - que faz uma conversão de unidades no histograma, que é apenas uma operação secundária que funciona como ferramenta, e bnthresd(...) - que efectua a binarização da imagem, ambas em TFUNCOES.C

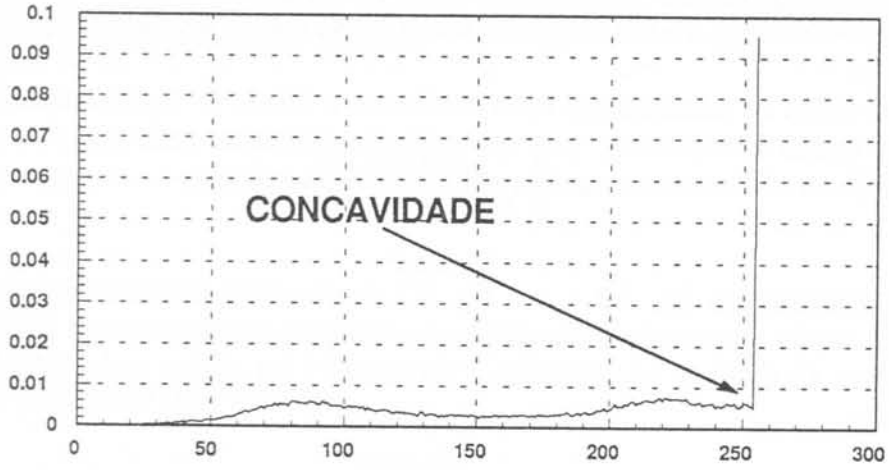


fig. 3.4 Histograma da imagem da fig. 3.2

3.2.2-4 MÉTODO DE NOBUYUKI OTSU

Este método não paramétrico e não supervisão, apresentado por Otsu[21-B] para a determinação do *patamar de segmentação* de imagens, utiliza apenas momentos de ordem zero e de primeira ordem do histograma dos níveis de cinzento da imagem. A extensão a problemas multinivel de segmentação podem ser alcançados com facilidade por este método.

Este método é fundamentalmente indicado para situações que não possam ser resolvidas pelo método anterior, ou seja onde os vales e os picos não sejam claramente visíveis. Tem a vantagem de utilizar o histograma dos níveis de cinzento da imagem e o método ser de aplicação completamente geral.

.

O patamar de segmentação óptimo é seleccionada automaticamente e de uma forma estável, não se baseando em processos de diferenciação (ou seja em propriedades locais, como um vale) mas em processos de integração (propriedades globais) do histograma. A estimação da média de cada uma das classes separadas, e a separabilidade entre as essas classes pode ser calculada por este processo.

Com este método, a operação de determinação do patamar de segmentação é vista, como a partição da imagem em duas classes, C_0 e C_1 por um patamar de segmentação *'T'= r*.

$$C_0 = \{ 0, 1, \ldots , r\} \text{ e } C_1 = \{r+1,r+2, \ldots , l-1\} \qquad (3.4)$$

σ_W^2, σ_B^2 e σ_T^2 representam respectivamente as variâncias intra classes, variância entre as classes e a variância total da imagem, e 'r' representa, neste caso, o nível de cinzento que estabelece a partição na imagem.

O patamar de segmentação 'T'= r, pode ser calculado através da minimização de uma das seguintes expressões

$$\lambda = \frac{\sigma_B^2}{\sigma_W^2} \qquad \eta = \frac{\sigma_B^2}{\sigma_T^2} \qquad \kappa = \frac{\sigma_T^2}{\sigma_W^2} \tag{3.5}$$

A probabilidade de ocorrência duma classe e a média dessa classe, estabelecida a partição através de r, é dada pelas seguintes expressões:

$$w_0 = \Pr(C_0) = \sum_{i=0}^{r} p_i = w(r)$$

$$w_1 = \Pr(C_1) = \sum_{i=r+1}^{l-1} p_i = 1 - w(r)$$

e

$$\mu_0 = \sum_{i=0}^{r} i \Pr(i|C_0) = \sum_{i=0}^{r} i p_i / w_0 = \mu(r) / w(r)$$

$$\mu_1 = \sum_{i=r+1}^{l-1} i \Pr(i|C_1) = \sum_{i=r+1}^{l-1} i p_i / w_1 = \frac{\mu_T - \mu(r)}{1 - w(r)}$$

onde

$$w(r) = \sum_{i=0}^{r} p_i$$

e

$$\mu(r) = \sum_{i=0}^{r} i p_i \tag{3.6}$$

verifica-se que

$$w_0\sigma_0 + w_1\sigma_1 = \mu_T \qquad \text{e} \qquad w_0 + w_1 = 1 \tag{3.7}$$

A variância das classes é dada por

$$\sigma_0^2 = \sum_{i=0}^{r} (i-\mu_0)^2 \Pr(i|C_0) = \sum_{i=0}^{r} (i-\mu_0)^2 p_i / w_0$$
$$\sigma_1^2 = \sum_{i=r+1}^{l-1} (i-\mu_1)^2 \Pr(i|C_1) = \sum_{i=r+1}^{l-1} (i-\mu_1)^2 p_i / w_1 \qquad (3.8)$$

Para avaliar o *"goodness"* do patamar de segmentação, estabelecido no nível de cinzento *r*, calcula-se o valor duma das funções definidas em (3.5), onde

$$\sigma_W^2 = w_0\sigma_0^2 + w_1\sigma_1^2 \qquad (3.9)$$

$$\sigma_B^2 = w_0(\mu_0 - \mu_T)^2 + w_1(\mu_1 - \mu_T)^2 \qquad (3.10)$$

$$\sigma_B^2 = w_0 w_1(\mu_1 - \mu_0)^2 \qquad (3.11)$$

e

$$\sigma_T^2 = \sum_{i=0}^{l-1} (i-\mu_T)^2 p_i \qquad (3.12)$$

A optimização do problema, consiste na procura do patamar de segmentação *T='r'*, que maximize a função objecto. Nesse pressuposto, o patamar que fornecer a maior separação entre classes, também será aquele que melhor separará as regiões na imagem (agrupados por *níveis de cinzento*).

η é a medida mais fácil de calcular em relação a 'r', logo é o critério mais simples de utilizar. Isso acontece porque σ_B é baseado em estatística de primeira ordem (médias) enquanto σ_w é baseada em estatística de segunda ordem. O nível de segmentação r^* que maximizar η, maximizará também σ_B^2

$$\eta = \sigma_B^2 / \sigma_T^2 \qquad \sigma_B^2(r^*) = \max \sigma_B^2(r) \qquad \text{com } i < r < l\text{-}1 \qquad (3.13)$$

A função objecto $\sigma_B^2(r)$ ou a equivalente medida de critério $\eta(r)$ é sempre bimodal, o que constitui uma vantagem do método e que implica a estabilidade do mesmo. O valor máximo de $\eta(r^*)$ pode ser utilizado para medir a separabilidade das classes.

A imagem da fig. 3.5 foi segmentada pelo processo acabado de descrever[2], e mostra a imagem binarizada que se obtém da fig. 3.2. Esta segmentação apresenta melhor qualidade do que aquela que se obtinha pelo método de segmentação por análise de concavidades.

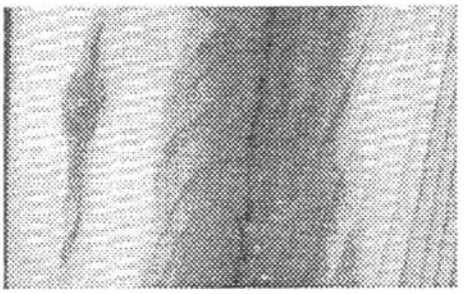


fig. 3.5 Imagem de prancha madeira

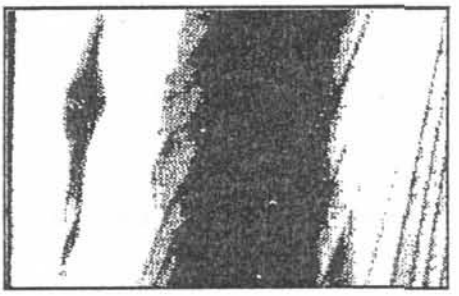


fig. 3.6 Imagem binarizada, por *nível de segmentação* determinado pelo método de Otsu

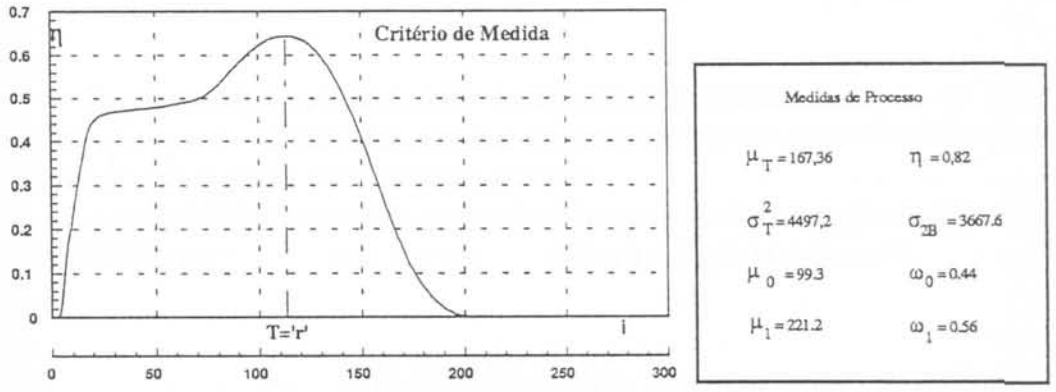


fig. 3.7 Critério de medida e resultados obtidos no processo de determinação do *"threshold"*

[2]O algoritmo utilizado implementa no cálculo do threshold, o método de Nobuyuki Otsu. As funções utilizadas nesse processo são: thistogm(...) - que calcula a distribuição de pixels na imagem, e lança os dados num ficheiro HIST1.TXT; a função vlsegm1(...) implementa o método retornando o valor da segmentação, lançando os dados de teste em HTO2B.TXT , no arquivo TFUNC.C; bnthresd(...) - efectua a binarização da imagem, em TFUNCOES.C

3.2.2-5 Métodos Utilizando Entropia

Existem vários processos de determinação do nível de segmentação a partir da entropia mas o método que vamos descrever foi apresentado por Kapur, Sahoo e Wong[3]. Com este método, derivamos da imagem original duas distribuições de probabilidades (correspondentes aos objectos a segmentar e ao fundo "*backgroud*"), da forma como se explícita em seguida:

$$\frac{p_0}{p_t}, \frac{p_1}{p_t}, \ldots, \frac{p_t}{p_t}$$
$$e \qquad (3.14)$$
$$\frac{p_{t+1}}{1-p_t}, \frac{p_{t+2}}{1-p_t}, \ldots, \frac{p_{t-1}}{1-p_t}$$

onde T é o valor do nível de segmentação e $P_t=\Sigma^t_{i=0} P_i$. Define-se

$$H_b(t) = -\sum_{i=0}^{t} \frac{p_i}{p_t} \log_e(\frac{p_i}{p_t})$$
$$H_w(t) = -\sum_{i=t+1}^{t-1} \frac{p_i}{p_t} \log_e(\frac{p_i}{p_t}) \qquad (3.15)$$

O nível de segmentação óptimo T^* é definido como sendo o nível de cinzento, que maximize $H_b(t) + H_W(t)$, ou seja

$$T^*=\text{ArgMax}\{ H_b(t) + H_W(t) \} \quad \text{em que } t \in G \qquad (3.16)$$

3.2-3 Procura dum Patamar Optimizado por Processos Paramétricos

Segundo processo proposto por Besançon[7-A], considere-se que num determinado histograma podem ser reconhecidos dois grupos de níveis de cinzento

[3]Citado em "A survey of thresholding Techniques", Sahoo P. K. e outros, referência [22-B] da bibliografia.

distintos, e que a probabilidade de ocorrência de um dado cinzento na imagem é dada por $P(r)$. Se a função densidade de probabilidade $P(r)$ poder ser considerada como a soma de duas densidades de probabilidades $P_1(r)$ e $P_2(r)$, então:

$$P(r)= P_1P_1(r) + P_2\,P_2(r) \tag{3.17}$$

onde $P_1(r)$ e $P_2(r)$ são as densidades de probabilidade do nível de cinzento r, P_1 e P_2 são as probabilidades à priori (probabilidade da existência dos dois níveis na imagem), com $P_1 + P_2 = 1$.

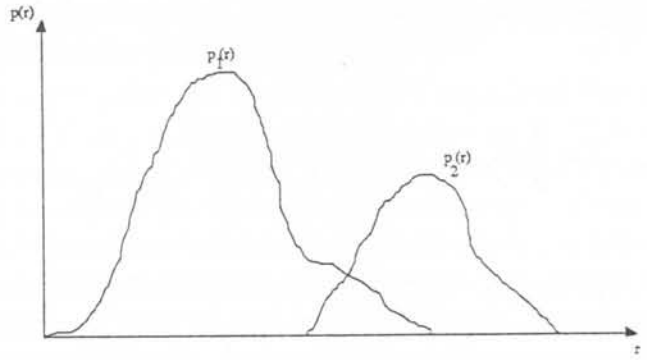


fig. 3.8 Curvas de distribuição de probabilidade em dois níveis

O patamar T divide a imagem em duas regiões, uma com níveis de cinzento menores do que T - zonas mais escuras da imagem e outra com níveis de cinzento maiores do que T, zonas mais claras da imagem.

A probabilidade esperada duma classificação errada de um *pixel* claro na zona escura, é dada por:

$$\varepsilon_1(T) = \int_{-\infty}^{T} P_2(r)\mathrm{d}r \tag{3.18}$$

A probabilidade esperada duma classificação errada de um *pixel* escuro na zona clara, é dada por

$$\varepsilon_2(T) = \int_{T}^{+\infty} P_1(r)\mathrm{d}r \tag{3.19}$$

A probabilidade total do erro é dada por

$$\varepsilon(T)=P_2\varepsilon_1(T)+P_1\varepsilon_2(T) \tag{3.20}$$

Como T é procurado para o erro mínimo $\varepsilon(T)=0$, encontra-se

$$P_2\varepsilon_1(T)=P_1\varepsilon_2(T) \tag{3.21}$$

A solução desta equação é dada para um T de abcissa mínima, comum a esses dois grupos dominantes. Se $P_1(r)$ e $P_2(r)$ forem duas distribuições Normais caracterizadas pelo valor das médias dos niveis de cinzento na imagem μ_1 e μ_2 e pelos desvios quadráticos σ_1 e σ_2., mostra-se que no caso de $\sigma_1 = \sigma_2.= \sigma$, T é único e igual a

$$T=\frac{1}{2}(\mu_1+\mu_2)+\frac{\sigma^2}{\mu_1-\mu_2}\ln(\frac{P_1}{P_2}) \tag{3.22}$$

3.2.4 Níveis de segmentação locais

3.2.4-1 Introdução

Numa segmentação local a imagem original é dividida em subimagens e o nível de segmentação é determinado para cada uma das subimagens. Isto leva à obtenção de uma imagem com descontinuidades nas fronteiras entre duas subimagens. Uma técnica de suavização pode ser aplicado para eliminar as descontinuidades. Os níveis de segmentação podem ser obtidos por métodos dependentes do ponto, ou da região.

Com este tipo de técnicas, pode não se determinar directamente o nível de segmentação. Em vez disso, transforma-se o histograma da imagem num outro equivalente, eventualmente em função doutras grandezas, com vales mais profundos e picos mais agudos, segundo as propriedades locais de cada *pixel*.

Chow e Kaneko[4] propõem a subdivisão da imagem em imagens de dimensão 7x7 e o cálculo do nível de segmentação para cada uma dessas subimagens. Contudo para subimagens com histogramas unimodais, o nível de segmentação é calculado por interpolação dos níveis de segmentação das subimagens fronteiras. Para as subimagens bimodais o nível de segmentação pode ser calculado por um processo paramétrico, por exemplo o descrito em 3.2-3.

Mason[19-A] sugeriu o uso de operadores de contorno("edges"), de modo que através deles pudessem ser estabelecidos valores ponderados para os *pixels* da imagem, que pudessem transformar o histograma.

O valor dos *pixels* sujeitos a operações de detecção de arestas, tomam valores grandes, em zonas com contornos nítidos e pequenos em regiões com níveis de homogeneidade grandes. A expressão (3.23) pode constituir um factor de ponderação do *pixel*, em que o valor do factor é inversamente proporcional ao gradiente(∇) do *pixel* nesse ponto da imagem

$$\frac{1}{(1+\nabla^2)} \qquad (3.23)$$

onde ∇ é o valor da aresta (*"edge"*).

Os *pixels* terão assim o seu valor transformado em função da sua posição na imagem, nomeadamente em função da sua situação em zonas homogéneas ou em zonas de declive acentuadas, o que implicará a transformação do histograma, traduzida na acentuação de vales e picos.

Outro método de transformação do histograma, é um método que utiliza árvores quaternárias (quadtree). Este método está baseado no facto de o desvio padrão em regiões homogéneas ser pequeno e em regiões não homogéneas ser elevado. Assim em regiões da imagem onde o desvio padrão em relação à média seja elevado, a imagem pode ser subdividida em quatro subregiões, de modo a que se encontrem regiões onde o desvio quadrático dos níveis de cinzento seja menor, ou seja subregiões com um grau de uniformidade maior. Este processo é repetitivo em cada um dos quadrantes obtidos.

O processo começa a partir da imagem total, que é dividida em quatro quadrantes, que são analisados em função da sua homogeneidade. Se o desvio padrão dos níveis de cinzento é superior a um determinado valor, a imagem volta a ser

[4]Citado em "A survey of thresholding Techniques", Sahoo P. K. e outros, referência [22-B] da bibliografia.

dividida nesse quadrante, sendo o processo repetido até que nenhum dos quadrantes possa mais ser subdividido.

Isto leva a que se obenha uma imagem decomposta da original, com desvios entre os *pixels* pouco significativos, o que torna razoável aproximar o nível de cinzento do bloco à média dos mesmos. À imagem obtida chama-se imagem-Q(*quaternária*). Devido à homogeneidade dos blocos, o histograma dos níveis de cinzento da imagem Q, terá picos maiores e vales mais profundos, de novo como nos métodos anterior estudados.

3.2.4-2 Níveis de segmentação Múltiplos

Um método que possibilita a determinação de vários níveis de segmentação foi já descrito, quando fizemos a descrição do método de OTSU. A determinação de vários níveis de segmentação numa imagem, permite separar várias regiões que ela contenha com níveis globais de cinzento que ela contenha.

A este propósito o método que aqui descreveremos diz respeito ao método de segmentação por amplitude. Este método proposto por Boukharouba[5], usa a função acumulada de probabilidades de uma imagem descretizada representada pelos seus níveis de cinzento. A curvatura da função de distribuição das probabilidades, é examinada em termos da procura dos seus máximos locais.

A função densidade de probabilidades acumulada $F(r)$ no ponto r é dada por:

$$F(r) = \frac{\sum_{i=0}^{r} H(i)}{\sum_{i=0}^{l-1} H(i)} \qquad (3.24)$$

e a curvatura de F é definida por

$$C(r) = F''(r)\,[1 + (F'(r))^2\,]^{-3/2} \qquad (3.25)$$

onde F' e F'' são a primeira e a segunda derivada de F respectivamente. Os zeros da curva determinam os níveis de segmentação a atribuir a cada classe.

[5]Citado em "A survey of thresholding Techniques", Sahoo P. K. e outros, referência [22-B] da bibliografia.

O método descrito por Wong e Haralick[16-B] utiliza uma técnica recursiva, para a determinação dos multiníveis de segmentação. Inicialmente os *pixels* são classificados como fazendo parte do conjunto de arestas ("edges") da imagem, ou como não fazendo parte desse grupo. Com base nos vizinhos destes *"edges pixels"* estes são classificados como claros ou escuros. Constrói-se depois dois histogramas dos níveis de cinzento, um para os edges claros e outro para os edges escuros. Um nível de segmentação é seleccionado baseado no nível de cinzento com o pico mais elevado de um destes histogramas. O processo é recursivo e permite, utilizando sucessivamente, primeiro só os níveis de cinzento superiores ao nível de segmentação seleccionado e depois os níveis de cinzento inferiores, retirar um conjunto de "thresholds" da imagem.

3.2.4-3 NÍVEIS DE SEGMENTAÇÃO VARIÁVEIS

Muitas vezes com a determinação de um só nível de segmentação 'τ ', não se evita que as regiões segmentados não apareçam com algum ruído introduzido, entendendo-se por ruído aquelas porções de imagem que não deveriam pertencer às regiões segmentadas, mas sim ao background. Supondo que podem ser definidos os níveis de segmentação τ_1 e τ_2, como sendo respectivamente o mínimo valor do objecto e o máximo valor do background, com $\tau_1 > \tau_2$.poderemos atribuir aos valores menores que τ_2 ao background e maiores que τ_1 ao objecto. Os valores entre τ_1 e τ_2, são atribuidos ao background ou ao objecto, consoante a atribuição maioritária que possa ser feita aos seus vizinhos.

Outra possibilidade, com alguma utilidade em determinadas aplicações particulares, consiste em fazer variar o "threshold" ao longo da imagem, em função de $\overline{f(i,j)}$ e σ(i,j)

$$\tau = K\sigma(i,j) + \overline{f(i,j)} \quad \text{com} \quad 1 \geq k \geq 0 \qquad (3.26)$$

σ(i,j) - desvio quadrático médio
$\overline{f(i,j)}$ - média dos f(i,j)

O "threshold" determinado por este processo, fica completamente relacionado com o valor dos *pixels* locais delimitados numa certa região.

Os métodos por *threshold* são métodos globais, enquanto os métodos baseados em crescimento de regiões, que aglomeram *pixels* com intensidades similares, são fundamentalmente locais.

Um outro processo que permite a extracção de um ou mais *thresholds* e que funde técnicas de crescimento de regiões com técnicas globais, pode ser descrito. Esse processo junta regiões homogéneas e é repetido recursivamente até que sejam encontrados grupos (ou *clusters*). As regiões são depois separadas por *threshold*, do modo seguinte:

Parte-se de I_0 , a imagem original

1-faz-se K=1

2-Gera-se uma imagem I_k, juntando pontos adjacentes cuja diferença de intensidade seja menor do que k, em regiões com intensidades reduzidas às médias.

3-Calcula-se o histograma H_k da imagem I_k.

4-Testa-se no histograma se I_k tem clusters nitidos. Se isto se verificar executa 5, senão faz k=k+1 e repete 2 (k pode ser incrementado de valores maiores que 1).

5-Retirar o(s) threshold(s) do histograma e binarizar a imagem.

3.2-5 Técnicas Baseadas em Divisão e Conquista ("split and merge"). Crescimento de Regiões

Por crescimento de regiões, entende-se todo aquele processo que permite englobar *pixels* de uma imagem na mesma região, se forem adjacentes e tiverem pelo menos uma propriedade comum. A segmentação de regiões por um processo de crescimento a partir de uma semente, consiste na junção de regiões previamente segmentadas (ou na separação de regiões-divisão de regiões), até que se atinja o grau de uniformidade suficiente (à luz de um critério determinado) ou por processos mistos envolvendo a junção e a divisão de regiões.

Num processo de crescimento, parte-se de uma semente e vão-se anexando *pixels* similares na região. No processo de segmentação por divisão parte-se a imagem global, e vai-se dividindo a imagem com base em diferenças.

fig. 3.9 Grupos de *pixels* R_1, R_2, R_3, R_4 classificados por crescimento de regiões

A conectividade entre *pixels* pode ser medida de várias maneiras, mas a mais frequente é através do conceito de mínima distância. Tem importância para o efeito a noção de conectividade, já que a ela está associada a noção de vizinhança. Considera-se a conectividade de dois tipos: *conectividade quatro* e *conectividade oito*.

fig. 3.10 Conectividade

Dizemos que dois *pixels* f_i e f_j são conexos na região R se:

1 - Existir uma sequência $\{f_i , \ldots , f_j\}$ em que os *pixels* adjacentes estejam ligados por uma propriedade

2 - Todos os pontos na sequência pertencerem à região R.

3 - A região R é conexa se todos os pares de pontos em R estiverem conectados/ligados pelo mesmo atributo.

O objectivo dos algoritmos de segmentação por crescimento de regiões(divisão/aglomeração), é a partição da imagem I_0, em conjuntos de *pixels* pertencentes aos subconjuntos $\{R_1, R_2, \ldots ; R_n\}$, tal que

Os conjuntos deverão ser disjuntos(a intersecção entre R_i e R_j deve ser vazia)

$$R_i \cap R_j = \varnothing \quad \text{para} \quad i \neq j \tag{3.27}$$

A união de todos os R_i deverá ser I_0, correspondente à imagem total

$$I_0 = \bigcup_{k=1}^{n} R_i \tag{3.28}$$

Para agrupar os *pixels* numa região, uma medida de similitude (ou dissemelhança) é requerida, e pode ser baseada em características, tais como: intensidade, cor, textura local, ou outras medidas.

Podemos definir uma função Booleana H(R) que meça a homogeneidade de uma região. A região é considerada hogénea se:

$$\begin{cases} H(R_k) = Verdade & para\ todo\ o\ k \\ H(R_i \cup R_j) = Falso & para\ todo\ o\ i \neq j \end{cases} \tag{3.29}$$

Uma função H baseada só na intensidade do *pixel*, podia ser definida de acordo com a expressão (3.30), em que o *threshold T* definido não precisava de ser fixo, podendo ser actualizado ao longo do processo de segmentação

$$H(R_i) = \begin{cases} \text{verdade} & \forall \text{pixel } (i,j) \text{ com conectividade 8, e } (i,j) \in R_i \\ & (i^*, j^*) \in R_i \quad \text{se} \quad |f(i,j) - f(i^*, j^*)| \leq T \\ \text{Falso} & \text{em caso contrário} \end{cases} \tag{3.30}$$

Se $H(R_i)$ = falso então um ou mais *pixels* não pertencem à região R_i

Se $H(R_i \cup R_j)$ = verdade, então as regiões R_i e R_j não se podem distinguir e devem por isso ser fundidas(*"merged"*).

Na segmentação por divisão, a imagem global é o ponto de partida inicinicial. A imagem é recursivamente dividida em subregiões e o critério de homogeneidadeidade é aplicado a essas regiões. Quando o critério for satisfeito para todas as regiões, es, ou um número previamente estabelecido for atingido, o processo termina.

Um processo de segmentação misto dos anteriores, elaborado por DivDivisão e Conquista na imagem, pode ser esquematizado segundo os critérios expostos os como se segue

Um critério de homogeneidade H e uma metodologia de fusão ão são estabelecidas.

O algoritmo segue os seguintes passos:

Repetir o processo, enquanto for possível efectuar mais fusões ou divisõvisões

{

Repetir Para todas as regiões R estabelecidas

Se $H(R_i)$ = falso
Dividir R_i
Se $H(R_i , R_j , R_k , \ldots)$ = verdade
então juntar R_i , R_j , R_k, ... em regiões de maior dimensânsão

}

Posso definir os critérios de homogeneidade como:

$$H(R) = \begin{cases} V & \text{Para todos os pares de pontos tais que } |f(i,j) - f(i^*,j^*)| < T \\ f & \text{em caso contrário} \end{cases}$$

(3.33.31)

Para implementar este algoritmo com a imagem, pode ser necessário usar uma estrutura especial, que envolva relações de natureza hierárquica em forma de pirâmide. A estrutura escolhida pode basear-se em árvores quaternárias, como foi já focado em 3.2.3-1, em que cada região mantém uma relação de ascendência com as quatro subregiões que lhe sucedem.

Se para qualquer região R da estrutura, H(R) é falso, divido R em quatro subregiões. Se para as quatro subregiões, R_{k1}, R_{k2},R_{k3}, R_{k4} é verdadeira a propriedade H,,

$$H(R_{k1} \cup R_{k2} \cup R_{k3} \cup R_{k4}) = V \qquad (3.32)$$

junto as quatro numa única região. Se existirem regiões vizinhas R_i e R_j tal que $H(R_i \cup R_j) = V$, juntar essas regiões.

Quando não for possível fazer mais divisões ou fusões, o algoritmo termina.

3.3 SEGMENTAÇÃO BASEADA NA DETERMINAÇÃO DO CONTORNO

3.3.1 INTRODUÇÃO

Na segmentação por extracção de contornos, o que se procura obter é sempre a fronteira, que forçosamente qualquer região tem obrigatoriamente que conter. As fronteiras são sempre referidas face ao fundo onde estão inseridas. Esses contornos podem ser extraíveis com mais ou menos facilidade, dependendo do contraste entre a região e o fundo em que se insere, e da complexidade dessa mesma fronteira.

As arestas são consideradas as fronteiras entre segmentos de imagem. Se aplicarmos à imagem um detector de arestas (*"edges"*), podemos ficar a conhecer as fronteiras entre regiões. Contudo estes operadores apenas produzem uma medida do valor do *"edge"* num ponto, e por esse motivo terá de ser arranjado um processo que ligue os segmentos em contornos de fronteira contínuos.

O processo deve basear-se nos passos seguintes:

1 - Calcular o gradiente em cada ponto da imagem. Duas informações podem ser obtidas: A amplitude das arestas e a direcção das mesmas.

2 - O operador gradiente produz *"edges"* muito pesados (valores do gradiente elevados, aliado a muitos *"edges"* próximos). O processo precisa de ser corrigido através do adelgaçamento (*"thining"*) do mapa de *"edges"* obtidos. Um processo utilizado, consiste em aproveitar apenas os edges que correspondam a valores máximos locais do gradiente, na sua direcção. Os vizinhos com a mesma direcção ou direcções próximas, são anulados, enquanto o *"edge" 'e'* é incrementado de alguma quantidade δ .

3 - Para determinar as arestas (ou limites) da região de fronteira, cada *"edge"* seleccionado será ligado com os vizinhos posterior e anterior (o processo pode ser difícil porque um destes ou ambos, poderão ser nulos).

Se e'_i for o vector ortogonal ao gradiente e_i no ponto i, e'_i apontará na direcção da fronteira no ponto i. A escolha do *pixel* de ligação, recairá sobre aquele vizinho que tiver o *"edge"* mais forte e com a direcção mais próxima do anterior.

4 - A ligação entre cadeias de *"edges"* formando segmentos de contorno continuos, obtidos das regiões, conseguem-se ligando os pontos terminais dessas cadeias, ou seja o ponto terminal anterior com o ponto terminal posterior da cadeia seguinte. Dessa forma é possível organizar conjuntos de fronteiras continuas delimitando os objectos na imagem. Algumas fronteiras abertas, poderão ser fechadas através da extensão dos seus pontos terminais

5 - A eliminação de algumas cadeias pode tornar-se necessária, em casos que possam ficar obscurecidos pela profusão de contornos muito próximos ou com formas específicas. Podem ser usados critérios de eliminação de cadeias, como sejam: eliminação de cadeias muito curtas ou com um raio de curvatura muito pequeno. Podem ser usados outros critérios, como o do

contraste médio ao longo da cadeia, ou um critério de forma, como o do perímetro ou área envolvida.

3.3-2 Extracção de Fronteiras Através de Pontos do Contorno (*"edges"*)

Intensidade dos *"Edges"*

Os edges estão localizados em locais onde a intensidade luminosa muda abruptamente. Seja $f(i,j)$ a intensidade luminosa no ponto (i,j).O módulo e a direcção do gradiente em (i,j) são definidos por:

$$|e(i,j)| = \sqrt{f_i(i,j)^2 + f_j(i,j)^2}$$
$$\angle e(i,j) = \mathrm{tg}^{-1}\frac{f_j(i,j)}{f_i(i,j)} + \frac{\pi}{2} \qquad (3.33)$$

onde f_i e f_j são a primeira derivada na direcção[6] de i e de j respectivamente.

Há muitos operadores que podem calcular estas derivadas, mas a forma mais simples de o fazer é através das expressões descritas abaixo:

$$f_i(i,j) = f(i+1, j) - f(i-1, j) \qquad (3.34)$$

$$f_j(i,j) = f(i, j+1) - f(i, j-1) \qquad (3.35)$$

Os *"edges"* fracos, menores do que um threshold determinado τ são eliminados. Para eliminar edges provocados por ruídos, faz-se uma supressão dos "edges" que não são máximos locais, do seguinte modo: Todos os edges que tenham um valor menor do que os seus dois vizinhos, na direcção perpendiculares à sua são suprimidos. Outra alternativa é a utilização da segunda derivada. O *edge* é detectado

[6]Sistema de coordenadas, segundo as direcções esquematizadas na fig3.1.

numa posição chamada de (*zero-crossing*) cruzamento do zero, onde a sa segunda derivada muda de sinal, a que corresponde a um máximo da primeira derivrivada. A soma das segundas derivadas segundo a direcção dos *iis* f_{ii} e segundo a direcçecção dos *jjs* f_{jj} é chamado Laplaciano.

Em resultado da supressão dos edges fracos, não só são suprimidosdos edges provenientes de ruído, mas também edges que fariam parte do contorno, no, o que acarreta como consequência o aparecimento de cadeias incompletas. Para corcompletar estas cadeias podem ser seguidos os seguintes critérios:

1 - Pegar numa aresta terminal incompleta (e_0) e iniciar a extensão.

2 - Procurar entre os três edges adjacentes a e_0, aquele que estiver mais próximo da direcção de e_0. Entre os três seleccionar aquele que maximize a função

$$f(e_0, e_i) = |e_i| \cos(\angle e_i - \angle e_0) \qquad (3.33.36)$$

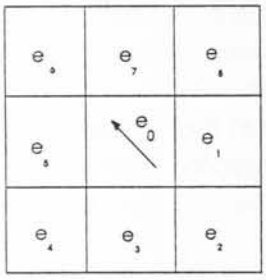


fig. 3.11 Extensão do edge e_0. A seta indica a direcção de e_0.

$f(e_0,e_i)$ dará o maior valor, quando $|e_i|$ for grande e a direcçãcção de $\angle e_i$ for igual à direcção de $\angle e_0$.

3 - Se e_i for um *edge* suprimido voltar a reactiva-lo, actualizar o edge e_0 que passa a ser o e_i anterior voltar ao ponto 2.

4 - Se for um *edge* vivo, parar a extensão e voltar a 1.

Se quiser fazer a extensão de uma aresta, na direcção oposta ao edge, faço como adjacentes de e_0 aqueles que estiverem na direcção oposta à direcção corrente de e_0. Se uma série de edges muito fracos forem reavivados, é preciso testar se são ou não maiores que um threshold τ , e em caso afirmativo abortar a extensão do edge.

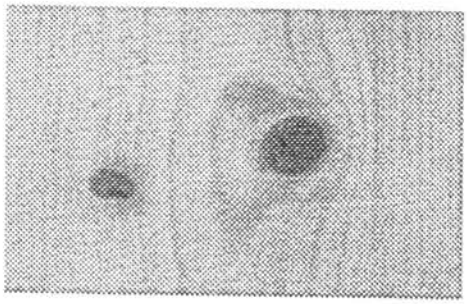


fig. 3.12 Madeira com nós

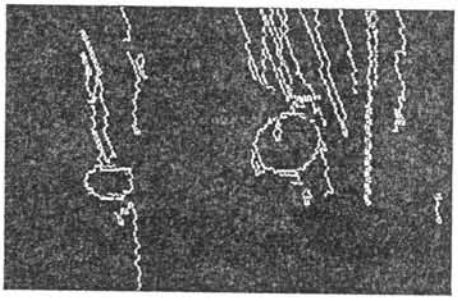


fig. 3.13 Contornos das regiões presentes na fig. 3.12, extraídos através de processo de detecção de orlas

A imagem da fig. 3.13 representa o contorno das regiões visíveis na fig. 3.12, através do processo descrito em 3.3.2[7]. Como se pode ver a extracção do contorno dos nós e de veios presentes na madeira, apresenta deficiências grandes, quer a nível da definição, quer a nível do fecho desses mesmos contornos. Isso mostra, que este processo de extracção (e do fecho) dos contornos, não é fácil e nem sempre é o mais adequada a todos os tipos de aplicações (como no caso das madeiras).

3.4 Segmentação por análise de texturas (região de texturas)

Existem vários processos de segmentação de imagens através das propriedades de textura contidas nas suas regiões. Serão aqui descritos apenas três desses processos.

[7]Os contornos da imagem da fig. 3.13, foram obtidos através da função sbel_ft1(...), desnvolvida no arquivo TFILTR1.C. Esta função apresenta dificuldades operacionais, principalmente ao nivel dos segmentos do contorno perdidos, e ligadas à definição da fronteira como uma linha continua e fechada

Se uma imagem não for caracterizada só pela cor (intensidade) dos seus *pixels*, mas também por múltiplas outras propriedades (textura ou topológicas), o processo de extracção de regiões pode ser efectuado pelo seguinte método:

1 - Calcula-se uma das múltiplas propriedades em cada ponto.

2 - Faz-se o mapeamento de todos os pontos, no espaço multidimensional das características, e classificam-se por agrupamentos em nuvens ("*clusters*").

3 - Atribui-se cada ponto a um aglomerado específico, e aglutinam-se os pontos adjacentes etiquetados da mesma maneira.

Outros processos utilizam mecanismos recursivos, por exemplo através da utilização de múltiplos histogramas unidimensionais, como no exemplo seguinte:.

1 - Seleciona-se a imagem não segmentada (inicialmente toda a imagem)

2 - Determina-se o histograma para todos as múltiplas propriedades definidas na imagem ou em regiões já identificadas e em estudo.

3 - Seleciona-se a propriedade que mostre a melhor separação entre picos nos histogramas.

4 - Segmenta-se a imagem, apenas em função da propriedade que foi seleccionada.

5 - Consideram-se as novas regiões conexas e repete-se o processo até que nenhum pico seja detectado em nenhum histograma.

Este processo apresenta desvantagens, já que se a imagem contiver regiões, em que uma das propriedades caia em duas gamas disjuntas do histograma, a segmentação pode dividir essa região. Um outro processo que procura evitar esta dificuldade, descrito por Tomita e Tsuji[31-A], é descrito a seguir.

Procura-se à volta de (i,j), uma área que contenha pontos só de uma região. Se uma fronteira nessa região for uma curva não muito fechada ou uma recta, pelo menos uma das cinco áreas vizinhas à volta de (i,j), $N_{2n}(i,j)$, $N_{2n}(i+n,j+n)$, $N_{2n}(i-n,i+n)$, $N_{2n}(i-n,j-n)$, $N_{2n}(i+n,j-n)$ fica inteiramente contida numa região homogénea.

Calculam-se então os índices de não homogeneidade para cada uma destas cinco vizinhanças, através da aplicação de operadores gradiente à partição quaternária dessas áreas - como o indicado na fig. 3.8-b), seleccionando-se aquela com menor índice de não homogeneidade. Uma melhor imagem para segmentação é gerada através do cálculo das propriedades de textura obtidos das áreas seleccionadas.

O procedimento, para a imagem I_0,envolve os seguintes passos:

1 - Fazer k=1

2 - Gerar uma imagem A_k, pelo cálculo de propriedades de textura, na vizinhança n de (i,j) $N_n(i,j)$, em cada ponto da imagem, a partir de I_{k-1}

3 - Selecionar uma área com o menor índice de homogeneidade, calculada a partir das cinco áreas definidas em volta do ponto, na imagem I_{k-1}.O índice de não homogeneidade de uma área $N_{2n}(p,q)$ é calculado por:

$$\begin{aligned} &| A_k(p+n/2,q+n/2) + A_k(p-n/2,q+n/2) - A_k(p+n/2,q-n/2) - A_k(p-n/2,q-n/2) | \\ &+ | A_k(p+n/2,q+n/2) + A_k(p+n/2,q-n/2) - A_k(p-n/2,q+n/2) - A_k(p-n/2,q-n/2) | \end{aligned} \qquad (3.37)$$

4 - Gerar uma nova imagem I_k, em que o valor do novo ponto, é a média do valor da textura determinado para a área $N_{2n}(p,q)$ seleccionada, no ponto correspondente em I_{k-1}.O valor da textura média é calculada por:

$$\{A_k(p+n/2,q+n/2) + A_k(p-n/2,q+n/2) + A_k(p+n/2,q-n/2) + A_k(p-n/2,q-n/2)\}/2)\}/4$$

(3.33.38)

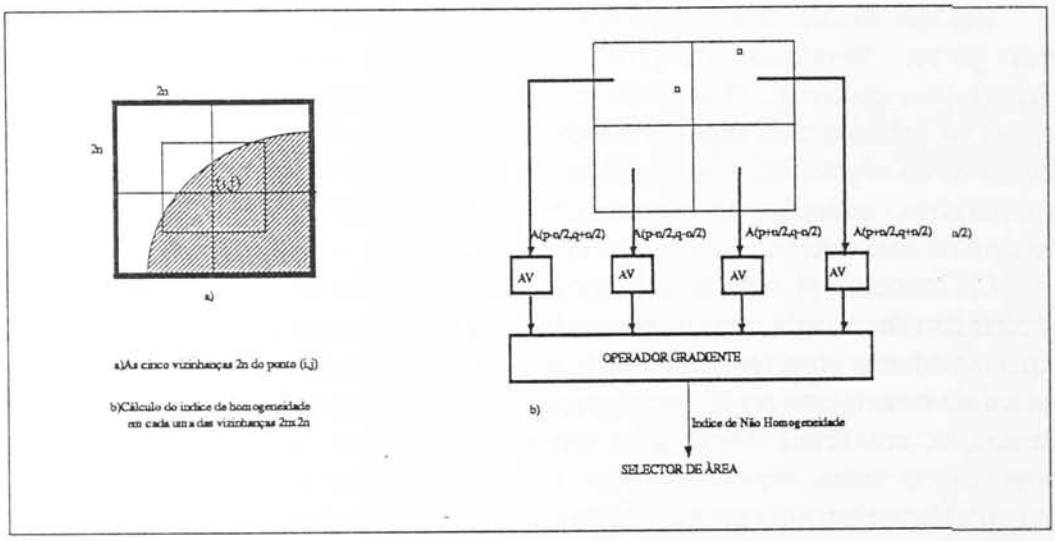


fig. 3.14 Vizinhanças de um ponto e cálculo dos índices de não homogeneidade numa vizinhança

5 - Testar se todos os picos do histograma I_k são agudos. Se forem, executar o ponto 7, senão fazer k=k+1

6 - Criar uma imagem de médias A_k, numa vizinhança n de (i,j) - $N_n(i,j)$, para cada ponto da imagem I_{k-1}, e executar 3.

7 - Binarizar a imagem I_k pelos valores do *"threshold"* retirados os do histograma, através do valor entre picos. São assim extraidas as as regiões da imagem.

Outro processo possível para fazer a segmentação por textura, consiste em etiquetar toda a área, em que mais de 50% esteja coberta pela mesma *propriedade de textura*, juntar em seguida regiões adjacentes com a mesma *etiqueta de textura* (ou conexas). Considerar a região conexa, sempre que tiver propriedades similares de intensidade ou textura.

3.5 CODIFICAÇÃO DE IMAGEM BINÁRIA BIDIMENSIONAL POR *QUADTREE*

Uma técnica baseada em árvores com estrutura quaternária, descrita entre outros por Besançon em [7-A], é utilizada na representação de imagens binárias[branco/preto]. Não sendo um processo de segmentação, é por vezes uma estrutura utilizada para suportar imagens binárias obtidas por segmentação. Como o processo de segmentação apresentado em 3.6, se refere a estruturas do tipo *árvore quaternária*, utilizadas no armazenamento de suporte da informação, justifica-se fazer aqui uma referência a este tipo de estrutura.

Considera-se uma árvore estruturada por níveis equivalentes aos níveis de uma pirâmide. A cada nível correspondem regiões homogéneas ou não homogéneas. A um quadrante cinzento corresponde uma região não homogéneae e a um quadrante preto ou branco uma região homogénea. No primeiro nível todos os quadrantes são cinzentos, nos níveis intermédios, todos os quadrantes são cinzentos (brancos e pretos), ou todos pretos ou todos brancos. No nível mais alto, a pirâmide é constituída exclusivamente por quadrantes brancos e quadrantes pretos.

Um quadrante (num nível intermédio) é branco ou preto, consoante os pixels na imagem correspondente, são todos brancos ou todos pretos. Se nesse quadrante alguns *pixels* são brancos e outros pretos, o valor assumido na árvore em correspondencia com esse quadrante, assume o valor cinzento.

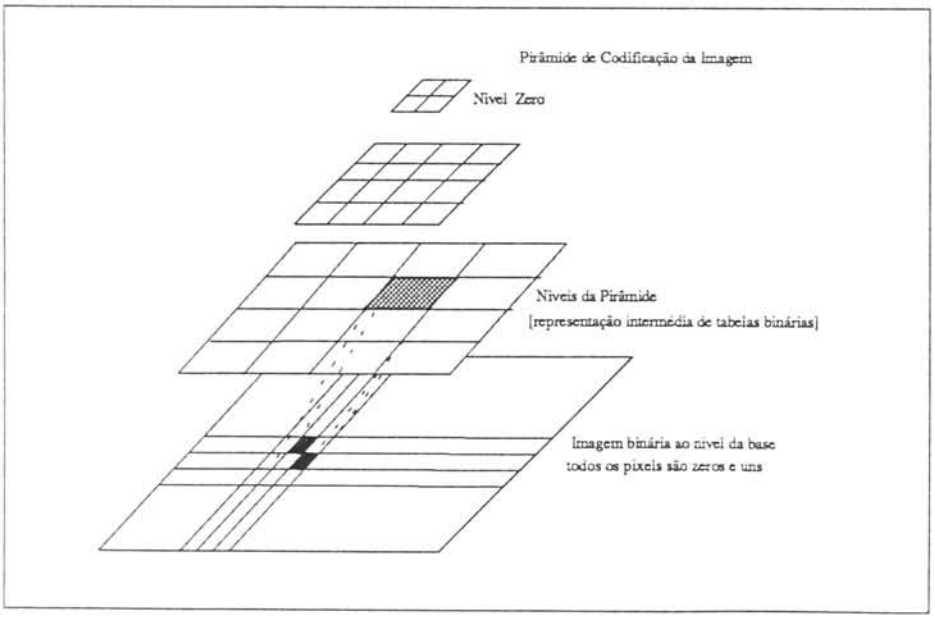


fig. 3.15 Pirâmide de codificação de imagem binária

A árvore é construída de uma forma recursiva, desde a raiz até à base, fazendo corresponder a cada nível da árvore, um estrato da pirâmide. Se um quadrante é inteiramente branco ou inteiramente preto, este quadrante forma um nó terminal ou folha. Se o quadrante não é uniforme, forma um nó cinzento, que por conseguinte tem de ser dividido em quatro subquadrantes.

Este algoritmo exige que a imagem seja quadrada, constituída por $2^n x 2^n$ pixels, produto da altura da árvore vezes a largura, permitindo que a imagem seja dividida no máximo até *'n'* níveis

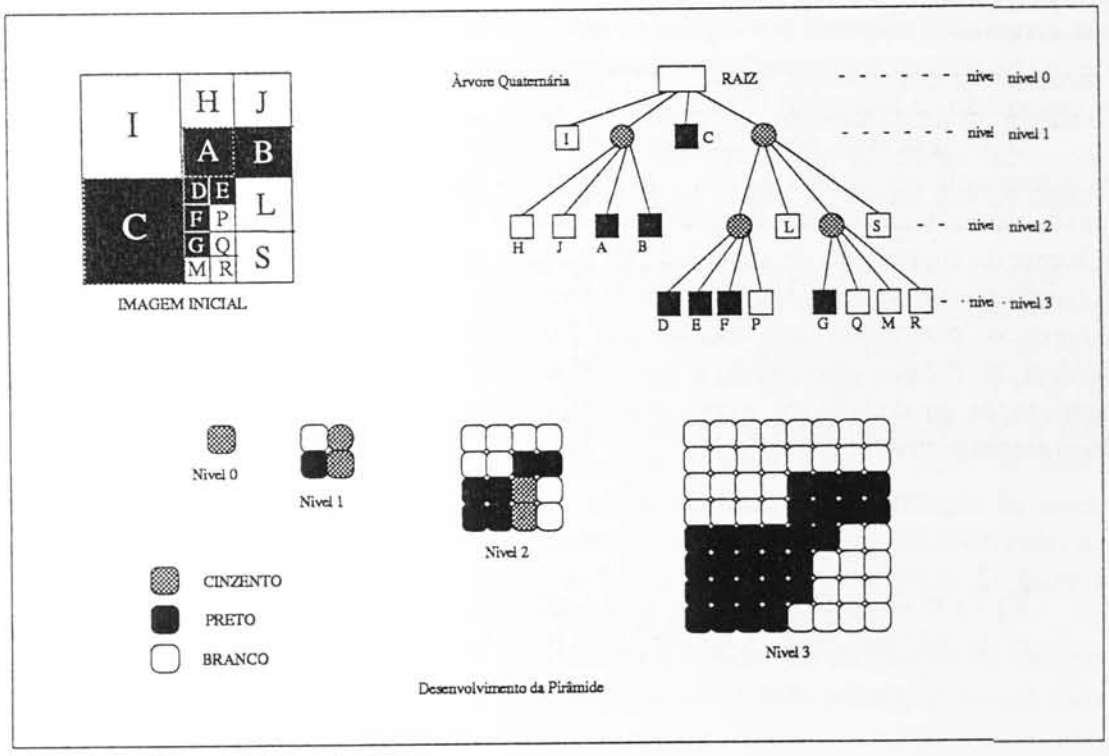


fig. 3.16 Representação de uma imagem binária por árvore quaternária

A fig. 3.16 mostra como pode ser estruturada a informação numa uma imagem binária, através da utilização de árvores quaternárias (*"quadtree"*).

3.6 EXEMPLO DE SEGMENTAÇÃO, UTILIZANDO TÉCNICAS DE DIVISÃO DA IMAGEM POR REGIÃO, SERVINDO-SE DE ESTRUTURA QUATERNÁRIA DO TIPO ÁRVORE

O processo de segmentação utilizado neste exemplo, explora algumas das técnicas descritas nas páginas anteriores. Utiliza em primeiro lugar uma técnica recursiva de divisão da imagem em quatro partes, por forma a pesquisar na imagem as suas regiões homogéneas. Esta técnica utilizada sucessivamente, permite chegar a uma imagem mapeada por regiões de cor uniforme, facilitando a operação posterior, que consiste na segmentação por threshold (método das concavidades 3.2.2-3 e [22-B] ou método de Nobuyuki Otsu 3.2.2-4 [21-B]).

O processo pode ser descrito do modo seguinte: A imagem é dividida inicialmente em quatro partes, e analisada do ponto de vista da sua uniformidade, através de uma técnica de gradiente. Sempre que seja detectada uma região não uniforme o processo de subdivisão da imagem em quatro partes continua, até que a uniformidade seja encontrada. Das quatro partes (ou blocos) em que é dividida a imagem, é extraída uma medida que caracterize cada um dos blocos o melhor possível. Para o caso citado a característica utilizada é a média de cinzentos ou o contraste, em cada bloco, e com base nessa medida é determinado o gradiente, para o bloco respectivo.

$$\nabla_J = | A_k(i+n/2\ j+n/2) + A_k(i-n/2\ j+n/2) - A_k(i+n/2\ j-n/2) - A_k(i-n/2\ j-n/2) | + | A_k(i+n/2\ j+n/2) + A_k(i+n/2\ j-n/2) - A_k(i-n/2\ j+n/2) - A_k(i-n/2\ j-n/2) |$$

(3. 38)

A_k representa o valor médio dos níveis de cinzento na área (ou secção) de imagem correspondente ao nível k da árvore, centrado no ponto $(i+/-n/2, j+/-n/2)$, e ∇_J representa a medida do gradiente na região considerada, para esses quatro blocos.

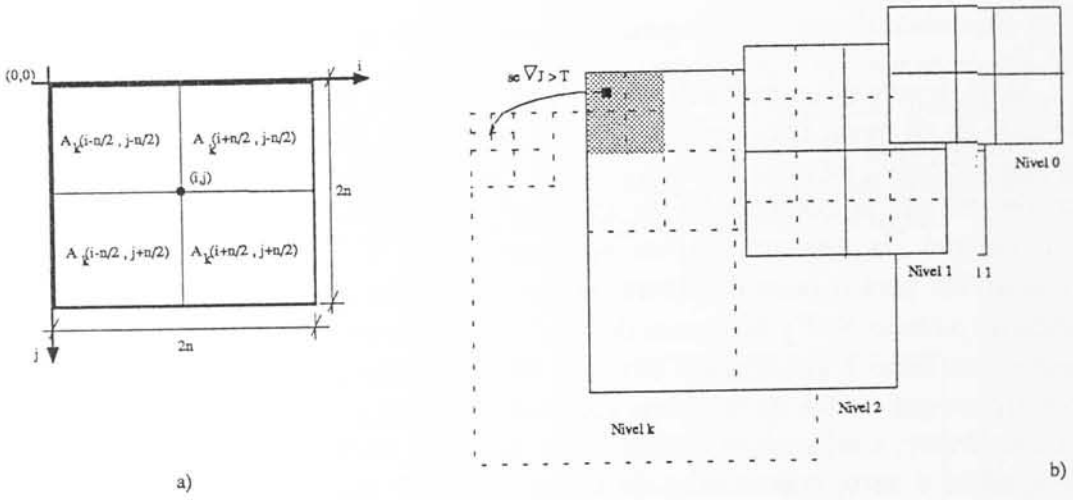


fig. 3.17 Processo de divisão da imagem em a), e arquitectura de divisão da árvore quaternária

Se ∇_J for maior do que um (*"threshold"*) "τ ", num determinado bloco de imagem, esta é dividida em quatro novas partes. O valor do centroide dessa região e a média, são armazenados numa estrutura equivalente à de um nó duma árvore quaternária (*quadtree*), já que este tipo de estrutura se adapta à recolha dos dados que constituem a informação.

Para se poder achar o gradiente, é necessário conhecer em cada caso, o centro do bloco de imagem e a sua dimensão (largura x altura). Essa informação está armazenada em cada nó da árvore. Cada nó possui uma estrutura como a que está representada no esquema da fig. 3. 17, que armazena além do centroide da área de imagem em consideração, a média dessa mesma área e o nível da árvore a que corresponde a subdivisão (a raiz da árvore situa-se ao nível zero). A partir dessa informação é possivel determinar a largura e altura do bloco. A informação nivel, permite através de tabela[6], saber qual a dimensão n x m do bloco de subdivisão da imagem. Pelo facto de não ser quadrada (da forma $2^n x 2^m$), não se pode fazer a divisão da imagem de um modo regular, em todos os níveis da árvore.

Este processo tem a vantagem de aumentar a rapidez de divisão da imagem em áreas homogéneas, uma vez que não se trabalha sobre toda a imagem em cada nível de implementação.

mi média do sector	Nivel nivel de subdivisão arvore (entrada da tabela A[8][2])	coordyj cordenada j do centroide	coordxi cordenada i do centroide	esqit ponteiro/esquerda alta	esqbx ponteiro/esquerda baixa	dirit ponteiro/direita alta	dirbx ponteiro/direita baixa

fig. 3.18 Esquema representando a estrutura de um nó da árvore quaternária

[6]Uma tabela em Arv_q2(...), permite para cada nível de profundidade da árvore, saber a r a dimensão dos blocos em que deve ser dividida a imagem. Isto é necessário pelo facto de se estar a trabalhar com uma imagem rectangular, onde a sua dimensão não é múltipla duma potência de 2 (2^k para k níveis).

A árvore é construída de acordo com o esquema representado na fig. 3.19, e onde cada nó tem a estrutura representada no esquema da fig. 3.18. Possui em cada nível intermédio, informação sobre o nível a que corresponde o bloco, o centroide do bloco (ou seja as coordenadas do seu centro), e a média do bloco. Os ponteiros constituintes da estrutura de um nó apontam, se ∇_J for maior do que "τ " (*threshold*), para o ramo da àrvore respectivo, um dos quatro em que o bloco foi dividido o bloco. Se ∇_J for menor do que "τ " o bloco é b considerado uniforme e o respectivo ramo é considerado terminal, ou seja, passa a constituir uma folha da árvore em que todos os ponteiros apontarão para NULL. Refira-se que no caso da raiz da árvore, a informação contida no nó da árvore correspondente, traduz-se por: nivel igual a zero, coordenadas do centro do bloco (neste caso toda a imagem) [160,200], e média de toda a imagem. [7]

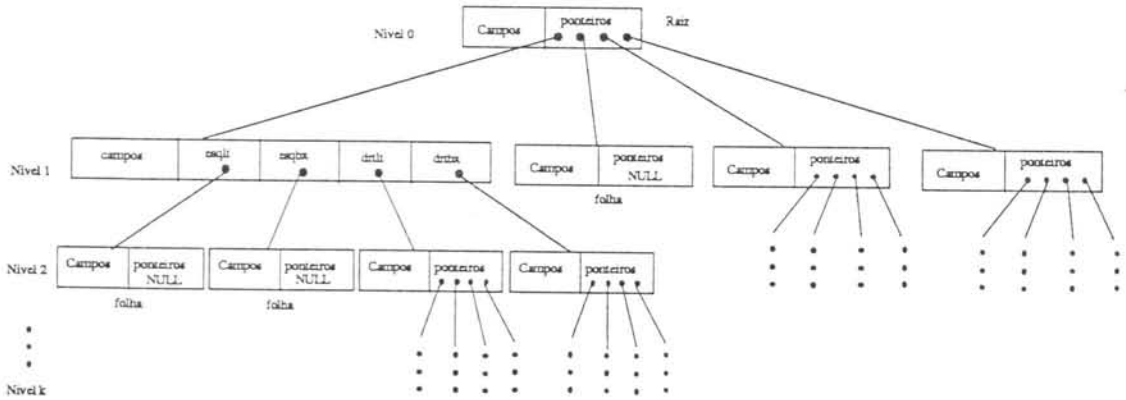


fig. 3.19 Organização simplificada e parcial da estrutura de uma árvore quaternária, utilizada na retenção de informação do algoritmo de segmentação por *"quad tree"*

Depois da árvore estar completamente organizada, a informação fica retida nas suas folhas, e pode ser recuperada por leitura. Essa informação corresponde à média das intensidades luminosas dos blocos das regiões consideradas homogéneas, e ao seu respectivo centroide.

Essa informação é utilizada para efectuar a segmentação da imagem, por *threshold*, já que o seu histograma apresenta agora maiores possibilidades para ser dividido. O histograma desta nova imagem criada, apresenta picos e vales mais nítidos, como se pode ver na fig. 3.24. Além de se terem eliminado ilhas dispersas de ruído, que apareceriam na imagem segmentada pelos processos directos, torna-se

[7]Os programas foram desenvolvidos em C, no sistema operativo MS-DOS, microprocessador 80286 pelo que o endereçamento directo só pode ser feito para segmentos de 64kbytes, no máximo. Por essa razão todas as imagens foram utilizadas no formato 320x200(colunas x linhas) e um cabeçalho de 4 bytes, num total de 64 004 bytes.

mais fácil encontrar os níveis de segmentação adequados. Essa facilidade para eliminar zonas dispersas nas regiões de segmentação, pode ser acentuada ou diminuida através da variação de valor do *threshold* 'τ ' escolhido quando da definição de homogeneidade para as regiões (utilizado na avaliação do gradiente).

O algoritmo esquematizado na fig. 3.20, mostra através de um diagrama de blocos, o processo utilizado para implementar este processo de segmentação.

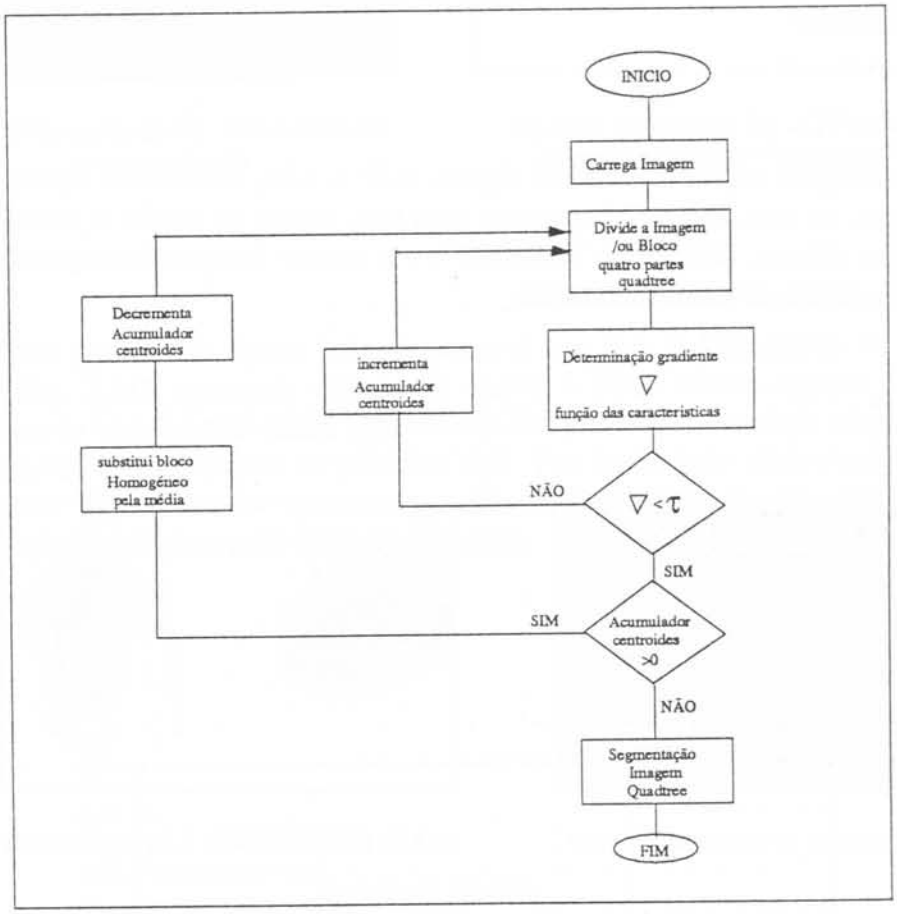


fig. 3.20 Diagrama de blocos, mostrando o processo de implementação do algoritmo de segmentação, por divisão da imagem e utilização de quadtree

O programa de implementação segue o esquema apresentado na fig3.19, que utiliza as funções indicadas na nota[8], que desempenham as tarefas aí descritas.

[8]Os programas que executam a segmentação representada no diagrama de fluxos do esquema da fig. 3.19 estão desenvolvidos no arquivo RBSEGQT.C(segmentação por quadtree) e envolvem as funções: le_t(...)- que chama as funções de construção da árvore, de leitura da árvore, e de eliminação da árvore. A função Arv_q2(...)-constrói a árvore quaternária(versão 2), analisando as

Este processo de segmentação tem a vantagem de permitir dividir a imagem em regiões, não em função da distribuição de cinzentos de toda a imagem, mas em função das diferenças nas distribuições de cinzento analisados localmente, ou seja em função da sua uniformidade de cor ou textura local. A variável testada é uma grandeza exclusivamente dependente das intensidades luminosas (médias das intensidades) de cada ponto na imagem, mas poderia ter sido utilizada outra grandeza textural qualquer.

As imagens representadas nas figuras 3.22 a 3.24, mostram a operação de segmentação, de uma imagem de madeira com nós, depois de sujeita a processo de segmentação directo, através de *"threshold"*, e a mesma imagem homogeneizada e segmentada depois de homogeneização.

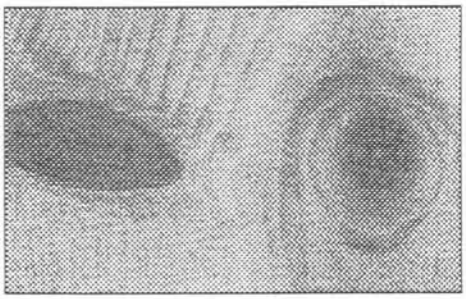


fig. 3.21 Imagem de madeira com nós

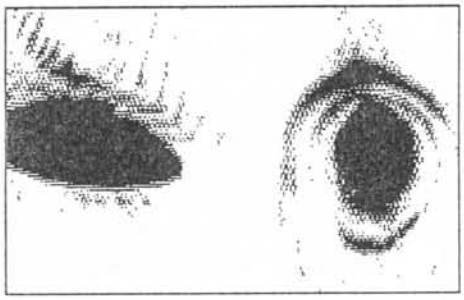


fig3.22 Imagem da fig. 3.21 segmentada por *"threshold"*

---

condições de homogeneidade através da análise do gradiente descrito na pg28?.A função readtree(...)-lê a imagem e faz a sua aquisição para um buffer, e dlttree(...)-apaga a estrutura árvore(delete tree).A estrutura dum nó da árvore é a seguinte

```
struct quadt_e
{
        unsigned char md,nivel,coordyi;
        int coordxj;
        struct quadt_e  *esqlt,*esqbx,*drtlt,*drtbx;
}
```

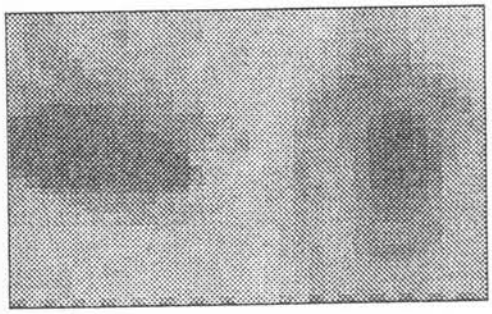


fig. 3.23 Imagem da fig. 3.21 depois de homogeneizada

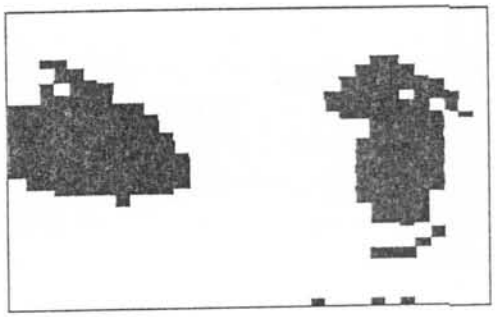


fig. 3.24 Imagem da fig. 3.23 depois de segmentada por *"threshold"*

Pelo gráfico da figura 3.25, é possível ver que o histograma da imagem original (fig. 3.21), apresenta vales mais suaves e picos menos agudos do que o histograma da imagem suavizada. Isso reflecte dificuldades que podem surgir ao tirar o *"threshold"* numa imagem do primeiro tipo. Por outro lado observa-se também, uma maior dispersão de pequenas regiões sem significado no resultado da segmentação de imagem pelo primeiro processo.

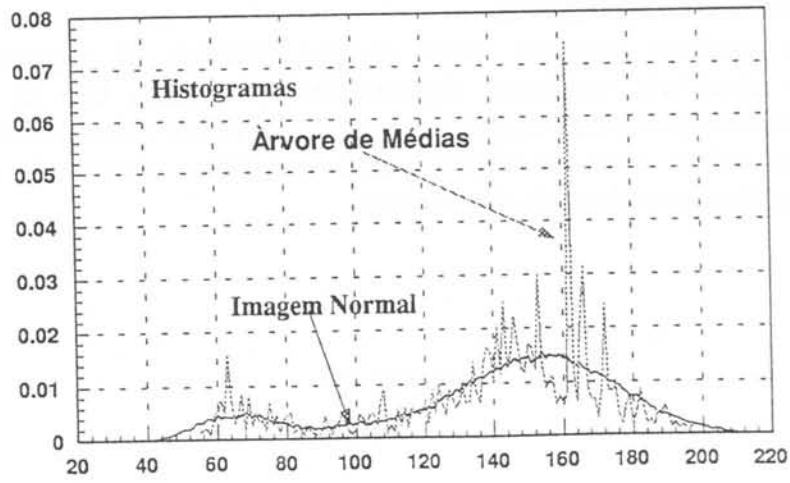


fig. 3.25 Histogramas das imagens representadas nas fig. 3.21 e fig. 3.23.

F.E.

3.7 Conclusões

A segmentação apresenta uma grande relação com a análise, na medida que pode fundamentar a divisão duma imagem em regiões, com a extracção de variáveis características dessa imagem. De salientar que o inverso também é verdadeiro. De facto, pode segmentar-se uma imagem, com o objectivo de lhe retirar características representativas.

Neste capítulo procurou fazer-se um resumo de algumas das técnicas e tipos de segmentação, utilizados em processamento de imagem. O objectivo da segmentação consiste em delimitar nas imagens, regiões de superfície homogénea, ou contornos contínuos e fechados dessas regiões.

Nessa descrição envolvendo processos de segmentação, não nos baseamos exclusivamente em processos de partição da imagem (decomposição e análise), mas também em princípios de aglomeração, junção de regiões ou de porções de imagem com características semelhantes.

Os processos descritos por Otsu, *"threshold"* por concavidade, e segmentação depois de homogeneização da imagem, foram os que se revelaram para o caso das madeiras com maior consistência e por esse motivo, utilizados na implementação dos algoritmos de *localização e reconhecimento de defeitos em madeira*, abordados no capítulo VII.

Outras técnicas descritas, como as que trataram da segmentação por extracção de contornos, revelaram algumas limitações neste tipo de aplicação. Alguns processos foram apenas enunciados com objectivos de enquadramento, já que não é possível referir todas as técnicas, uma vez que o número de algoritmos propostos pela literatura adequada é vastíssimo.

CAPÍTULO IV

OPERAÇÕES SOBRE IMAGENS BINÁRIAS

MORFOLOGIA E CONTORNOS

RUI BEBIANO

4 - Tratamento de imagens binárias. Operações Lógicas Contornos e Morfologia

4-1 Introdução

Este capítulo trata de assuntos relacionados com a manipulação, e com a recolha de variáveis (padrões ou primitivas) a partir de imagens binárias. A operação com imagens binárias, deve-se à simplicidade da sua análise, e ainda ao facto da informação contida nestas imagens ser frequentemente relevante. Entenda-se por imagem binária, toda aquela imagem que contenha apenas dois níveis de cinzento ou de cor.

A utilização de imagens binárias, permite extrair padrões da imagem que podem ser úteis como variáveis características no plano da sua classificação. Nessas imagens a extracção de contornos, áreas e formas é muito mais clara e simples, já que se trabalha apenas com dois níveis de cinzento '0' (nível de cinzento 0) e '1' (nível de cinzento 255, valor máximo). Com estes dois níveis '0' e '1' as operações Booleanas elementares (AND, OR, NOT), as operações geométricas, morfológicas e topológicas, são fáceis e rápidas de implementar e por vezes de extrema utilidade.

Este capítulo faz uma síntese de algumas técnicas de operação com imagens binárias, como sejam as operações lógicas entre imagens pixel a pixel, operações de morfologia binária, como sejam: dilatação, erosão, fecho e abertura, e operações topológicas, envolvendo a determinação de propriedades características associadas à relações de posição dos elementos constituintes da imagem.

4.2- Operações com Imagens Binárias. Operações Booleanas Locais

As operações lógicas (Booleanas) AND, OR, XOR, EOR, NOT, com imagens binárias podem ser representadas pelas tabelas seguintes:

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

A	B	A EOR B
0	0	1
0	1	0
1	0	0
1	1	1

quadro 4-1 Operações lógicas "AND" "OR" e "EOR"

A	NOT A
0	1
1	0

quadro 4-2 Operação "NOT"

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

quadro 4-3 Operação "XOR"

Estas operações booleanas permitem transformar imagens binárias noutra(s) imagens binárias, através de operações lógicas elementares, segundo a forma descrita pelas tabelas anteriores, que por vezes se revestem de grande utilidade. A fig. 4.1 ilustra algumas destas operações binárias, através da utilização de duas imagens esquemáticas, representadas por 'A' e 'B'

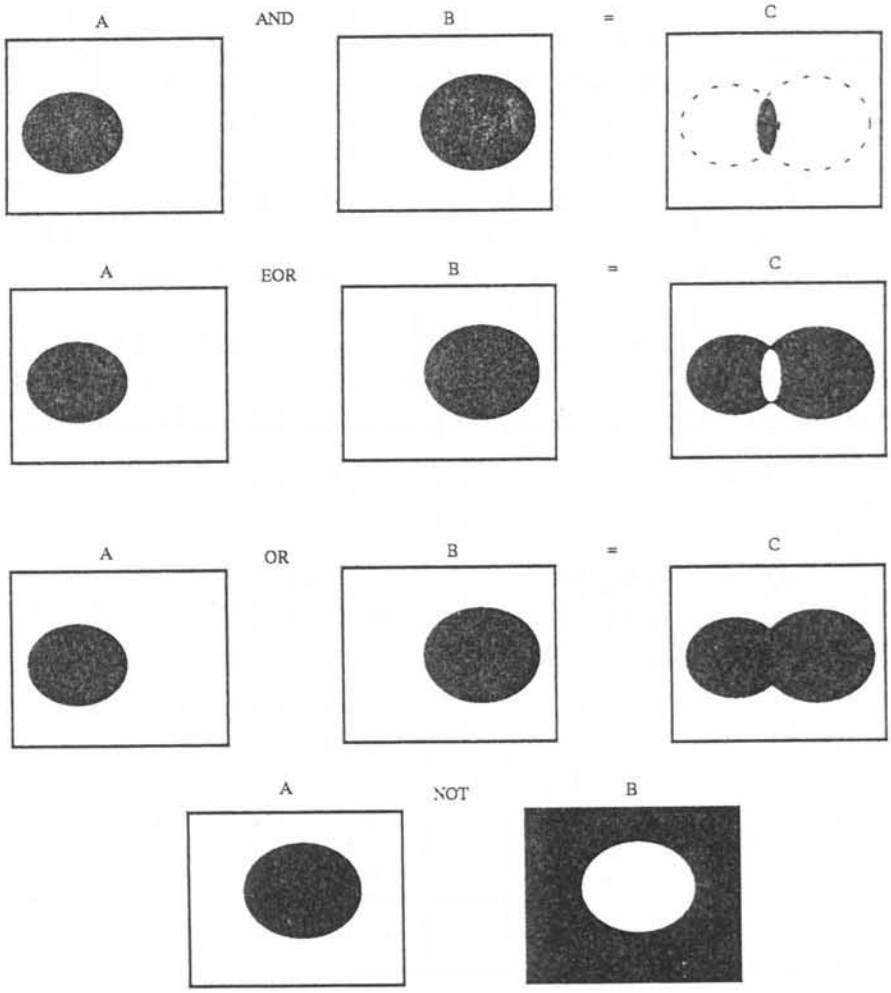


fig. 4.1 Operações booleanas sobre imagem

As operações booleanas locais, são o ponto de partida para algumas formas de processamento de imagens binárias. As operações booleanas podem ser comparadas a uma convolução, na medida que podem ser entendidas como operações envolvendo uma máscara, aplicada a cada ponto da imagem, mas em vez de operações aritméticas são efectuadas operações lógicas.

Segundo Mike James[20-A], uma operação booleana local pode ser escrita na forma seguinte:

$$a_{ij} = f(a_{ij}, \text{com todos os seus vizinhos}) \qquad (4.1)$$

f - representa uma das funções booleanas seguintes:
AND , *OR*, *NOT* , *EOR* , etc

Nestas operações podem-se considerar como vizinhos de um pixel, janelas de dimensão 3x3 centradas no pixel em consideração, sobre os quais vão ser efectuadas as operações em causa.

A título exemplificativo, são apresentados nos parágrafos seguintes, dois conjuntos de operações booleanas, que podem ser efectuadas sobre uma imagem: a primeira permite retirar uma fronteira e a segunda retira pixels isolados numa imagem binarizada.

A operação que permite extrair um pixel da fronteira numa imagem binária, através de operações booleanas, pode ser determinado do modo abaixo indicado, considerando que um pixel de fronteira, tem pelo menos um pixel vizinho pertencente ao *"background"*.

$$B = (NOT\ a_1)\ OR\ (NOT\ a_2)\ OR\ (NOT\ a_3)\ OR\ (NOT\ a_4)\ OR\ (NOT\ a_5)\ OR\ (NOT\ a_6)\ OR\ (NOT\ a_7)\ OR\ (NOT\ a_8) \quad (4.2)$$

a_1	a_2	a_3
a_8	a_0	a_4
a_7	a_6	a_5

fig. 4.2 Representação dos vizinhos de a_0, numa vizinhança 3x3

B só seria '1' (pixel interior) se todos os vizinhos de a_0(=1) fossem iguais a '1'. A operação que retira a fronteira é dado por

$$a_0\ AND\ B \quad (4.3)$$

Os vizinhos de a_0, sendo considerados com conectividade 8, retiram uma fronteira com conectividade 4($P4$), se os vizinhos de a_0 tiverem conectividade 4, retiram uma fronteira com conectividade 8($P8$). B é obtido neste caso por:

$$B = (NOT\ a_2)\ OR\ (NOT\ a_4)\ OR\ (NOT\ a_6)\ OR\ (NOT\ a_8) \quad (4.4)$$

Para encontrar e remover pontos isolados na imagem, provavelmente provocados por ruído, pode ser conseguido através de operações binárias. Um pixel isolado é aquele que não tem nenhum vizinho que pertença à região.

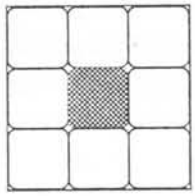


fig. 4.3 Vizinhança 3x3 dum pixel isolado numa imagem

Para o caso de um pixel isolado, todos os vizinhos pertencem ao *"background"*. Uma função local que pode retornar '1' nestas circunstâncias, e zero em todos as outras, é dada por:

$$B = (NOT\ a_1)\ AND\ (NOT\ a_2)\ AND\ (NOT\ a_3)\ AND\ (NOT\ a_4)\ AND\ (NOT\ a_5)\ AND\ (NOT\ a_6)\ AND\ (NOT\ a_7)\ AND\ (NOT\ a_8) \qquad (4.5)$$

Para remover este pixel isolado $a_0(=1)$, quando B for um, aplico a operação Booleano dada por:

$$a_0\ AND\ NOT\ B \qquad (4.6)$$

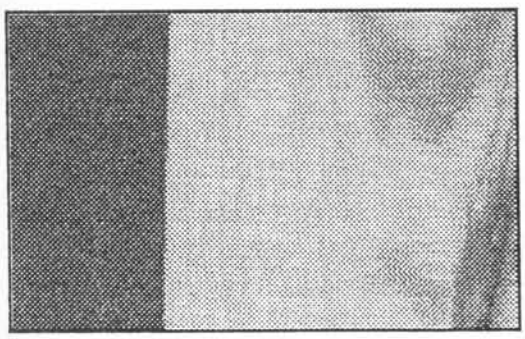


fig. 4.4-a) Prancha de pinho com *background* e brecha de resinagem

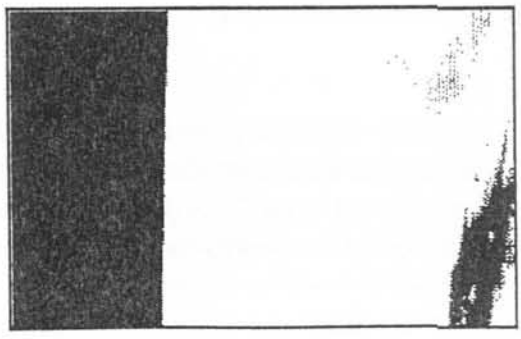


fig. 4.4-b) Prancha depois de binarização

As imagens representadas na figura 4.4, procuram ilustrar o efeito obtido pela aplicação de operadores lógicos[1] a uma imagem, neste caso uma operação de eliminação de ruído sobre a imagem. É de salientar que os resultados obtidos em 4.5-c) são pouco significativos(visíveis), em relação à imagem binarizada alvo 4.4-b), já que esta apresenta poucos pixels isolados (pixels pretos com oito vizinhos brancos), uma vez que este tipo de ruído tem geralmente uma proveniência exterior à imagem. Melhores resultados são obtidos em 4.4d), pela a aplicação de um algoritmo, utilizando um processo morfológico de eliminação de regiões espúrias (não compactas) dispersas na imagem.

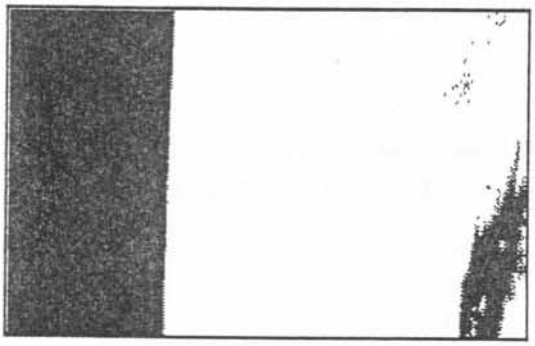


fig. 4.4-c) Prancha depois de operação de eliminaçãode pixels isolados

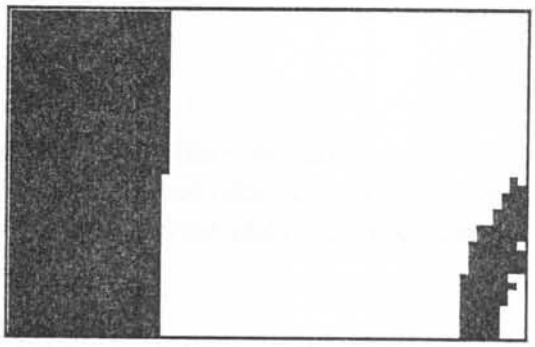


fig. 4.4-d) Prancha de madeira depois de operação morfológica de eliminação de zonas espúrias

4.3 PROPRIEDADES DAS IMAGENS BINÁRIAS

4.3.1 CONECTIVIDADE

Numa subregião/objecto duma imagem binária, é possível distinguir dois tipos de pixels. Pixels interiores que não têm nenhum vizinho fronteiriço que faça parte do *"background"*, e pixels da fronteira que têm pelo menos um vizinho imediato com esse *"background"*. Estes pixels de fronteira formam um contorno

[1]A imagem da figura 4.4-d, foi obtida por eliminação de ruído pontual (através da relação apresentada em (4.6), a partir de 4.5-b, a sua correspondente imagem binária com ruído. A função utilizada foi suav(...), implementada no arquivo TCLEAN.C.

A imagem 4.4-d, foi obtida de 4.4-b, através de operações lógicas sobre a imagem, utilizando máscaras 3x3, e juntando um critério de optimização, que consistia em colocar toda a região 3x3 da imagem com a mesma cor, se a cor dominante ocupasse mais de 70% da área ocupada pela máscara. A função utilizada foi limpz1(...) em TCLEAN.C.

(*"binary edges"*), cuja forma depende bastante do critério usado na definição do que se entende por vizinho mais próximo. Um critério possível seria estabelecer que um pixel da fronteira está conectado com o background, enquanto um pixel interior não está.

Segundo Anil Jain[1-A], Mike James[20-A] e outros, existem dois tipos de conectividade[2]: *conectividade oito*, onde um pixel está ligado aos seus oito vizinhos mais próximos e *conectividade quatro*, onde um pixel está ligado aos seus quatro vizinhos mais próximos (segundo a direcção horizontal e segundo a direcção vertical). As figuras 4.5 e 4.6, mostram esses dois tipos de conectividade.

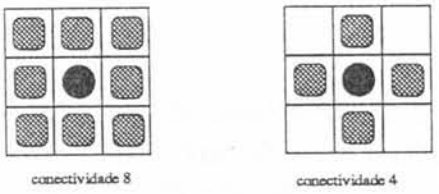


fig. 4.5 Estabelecimento de *conectividade oito* e *conectividade quatro* dum pixel

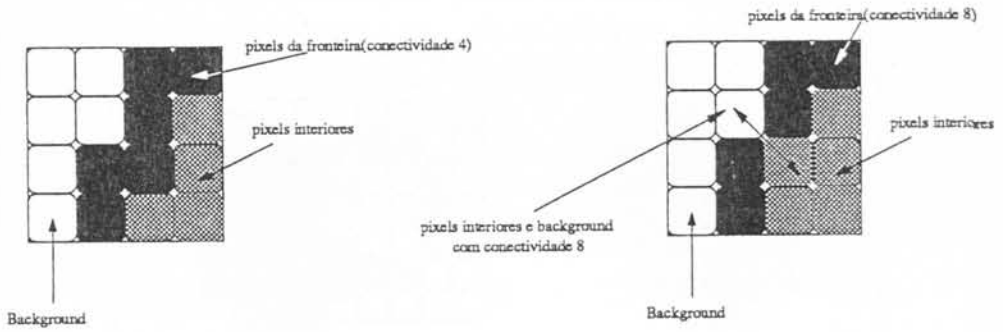


fig. 4.6 Fronteiras com *conectividade quatro* e *oito* respectivamente

Na descrição simbólica de uma imagem binária, representada por um conjunto de pixels com valores '0' ou '1', um dos aspectos mais relevantes, prende-se com a descrição da relação geométrica, ou conectividade, entre os pixels supostamente pertencentes à mesma região. Esses pixels pertencentes à mesma região, dizem-se conexos, uns em relação aos outros.

[2]A conectividade é implementada pela função conectividade(...), em TSEGMT3.C, operação que foi executada no caso da imagem da fig. 4.8.

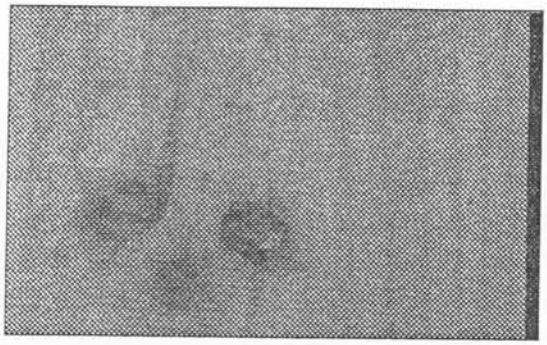


fig. 4.7 Imagem de prancha de madeira com nós

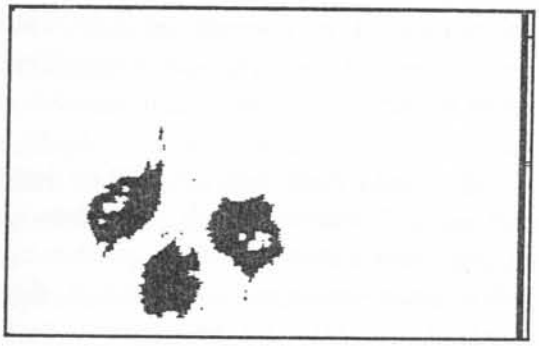


fig. 4.8 Imagem depois de eliminados os pixels não conexos na imagem binarizada

Há situações, onde a conectividade, ou as relações entre pixels se tornam ambíguas, o que obriga sempre à definição adicional de conjuntos de propriedades *S*, que são depois atribuídas aos pixels, e com as quais se procura desfazer a ambiguidade. Ou seja, são definidas primitivas descritivas, e estabelecidas relações para cada pixel com os seus vizinhos, baseados em relações de brilho, cor, textura, ou outras.

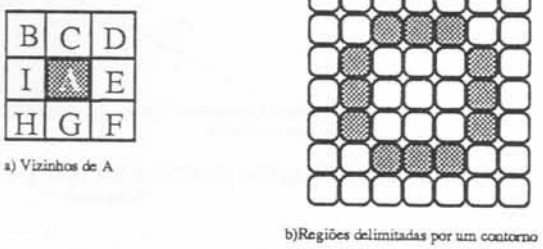


fig. 4.9 Outro modo de ilustração da conectividade

O pixel *A* pode ser potencialmente conectado com os pixels *I*, *C*, *G*, *E*, no caso da conectividade quatro, ou com os pixels I, *C*, *G*, *E*, *B*, *D*, *F*, *H*, no caso da conectividade oito. Posso definir os pixels como conexos, se todos eles tiverem o mesmo conjunto de propriedades.

A figura 4.9-b mostra um contorno, que pode delimitar duas regiões, uma interna e outra externa, mas esse contorno apresenta alguma ambiguidade. Observa-se que o conjunto dos quatro segmentos marcados em 4. 9b podem ser considerados como um contorno fechado, ou simplesmente ser considerado como um conjunto de quatro segmentos rectos independentes, não constituindo qualquer tipo de contorno fechado, conflito que pode ser resolvido através de considerações de conectividade.

4.2.2 DILATAÇÃO (*"EXPAND"*) E EROSÃO (*"SHRINK"*)

A dilatação é uma operação que expande a fronteira de um objecto ou região binária para o seu *"background"*, tornando-o por isso maior, enquanto a erosão de um objecto binário torna esse objecto mais pequeno, pelo facto de remover a fronteira mais exterior do mesmo.

A dilatação e a erosão podem ser definidos através de operações lógicas sobre imagens binárias, equivalentes em termos de processo de cálculo, às operações de filtragem obtidas por convolução das imagens.

O operador local que efectua a erosão de um pixel com *conectividade oito*, é dado por

$$B = (NOT\ a_1)\ OR\ (NOT\ a_2)\ OR\ (NOT\ a_3)\ OR\ (NOT\ a_4)\ OR\ (NOT\ a_5)\ OR\ (NOT\ a_6)\ OR\ (NOT\ a_7)\ OR\ (NOT\ a_8) \quad (4.7)$$

$$DILATAÇÃO = a_0\ AND\ NOT\ B \quad (4.8)$$

Para o caso da erosão para pixels com *conectividade quatro*, ignoram-se os pixels vizinhos situados nas diagonais.

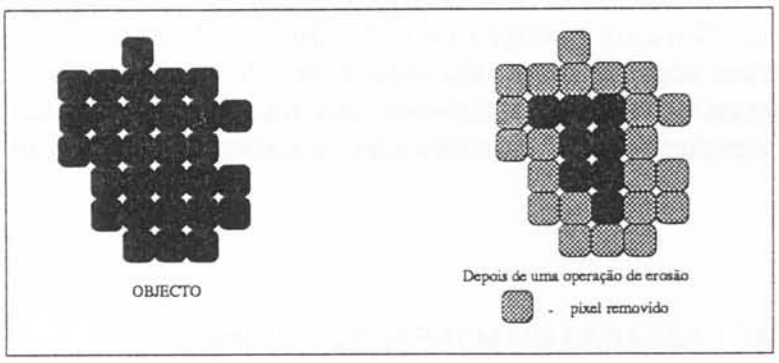


fig. 4.10 Operação de erosão

A operação de dilatação é a operação inversa da anterior, e permite adicionar a um objecto os pixels do *"background"* que sejam vizinhos da fronteira do objecto. O operador booleano para a operação de expansão é dado por:

$$B = (a_1)\ OR\ (a_2)\ OR\ (a_3)\ OR\ (a_4)\ OR\ (a_5)\ OR\ (a_6)\ OR\ (a_7)\ OR\ (a_8) \quad (4.9)$$

$$EXPANSÃO = NOT\ a_0\ AND\ B\ OR\ a_0 \quad (4.10)$$

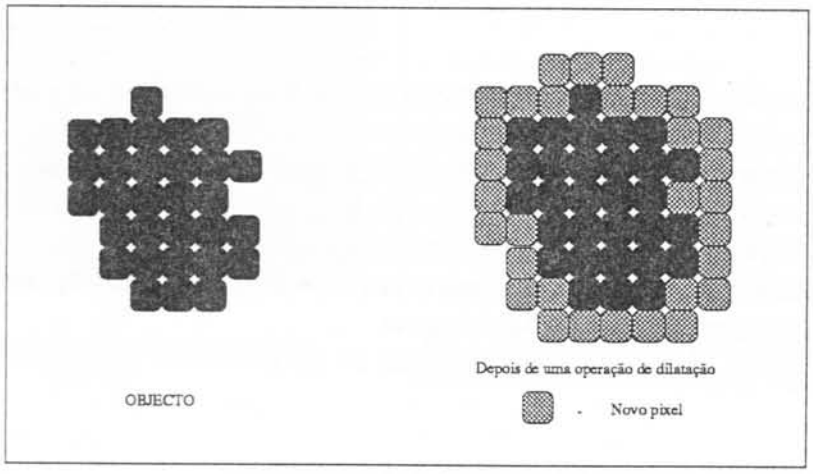


fig. 4.11 Operação de dilatação

Para o caso da dilatação, para pixels com *conectividade quatro*, ignoram-se os pixels vizinhos situados nas diagonais.

A erosão tende a remover pequenos objectos e pequenas projecções dos objectos, enquanto as operações de dilatação permitem encher pequenos buracos e pequenas concavidades de objectos. Efectuar as duas operações de seguida, permite ultrapassar o problema da alteração da dimensão dos objectos obtidos, que estas operações sempre implicam. A erosão seguida de dilatação, permite a remoção de pequenos objectos sem alterar a dimensão dos restantes. A dilatação seguida de erosão, tende a encher buracos e concavidades, sem alterar o tamanho dos objectos.

4.4 TÉCNICAS BASEADAS EM MORFOLOGIA BINÁRIA

4.4-1 INTRODUÇÃO

A morfologia está ligado ao estudo das formas e contornos de objectos e figuras. Os operadores morfológicos são preferencialmente aplicados a imagens binárias, apesar de haver teoria desenvolvida sobre operações morfológicas aplicadas a imagens não binárias. É a primeira abordagem que aqui nos vai interessar.

A determinação de atributos geométricos de imagens sujeitas a processamento prévio, tem-se revelado útil no tratamento das mesmas, já que podem por seu intermédio ser decompostas em componentes com significado. A morfologia

pode simplificar os dados de análise de imagem, preservando o essencial das suas características de forma, e eliminando os aspectos irrelevantes. Estas técnicas encontram-se desenvolvidas em vários livros de processamento de imagem, citaremos apenas as referenciadas na bibliografia, Robert Haralick[19-A] e Schalkoff[29-A].

Na morfologia Matemática, as formas são representados por conjuntos e são também esses conjuntos que nas imagens binárias (ou de dois níveis de cinzento) representam os objectos.

O conjunto de todos os pixels negros numa imagem binária, caracterizam completamente essa imagem. Num espaço bidimensional Euclidiano, os pixels negros denotam os objectos(*"foreground"*-1º plano).

Em morfologia matemática o conjunto dos tuplos formam a imagem binária. O conjunto dos tuplos numa imagem morfologicamente transformada, constituem os pontos seleccionados e o conjunto complementar os pontos não seleccionados. As operações morfológicas primárias são a dilatação e erosão. A partir destas operações elementares, são definidas operações morfológicas secundárias como *fecho* e *abertura*.

4.4-2 OPERAÇÕES MORFOLÓGICAS SOBRE IMAGENS, UTILIZANDO OPERADORES DE MINKOWSKI

4.4.2-1 GENERALIDADES

A análise formal da morfologia matemática, pode ser estudada através da álgebra de Minkowski. As operações conhecidas como adição de Minkowski e subtracção de Minkowski são fundamentais.

As imagens são analisadas em termos de forma e dimensão, através de um elemento, o *'padrão elementar'*, ou também chamado *'elemento estruturante'*. A operação é análoga à da convolução discreta, onde o *'elemento estruturante'* faz as mesmas vezes do *"kernel"* utilizado na convolução. Enquanto que a convolução envolve multiplicações e adições as operações morfológicas envolvem operações efectuadas sobre conjuntos.

As operações entre conjuntos, como *União* ($\cup$), *Intersecção* ($\cap$), e *Complemento*, são fundamentais e podem ser definidas da mesma forma que em 4.1, bem como os conceitos de conjunto e de translação.

Se **A** e **B** representarem conjuntos no Espaço Euclidiano de N dimensões (E^N), para o caso presente o que nos interessa é o espaço R^2 (duas dimensões), com *a* e *b* representando respectivamente os *N* elementos coordenados(tuplos) de **A** e **B**, em que

$$a = (a_1, \ldots, a_N) \quad \text{e} \quad b = (b_1, \ldots, b_N) \tag{4.11}$$

Podemos definir a translação do seguinte modo

$$A_t = A + \underline{x} = \left\{ C_i \in E^N \,\middle|\, c_i = a_i + \underline{x} \,\middle|\, a_i \in A \right\} \tag{4.12}$$

em que $\underline{x} \in E^N$, ou em R^2

$$A_t = A + \underline{x} = \left\{ C_i \in R^2 \,\middle|\, c_i = a_i + \underline{x} \,\middle|\, a_i \in A \right\} \tag{4.13}$$

4.4.2-2 ADIÇÃO DE MINKOWSKI E DILATAÇÃO.

A dilatação morfológica[3], equivalente à adição de Minkowski, J. Serra[30-A]. É uma transformação morfológica que combina dois conjuntos, através do uso da adição vectorial de conjuntos de elementos. Assim a dilatação de A através de B, resulta num conjunto formado por todas as possibilidades da soma vectorial de pares de elementos, sendo uns provindos de A e outros de B.

[3]Para o caso da dilatação a função Fecho(....), numa das suas opções de entrada, em tsegmt3.c implementa a operação mostrada na fig. 4.13. Para o caso da erosão, a função Erosao(....), em tsegmt3.c implementa a operação mostrada na fig. 4.14.

$$A \oplus B = \left\{ C_i \in R^2 \,\middle|\, c_i = a_i + \underline{x} \;\middle|\; \forall\ a_i \in A\ \Lambda\ x \in B \right\}$$

ou seja

$$A \oplus B = \bigcup_{\substack{a_i \in A \\ x \in B}} (a_i + \underline{x})$$

ou

$$A \oplus B = \bigcup_{x \in B} (A + \underline{x})$$

(4.14)

Considerando o conjunto, formado pelos elementos representados **A**={ (0,1) , (1,1) , (2,1) , (2,2) , (3,0) }e o conjunto **B**={ (0,0) , (0,1) } o *elemento estruturante*, obtemos através da operação de dilatação o seguinte conjunto:

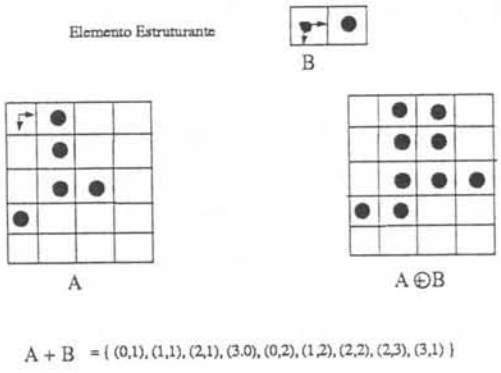


fig. 4.12 Operação de dilatação, representada em termos de conjuntos e a sua correspondente representação binária

fig. 4.13 Imagem binarizada, correspondente à imagem representada na fig. 4.7

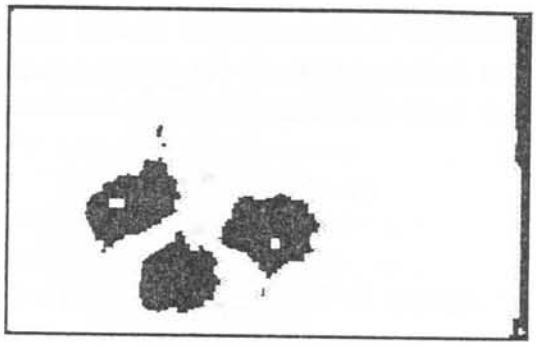


fig. 4.14 Dilatação da imagem representada na fig 4.8

4.4.2-3 SUBTRACÇÃO DE MINKOWSKI E EROSÃO BINÁRIA

A e **B** são conjuntos no espaço Euclidiano R^2, e a operação de erosão pode ser considerada equivalente à operação de subtracção de minkowski, se o elemento estruturante tiver propriedades de simetria, situação que é admitida aqui no trabalho e na descrição da operação de erosão.

$$\mathbf{A} \ominus \mathbf{B} = \left\{ C_i \in R^2 \mid (c_i = a + \underline{x}) \in A \mid \forall x \in B \right\} \qquad (4.15)$$

A erosão de uma imagem **A** através de um elemento estruturante **B**, pode ser descrita pelo conjunto de todos os elementos $C_i \in \mathbf{R^2}$, para os quais B sujeito à translação x, está contido em A. Pode ser vista também como a operação morfológica, que combina dois conjuntos, usando um vector subtracção de elementos do conjunto

$$\mathbf{A} \ominus \mathbf{B} = \left\{ x \in R^2 \mid \forall\, b \in B, \exists\, a \in A \therefore x = a - b \right\} \qquad (4.16)$$

O exemplo da figura 4.15, diz respeito a uma operação de Erosão de A, através do conjunto B (*elemento estruturante*).

A={ (1,0), (1,1), (1,2), (1,3), (1,4), (1,5), (2,1), (3,1), (4,1), (5,1) }e o conjunto B={ (0,0) , (0,1) }

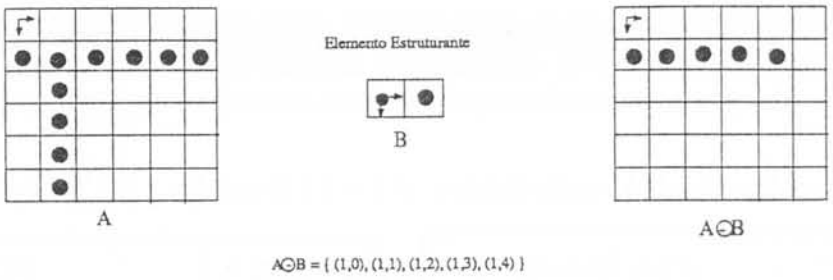


fig. 4.15 Operação Erosão e o correspondente exemplo aplicação binária

4.4.2-4 OPERAÇÕES DE REFLEXÃO E DIFERENÇA DE MINKOVSKI

A reflexão é outra operação binária, que pode ser descrita do seguinte modo

$$-A = \left\{ -a_i \middle| \; a_i \in A \right\} \tag{4.17}$$

O complemento de um conjunto, pode ser definido como

$$A^c = \left\{ x \in R^2 \middle| \; \underline{x} \notin A \; \right\} \tag{4.18}$$

Outra operação importante, definida à custa das operações reunião e intersecção de conjuntos, é a operação diferença

$$\begin{aligned} A / B &\equiv \left\{ x \in R^2 \middle| \; \underline{x} \in \; A \text{ XOR } B \; \right\} \\ &= \left\{ x \in R^2 \middle| (\underline{x} \in (A \cup B) \cap (\underline{x} \notin (A \cap B)) \right\} \end{aligned} \tag{4.19}$$

Para se poder avaliar da equivalência da erosão em relação à subtracção de Minkowski, no caso de elementos estruturantes simétricos, basta atender à definição de erosão e à definição de reflexão

$$A\,(--)\,B = A[-]\,(-B) \text{ e se } -B = B \qquad A\,(--)\,B = (A)[-](B) \tag{4.18}$$

Para o caso da dilatação, a equivalência com a adição de Minkowski verifica-se sempre

$$A\,(++)\,B = A\,[+]\,(+B) \quad \text{equivalente a} \quad A\,(++)\,B = A\,[+]\,(B) \tag{4.20}$$

$$er(A,B) = A(--)B = \{\, \underline{x} \mid (B+\underline{x}) \subset A \,\} \tag{4.21}$$

4.4.2-5 Operações de abertura e fecho

A dilatação e a erosão são usualmente usadas aos pares, quer uma dilatação seguida de uma erosão quer o inverso. Em ambos os casos o resultado é a eliminação de detalhes na imagem, menores do que o elemento estruturante, sem as distorções geométricas que as operações anteriores introduziam. As operações resultantes do emprego de dilatações e erosões, são idenpotentes, ou seja não acrescentam mais alterações na imagem do que a que resulta da primeira aplicação, se forem repetidas sucessivamente.

Em termos funcionais as operações de abertura e fecho, correspondem à especificação de filtros passa banda, e estabelecem a operação equivalente à projecção ortogonal, para a álgebra linear.

A operação de abertura é definida do seguinte modo

$$A \circ B = \text{open}(A, B) \equiv \text{dil}(\text{er}(A, B), B) = (A \ominus B) \oplus B \tag{4.22}$$

A operação de fecho é definida por

$$A \bullet B = \text{close}(A, B) \equiv \text{er}(\text{dil}(A, B), -B) \tag{4.23}$$

se o elemento estruturante for simétrico, ou seja invariante com rotações de 180º, a operação de fecho reduz-se a:

$$= (A \oplus B) \ominus B \tag{4.24}$$

Se **A** permanecer invariante com a aplicação de uma operação de fecho, através de **B**, dizemos que **A** é fechado em relação a **B**, se **A** permanecer invariante

com a aplicação de uma operação de abertura, através de **B**, dizemos que **A** é aberto por **B**. As fig4.15 e 4.16 representam precisamente uma operação de erosão e de abertura respectivamente, efectuada sobre a imagem binarizada da fig4.8.

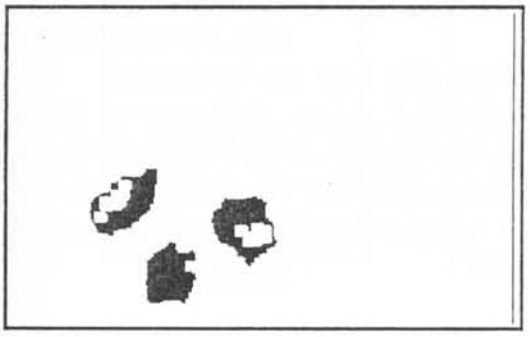


fig. 4.16 Imagem correspondente à prancha da fig. 4.8, depois de erosão

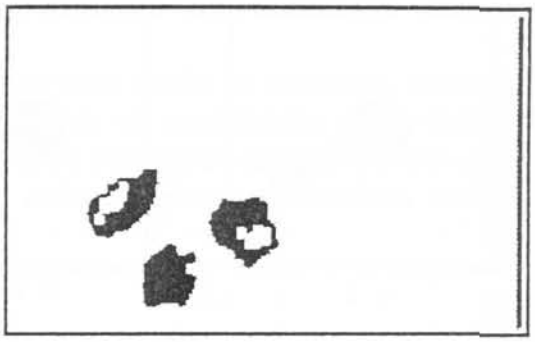


fig. 4.17 Imagem correspondente à prancha da fig. 4.8, depois de abertura

As diferenças entre as duas imagens é pouco visível, já que a operação de abertura tem em relação à de erosão apenas a função de repor as dimensões originais do objecto. Se o *elemento estruturante* for de dimensões reduzidas como é o caso, os objectos não serão muito diferentes depois das duas operações.

A capacidade de uma operação de abertura, para poder separar os elementos constituintes de um objecto, resulta da constituição do elemento estruturante e da definição da operação de abertura, descrita do modo seguinte:

$$A \circ B = \left\{ x \in A \middle| \text{ para } \forall\, y \in A \ominus B,\ x \in B_t \ \Lambda\ B_t \subseteq A \right\} \qquad (4.25)$$

A operação de abertura de **A** por **B** selecciona precisamente aquelas partes de **A** que encaixam em **B**, no sentido em que possam ser encontradas por translação do elemento estruturante **B**, através de **A**. A operação de abertura, por intermédio de um *disco estruturante*, permite suavizar um contorno, elimina pequenas ilhas e pequenos istmos nas figuras. O fecho de uma imagem com um disco estruturante também suaviza o contorno, elimina pequenos buracos além de encher pequenas reentrâncias.

As primitivas de uma figura podem ser extraias através de operações morfológicas, em que a ordem das operações assume aspecto relevante.

O exemplo da fig 4.18 mostra uma operação de selecção de primitivas, através da utilização de operações de abertura e diferença. O elemento **L** utilizado

como *elemento estruturante*, é um disco circular. Abrindo F com **L** (F o L)obtenho o corpo do objecto, fazendo F - (F o L) obtenho o restante.

fig. 4.18 Extracção de primitivas da imagem, através de operações de Abertura e Diferença

A operação de fecho pode ser usada para determinar, agrupamentos de características em *"clusters"*, como se ilustra na fig. 4.19. Para a referida figura, supondo que o afastamento médio entre características pertencentes à mesma classe, é sempre inferior a $\rho_0/2$, e entre pontos pertencentes a *"clusters"* diferentes é sempre superior a $2\rho_0$, um elemento estruturante quadrado de aresta ρ_0, teria capacidade para agrupar os pontos pelas classes respectivas, através da utilização de operações de fecho.

fig. 4.19 Imagem binária depois de sujeita a operação de fecho, através de um *elemento estruturante* quadrado de aresta ρ_0

4.5.DESCRIÇÃO DE FORMAS

A descrição das formas dos objectos, exige o conhecimento de técnicas de extracção das propridades geométricas dos objectos, Baxes[5-A], Besançon[7-A], etc. e a possibilidade da codificação ou representação dessa informação, através de um descritor qualquer.

Saliente-se que essa informação tanto pode residir no contorno, como pode residir na superfície. Assim, linhas curvas podem ser consideradas como elementos primitivos de estruturas mais globais de objectos complexos. Essas estruturas globais (objectos) podem ser descritas pelos seus atributos de forma: métricos, analíticos, etc, extraíveis por análise do objecto

4.5-1 ATRIBUTOS MÉTRICOS

Estes atributos estão baseados em medidas de distância entre pontos $d[(x_i,y_i) , (x_j,y_j)]$ em que (x_i,y_i) e (x_j,y_j) são pontos que satisfazem as seguintes propriedades:

$$1 - d[(x_i,y_i) , (x_j,y_j)] \geq 0 \tag{4.26}$$

$$2 - d[(x_i,y_i) , (x_j,y_j)] = d[(x_j,y_j) , (x_i,y_i)] \tag{4.27}$$

$$3 - d[(x_i,y_i) , (x_j,y_j)] + d[(x_j,y_j) , (x_k,y_k)] \geq d[(x_i,y_i) , (x_k,y_k)] \tag{4.28}$$

Algumas das medidas que satisfazem estas propriedades, são:

Distância Euclidiana

$$d_E = [(x_i - x_j)^2 + (y_i - y_j)^2]^{1/2} \tag{4.29}$$

Distância em amplitude

$$d_M = | x_i - x_j | + | y_i - y_j | \tag{4.30}$$

Máximo valor da distância

$$d_X = \max \{ | x_i - x_j | , | y_i - y_j | \} \tag{4.31}$$

Com estas medidas de distância diversas características métricas podem ser atribuidas aos objectos/regiões na imagem.

4.5-2 DESCRITORES DE CONTORNO

4.5.2-1 CADEIA NUMÉRICA E NÚMEROS DE FORMA

As cadeias numéricas descrevem um contorno a partir dum código de direcções, que por sua vez depende do ponto de arranque dessa cadeia. Com o objectivo de normalizar estes códigos, define-se o código diferença para a cadeia, em que o que é medido é a diferença entre os códigos na cadeia, efectuada no sentido trigonométrico. A *primeira diferença* está ligada à menor diferença existente na cadeia numérica, e é definida, contando no sentido trigonométrico o número de direcções que separam dois elementos adjacentes na cadeia. Define-se ainda o número de forma dum contorno, obtido a partir da cadeia numérica e estabelecido pela primeira diferença mais pequena em amplitude encontrada entre dois pontos do contorno. O número de ordem **n** da cadeia é o número de segmentos contido na cadeia. Uma cadeia fechada é caracterizada por um **n** par (para a conectividade quatro).

Também para normalização dos códigos de cadeia, podem ser definidos critérios para o estabelecimento da grelha de conectividade do código. Assim estabelecem-se as seguintes definições:

Eixo principal do contorno - linha direita ligando os dois pontos mais afastados do contorno

Eixo secundário - linha perpendicular à primeira e limitada pelo contorno

Excentricidade - relação entre eixos

Rectângulo circunscrito - definido pelas linhas perpendiculares aos eixos.

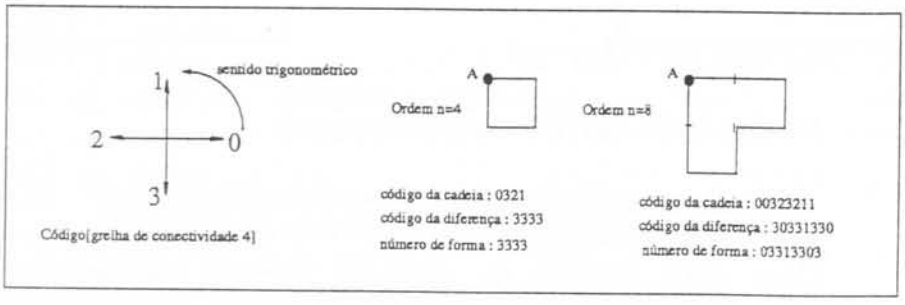


fig. 4.20 Códigos de Cadeia e Números de Forma

4.5.2-2 CÓDIGO DE CADEIA - *CHAIN CODE.*

Muitos métodos binários, utilizam a redução da dimensão dos códigos da fronteira de objectos numa imagem, através da utilização de *"Chain codes"*, já que a utilização de uma dimensão representa economias de espaço e tempo consideráveis no processamento das imagens. Se cada pixel vizinho de a_0, for etiquetado com um nome tal como está representado na fig. 4.20, a fronteira de um contorno qualquer, pode ser representada por uma sequência numérica, de acordo com o quadro da fig. 4.20-a, começando num ponto arbitrário qualquer do contorno, que deverá ser referenciado. Este código é equivalente ao anterior, apenas apresentando um nome e um número de direcções maior e previamente estabelecidas.

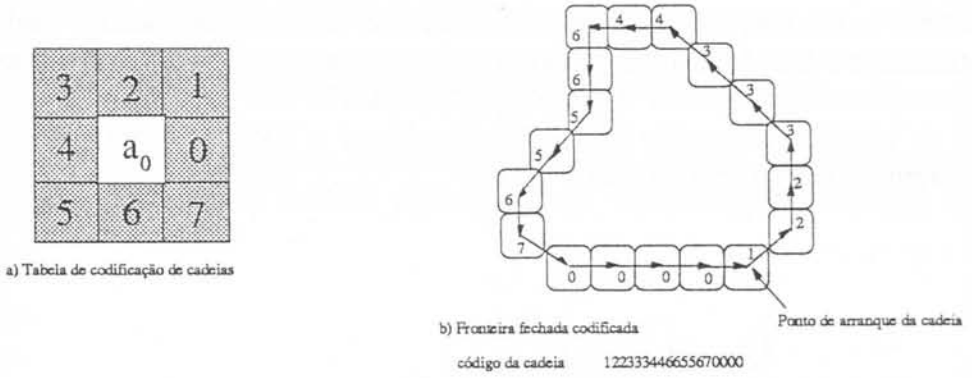


fig. 4.21 Codificação de Cadeias - *"Chain Code"*.

O código da cadeia pode ser usado para reconstituir a fronteira de um objecto, a partir do código do qual foi derivado, desde que se conheça exactamente o ponto de arranque da mesma.

4.5.3 DESCRITORES DE SUPERFÍCIE

4.5.3-1 ATRIBUTOS/DESCRITORES TOPOLÓGICOS (NÚMEROS DE EULER)

Topologia é o estudo da configuração geométrica através de um conjunto de propriedades específicas, e invariantes com operações RST, Pratt[26-A]. A

excentricidade ou alongamento (dada pela relação *l/l'* = eixo maior/eixo menor, num objecto) pode ser uma dessas propriedades. Por exemplo o quociente (P_r^2/A_r), é invariante em relação a transformações RST e permite descrever o alongamento de um objecto, sendo mínimo para objectos circulares (igual a: 4π no círculo, 16 no quadrado e 18 no rectângulo [a x 2a]).

A conectividade é outro exemplo de atributo topológico, que pode ser utilizado na caracterização de objectos binários, entendidos como objectos conexos, podendo ter ou não buracos internos.

Existe uma relação fundamental numa imagem binarizada, entre o número de componentes conexos (ligados) *C* numa imagem e o número de buracos (*"Holes"*) *H*, chamado número de Euler (*E*) e calculado por $E = C - H$

Enquanto para uma região conexa ou ligada, todos os pixels dessa região são vizinhos conexos (*quatro* ou *oito*), um buraco é uma região completamente envolvida por uma região conexa. Dessa forma o número de Euler, é uma propriedade topológica, já que o são também *C* e *H*.

Uma forma simples de determinar o número de Euler, bem como medidas como perímetro e área de imagens binarizadas (em que atribuo '1' é dado ao objecto e '0' ao *background*), é através da utilização do cálculo dos *"bit quad"*, que são padrões de pixels de dimensão 2x2. Posso considerar o seguinte conjunto de *"bit quads"-arranjo quaternário de bits*:

$$Q_0 = \left\{ \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 0 & 0 \\ \hline \end{array} \right\}$$

$$Q_1 = \left\{ \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 0 \\ \hline \end{array}, \begin{array}{|c|c|}\hline 0 & 1 \\ \hline 0 & 0 \\ \hline \end{array}, \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 0 & 1 \\ \hline \end{array}, \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 1 & 0 \\ \hline \end{array} \right\}$$

$$Q_2 = \left\{ \begin{array}{|c|c|}\hline 1 & 1 \\ \hline 0 & 0 \\ \hline \end{array}, \begin{array}{|c|c|}\hline 0 & 1 \\ \hline 0 & 1 \\ \hline \end{array}, \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 1 & 1 \\ \hline \end{array}, \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 1 & 0 \\ \hline \end{array} \right\}$$

$$Q_3 = \left\{ \begin{array}{|c|c|}\hline 1 & 1 \\ \hline 0 & 1 \\ \hline \end{array}, \begin{array}{|c|c|}\hline 0 & 1 \\ \hline 1 & 1 \\ \hline \end{array}, \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 1 & 1 \\ \hline \end{array}, \begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & 0 \\ \hline \end{array} \right\}$$

$$Q_4 = \left\{ \begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & 1 \\ \hline \end{array} \right\}$$

$$Q_D = \left\{ \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 1 \\ \hline \end{array}, \begin{array}{|c|c|}\hline 0 & 1 \\ \hline 1 & 0 \\ \hline \end{array} \right\}$$

fig. 4.22 Diferentes tipos de *células quaternárias* (*"bit quad"*)

Para imagens segmentadas, em que o fundo corresponde a '1' e o *background* (envolvente) corresponde a '0', a determinação do perimetro e da área à custa dos *"bit-quad"* pode ser encontrada do seguinte modo:

$$P = n(Q_2) + 1/\sqrt{2}\,[\,n(Q_1) + n(Q_3) + 2.\,n(Q_D)\,] \qquad (4.32)$$

$$A = (1/4).\,n(Q_1) + (1/2).\,n(Q_2) + (7/8).\,n(Q_3) + n(Q_4) + (3/4).\,n(Q_D) \qquad (4.33)$$

Os bit-quad também proporcionam uma maneira prática de determinação dos números de Euler

$$E = 1/4\,[\,n(Q_1) - n(Q_3) + 2.\,n(Q_D)\,] \qquad (4.34)$$

para a definição de objectos de *conectividade quatro.*

$$E = 1/4\,[\,n(Q_1) - n(Q_3) - 2.\,n(Q_D)\,] \qquad (4.35)$$

para a definição de objectos de *conectividade oito.*

Se houver mais zonas conexas do que buracos, os números de Euler dão-nos o número aproximado de objectos que compõem a imagem.

4.5.3-2 DESCRITORES DE GEOMETRIA ELEMENTAR

Se tiver uma imagem binária representando objectos, as medições mais simples que se podem fazer na imagem, são a área e o perímetro. A área pode ser obtida simplesmente por contagem dos pixel na região interna definida como objecto (note-se que a área é sensível às mudanças de escala). O perímetro pode ser medido contando os pixels que circundam, ou formam a fronteira do objecto previamente segmentado. Isto ignora contudo que a distância entre pixels na horizontal ou vertical, é igual a 1, enquanto que entre pixel na diagonal é maior em 15%.

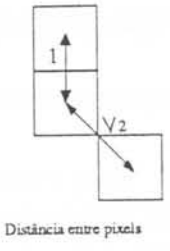


fig. 4.23 Distância entre pixels numa imagem

Uma aproximação com melhores resultados no cálculo do perímetro, do que a indicada primeiro, consiste na contagem dos pixels da fronteira para a *conectividade quatro*(P4) e depois a contagem para a *conectividade oito*(P8). O produto P4. P8 $\equiv P^2$.

A classificação de objectos numa imagem, é muitas vezes uma mera questão de capacidade para discriminar formas, pelo que o tipo de características que se procuram estão relacionados com índices de forma. Um índice geral de forma, fácil e rápida de calcular é simplesmente o quociente P^2/A, igual ao quadrado do perímetro pela área, índice que é independente de mudanças de escala (S), rotação (R), e translação (T). Apesar de não ter um poder discriminante muito grande, permite distinguir entre objectos com formas geométricas regulares, como quadrados, triângulos, elipses.

Outros indicadores reveladores da forma, referem-se por exemplo aos rectângulos englobados pelo objecto: o rectângulo envolvente, o rectângulo mínimo envolvente e o polígono convexo envolvente, representados na fig. 4.24.

a) Rectângulo envolvente b) Rectângulo mínimo envolvente c) Poligono convexo envolvente

fig. 4.24 Descritores geométricos de um objecto

4.5.3-3 MOMENTOS

Para formas gerais, existem soluções baseadas em momentos. Os momentos têm como objectivo fundamental, estudar as propriedades mecânicas dos objectos, e são descritores geométricos por excelência, Schalkoff[29-A]. O objectivo é encontrar propriedades que sejam invariantes com as mudanças de escala (S), rotação (R), e translação (T).

A transformada de imagem, representada pela função $f(x,y)$, num conjunto de momentos é dada por

$$m_{pq} = \int\int_{-\infty}^{+\infty} x^p y^q f(x,y)dxdy \qquad p,q = 0,1,2, \ldots, \infty \qquad (4.36)$$

Para o caso de uma imagem discretizada de dimensões M x N, representada em cada ponto por $f(i,j)$, um momento de ordem (p,q) para um determinado objecto na imagem, é determinado do seguinte modo

$$m_{pq} = \sum_{i=0}^{M} \sum_{j=0}^{N} i^p j^q f(i,j) \qquad (4.37)$$

Os pontos (i,j) são todos pixels interiores ou que fazem parte do contorno do objecto. Numa imagem binarizada(sujeita a segmentação), $f(i,j)$ só toma valores iguais a '0' e '1', e para o caso de um objecto R segmentado $f(i,j)$ é sempre 1, para qualquer dos seus pontos, logo:

$$m_{pq} = \sum_{i=0}^{M} \sum_{j=0}^{N} i^p j^q \qquad (4.38)$$

Os momentos não são contudo invariantes com as mudanças de escala (S), rotação (R), ou translação (T), pelo que se utilizam momentos centrais. Existem momentos com especificidades particulares, como o m_{00} que é a área e m_{20} e m_{02} são momentos de inércia. Os centroides m_{10} e m_{01} são invariantes com a translação, mas o mesmo não acontece para momentos de ordem superior.

Os centroides de um objecto são dados por:

$$\bar{i} = \frac{m_{10}}{m_{00}} \qquad \text{e} \qquad \bar{j} = \frac{m_{01}}{m_{00}}$$

e os momentos centrados dados por (4.39)

$$\mu_{pq} = \sum_{i,j} \sum_{\in R} (i - \bar{i})^p (j - \bar{j})^q$$

Para produzir índices de forma a partir de momentos centrados, a única coisa a fazer é a utilização de rácios de momentos, que tenham o mesmo valor de $i+j$. Os momentos centrados μ_{10} e μ_{01} são iguais a zero.

Os momentos centrados ainda são sensíveis às mudanças de escala *(S)* e rotação *(R)*. A invariância em relação às mudanças de escala pode contudo ser obtida através da expressão (4.40), obtendo assim um conjunto de momentos normalizados, invariantes com RST.

$$\eta'_{pq} = \frac{\mu_{pq}}{\mu_{00}^{[(p+q)/2+1]}} \qquad p+q = 2,3 \ldots \tag{4.40}$$

Podem-se usar as seguintes interpretações no que diz respeito aos momentos centrados de ordem mais baixa (dependentes contudo das mudanças de escala *(S)* e rotação *(R)*)

MOMENTO CENTRAL	INTERPRETAÇÃO
μ_{20}	Centroide horizontal
μ_{02}	Centroide vertical
μ_{11}	Diagonalidade (indicação do quadrante onde R tem mais massa)
μ_{12}	Divergência horizontal. Indica a extensão relativa da parte esquerda de R face à direita
μ_{21}	Divergência vertical. Compara a extensão relativa da base face ao topo
μ_{30}	Balanceamento horizontal. Localização do centro de gravidade com respeito à horizontal média
μ_{03}	Balanceamento vertical. Localização do centro de gravidade com respeito à vertical média

4.5.3-4 REPRESENTAÇÃO DUMA SUPERFÍCIE POR EIXO MÉDIO E ESQUELETIZAÇÃO

O eixo médio de uma região R pode ser obtido através da função

$$q(x,B) = min(\ d(x,y)\ )|_{y \in B} \tag{4.41}$$

Dada a região R com uma fronteira B, para qualquer ponto x interior ao objecto, podemos definir a função (4.41), em que d define uma distância Euclidiana entre x e y. Cada ponto interior a B, que possua mais do que um $q(x,B)$ iguais, pertence ao eixo médio da região. O conjunto de todos os centros das circunferências incluídas no objecto (ou região) formam o eixo médio. O máximo desta função $q(x,B)$

também nos dá o valor da maior circunferência inscrita no objecto. A figura pode assim ser esqueletizada, através da sua representação pelo eixo médio determinado. O eixo maior da esqueletização dá-nos o comprimento e a orientação do objecto.

a) Eixo médio

b) Circunferências inscritas no rectângulo centros determinam o esqueleto

fig. 4.25. Extracção do eixo médio

O uso da fronteira para descrever a forma dos objectos é útil e prática. Posso definir esqueleto de um objecto como sendo o conjunto dos pontos que se situam no meio do objecto, a igual distância das fronteiras. Isso corresponde à determinação do eixo médio do objecto MAT(Medial Axis Transform)[4-A]. Para definir o MAT, primeiro há que compreender o que se entende por distância ou profundidade de um pixel, de um objecto em relação ao *background*. Posso definir essa profundidade por distância Euclidiana, mas a forma mais simples de o fazer, é medir a distância mais curta do pixel ao *"background"*, que significa medir a mais curta *conexão quatro* (ou *conexão oito*), até ao encontro do *"background"*.

a) Profundidade por conectividade 4

b) Profundidade por conectividade 8

fig. 4.26 Determinação da profundidade de um pixel

Se substituir o valor de cada pixel pela sua distância ao background, fico com um mapa de profundidades do objecto, em relação ao *"background"*.

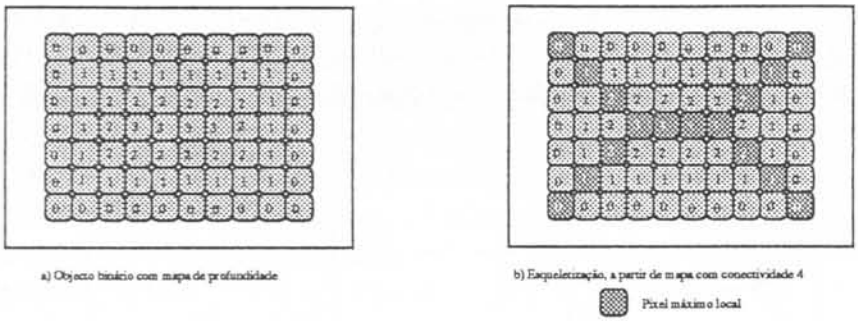


fig. 4.27 Mapa de profundidades e esqueletização

A determinação do MAT é agora estabelecido com base neste mapa de profundidades e consiste na selecção dos pixels com valor máximo local, em cada ponto da imagem , utilizando o mesmo critério de selecção do que foi utilizado para achar a profundidade. O MAT é um conjunto de pontos mínimos, com referência às distâncias, a partir dos quais é possível reconstruir o objecto, através da expansão dos seus pontos.

O MAT é uma esqueletização simples de procurar, que apresenta contudo algumas limitações, já que é bastante sensível a irregularidades do objecto e pode não se constituir como um conjunto conexo de pontos.

4.6 EXTRACÇÃO DA FRONTEIRA

É possível formular ou definir, o problema da extracção dos contornos ou orlas de um objecto de várias formas diferentes. Referiremos aqui três dessas possibilidades.

1 - ***Ajuste à curvatura dos pontos do contorno***: Este processo baseia-se no ajustamento ao contorno através de uma curva ou recta, que passe por todos esses pontos conhecidos do contorno. É contudo difícil retirar contornos por este processo, principalmente quando existam muitos objectos na imagem, ou quando existam objectos sobrepostos ou *edges* muito dispersos.

2 - ***Extracção do contorno, seguindo a fronteira***: Este processo procura extrair o contorno através da correcção do erro cometido quando se procura os limites da fronteira em zig-zag, seguindo o percurso da forma como mostra a fig. 4.28. Quando o contorno é cortado é seguido dum

erro. Se o percurso encontrou o *background* a correcção é feita à direita, se o percurso encontra o objecto o percurso é continuado procurando à esquerda. O ponto de arranque é efectuado sobre a fronteira do objecto, ponto que é depois da aplicação do algoritmo também o ponto de paragem. A fronteira será constituída por todos os pixels percorridos durante a pesquisa do contorno.

As desvantagens que este algoritmo pode apresentar, derivam do facto dos ruídos que eventualmente a imagem possa conter, poderem perturbar a extracção do contorno respectivo. Por exemplo, através do surgimento de contornos duplos e ainda pelo facto do ponto escolhido para o arranque influenciar a forma da fronteira extraída e desse modo poderem ser perdidas regiões espúrias, marginais, existentes nos objectos segmentados.

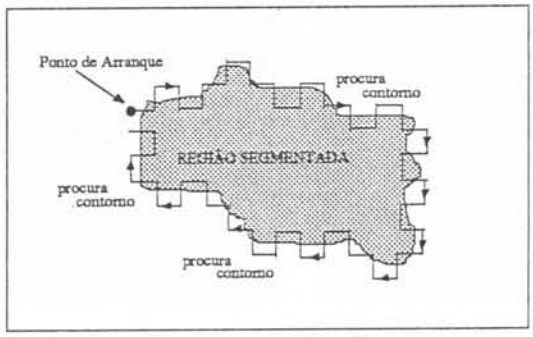
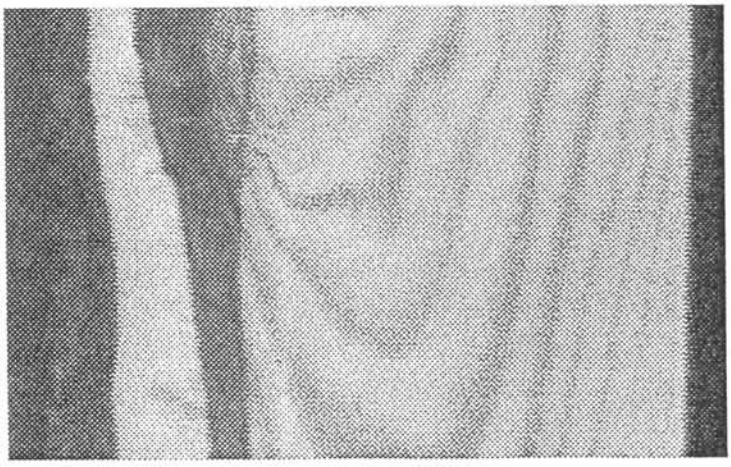


fig. 4.28 Procura do contorno seguindo a fronteira



fig, 4.29 Madeira com afloramentos de casca

3 - ***método da profundidade***: Se num objecto duma imagem segmentada for conhecido o mapa de profundidades dos seus pixels em relação ao *"background"*, os pixels de menor profundidade (ou profundidade zero), constituirão a fronteira.

As imagens representadas nas fig. 4.30 e fig. 4.31 representam os contornos dos objectos, identificados como defeitos na imagem da fig. 4.29, obtidos através de processos diferentes[4]. A primeira imagem utiliza o processo dois, descrito anteriormente, e a segunda um processo que mede a profundidade de cada pixel nas regiões segmentadas na imagem, processo descrito no ponto três.

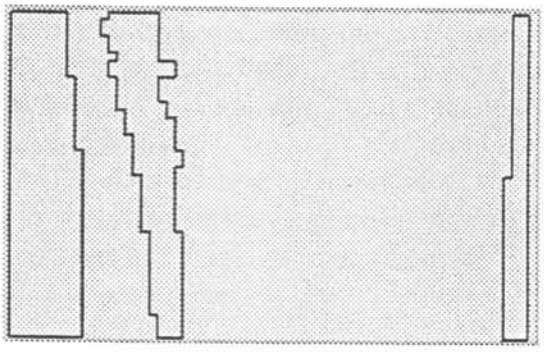


fig. 4.30 Fronteira retirada por pesquisa/correcção do contorno

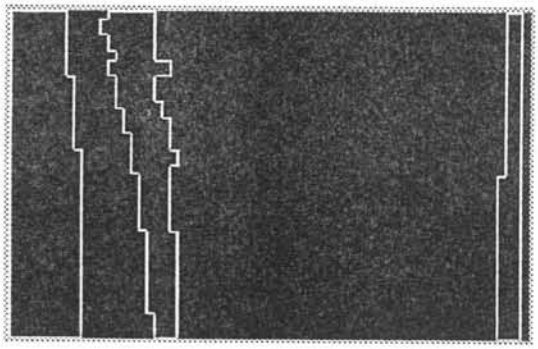


fig. 4.31 Fronteira retirada por profundidade

4.7 Conclusões

O sistema de *localização e reconhecimento de defeitos em madeira*, necessita antes de mais, de saber manipular e extrair contornos das regiões segmentadas de uma imagem de madeira. Conhecer os contornos das regiões, representa poder tratar cada uma dessas regiões de uma forma independente, e poder acelerar a velocidade de análise e classificação desses segmentos de imagem.

[4]A imagem da fig. 4.29 foi obtida por pesquisa das orlas na imagem, com o algoritmo implementado pela função tpct1(...) em TGCTN.C, enquanto a imagem da fig. 4.30, foi obtida através da função esqeltzcf2(...) em TSEGMT1.C.

Por outro lado, o conhecimento de certas propriedades do contorno, permitem ao algoritmo de reconhecimento, utilizar essa informação sob a forma de características geométricas de imagem.

Todo este capítulo, trata do assunto relacionado com a extracção de propriedades/padrões, de natureza topológica ou geométrica dos objectos contidos nas imagens, através de atributos de superfície ou do contorno. Foi dentro dessa perspectiva, que se fez um estudo breve, sobre a caracterização de objectos através de momentos.

A utilização de operações lógicas e operações de Minkowski, tiveram por objectivo, abordar o estudo da *dilatação*, *erosão*, *fecho* e *abertura*, operações que serão implementadas sobre imagens binárias, com objectivos de simplificação das mesmas, ou com o objectivo da sua caracterização geométrica.

CAPÍTULO V

TEXTURAS

5 - TEXTURAS

5-1 INTRODUÇÃO

Em muitas imagens, existem detalhes significativos que se repetem através de áreas extensas. Essas áreas da imagem podem ser caracterizadas através de elementos estruturais repetitivos que se encontram na imagem. Esses elementos estruturais podem ter natureza geométrica (macro estruturas) ou pelo contrário não conterem elementos representáveis pela sua geometria ou forma. Neste caso a imagem será caracterizada pela sua micro estrutura, que é aquilo que acontece vulgarmente, com imagens de substâncias naturais.

Enquanto que as texturas artificiais, são constituídas por arranjos simbólicos de elementos, sobre fundo neutro (símbolos que podem representar segmentos, linhas, pontos, caracteres, figuras geométricas, etc.), as texturas naturais, aquelas que nos interessa tratar, são imagens de cenas ou objectos naturais que contêm arranjos de pixels com estruturas repetitivas[Pratt-26], geralmente mal definidas por observação visual. Podem-se distinguir nas texturas naturais, aquelas em que o período de repetição de pixels ou grupo de pixels é largo, e aquelas em que o período de repetição é fino.

Sendo difícil a definição de textura, pode-se dizer que é uma estrutura composta por grande número de elementos (ou formas) semelhantes, mais ou menos ordenadas, sem que nenhum dos elementos se evidencie sobre os outros. Considera-se que nas imagens existem informações do tipo mono dimensional (contornos, arestas (*edges*), etc.) e do tipo bidimensional (superfície, regiões). São as informações bidimensionais, relativas ao estado da superfície, a que se chama textura.

Podemos também definir textura como tudo aquilo que não é contorno, ou como uma propriedade dos objectos que se manifesta pela repetição em: posição, orientação e em formações variadas de um elemento primitivo - o *Texel*. Segundo Hawkins [1970][1], textura dependeria de três ingredientes: Uma *'ordem local'* (elemento primitivo) que se repete por uma região maior, comparada com o tamanho da *'Ordem local'*. A *'ordem espacial'* correspondente a um arranjo não aleatório das *ordens* elementares; e as *'partes'*, entidades regularmente uniformes portadoras das mesmas dimensões em qualquer lugar da região onde se situe a textura.

Uma vez que a textura é uma medida ligada a propriedades de natureza espacial, as medições devem ser restringidas a regiões com propriedades uniformes. Por esse motivo pode ser necessário antes de aplicar processos de observação estatística de texturas, ter de proceder primeiro a técnicas de segmentação de imagem e de definição de regiões.

A fig. 5.1 representa as etapas principais num sistema de análise de texturas, como processo de extracção de padrões de uma imagem, com o objectivo da sua eventual classificação.

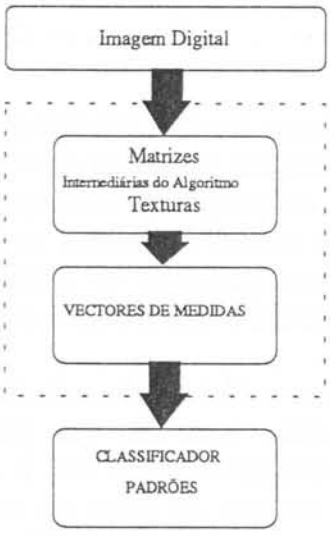


fig. 5.1 Etapas dum sistema de análise de texturas

Podemos definir dois tipos de texturas: *texturas determinísticas*, que são compostas por formas (ditas primitivas) que se repetem segundo regras precisas, e que são estudadas por análise estrutural ou sintáctica, e as *texturas aleatórias* que são formas visuais complexas onde não existem motivos isolados ou uma frequência

[1]Citado por Pratt em [26-A], na bibliografia.

de repetição dominante, e que são estudadas por análise estatística. A análise estatística de textura, calcula as características locais em cada ponto da imagem, e deriva de lá um conjunto de medidas, a partir da distribuição dessas características. As características locais são definidas, pela combinação das intensidades em dois pontos vizinhos da imagem.

De acordo com o número de pontos (ou pixels) para os quais são definidas as características locais, assim se classifica a estatística como de: 1ª ordem (baseada no valor dos pixels tomados individualmente), de 2ª ordem (baseada na relação entre dois pixels, em posições distintas da imagem) ou estatísticas de ordem superior, quando definidas para mais que dois pontos. As medidas estatísticas não precisam de ser apenas baseadas na intensidade dos pixels, mas podem ser baseadas em arranjos geométricos, arestas (*edges*), vales e picos, aglomerados de pontos, etc.

A textura é uma propriedade espacial. Assim ela pode estar ligada à divisão da imagem em regiões, partilhadas por uma propriedade comum. A textura pode ser usada para classificar regiões, para fazer a segmentação de imagens em regiões e ainda como auxiliar, na extracção de informação tridimensional.

Para discriminar diferentes tipos de textura, abordagens estatísticas, estruturais ou espectrais podem ser utilizadas. Na análise de texturas, os problemas vão desde a determinação dos elementos primitivos à descrição das interdependências e interacções, dessas primitivas. Para resolver esses problemas, a utilização de funções de auto-correlação; transformadas discretas; matrizes de co-ocorrência de níveis de cinzento; modelos auto-regressivos, etc., podem ser utilizados.

5 2 TEXTURA: PROCESSO ESTRUTURAL E SINTÁCTICO

Os métodos estruturais procuram descrever as primitivas que constituem a textura e as regras com as quais ela se organiza. As primitivas são conjuntos conexos, caracterizados por atributos, e as regras constituem relações espaciais entre as primitivas.

Os processos sintácticos utilizam gramáticas para descreverem a estrutura das texturas. Três tipos de gramáticas são utilizadas: gramáticas de forma, gramáticas em árvore (hierárquicas) e gramáticas matriciais.

A análise de texturas em imagens binárias através de morfologia matemática também foi proposto por Matheron, Serra e Verchery[30-A], e baseia-se fundamentalmente na escolha de um *elemento estruturante*, constituído por um conjunto de pontos segundo uma forma particular. Uma nova imagem binária é gerada, através de operações de erosão ou dilatação do elemento estruturante com a imagem binária inicial. A textura pode depois ser caracterizada pelo número de objectos existentes na imagem, depois de submetida às operações morfológicas adequadas.

A textura pode também ser identificada pelos elementos constituintes da sua estrutura. As texturas são assim separadas e discriminadas, através dos seus elementos estruturais. Laws[1980][15-B] propôs um conjunto de máscaras 3x3, para serem convulcionadas com uma imagem, assinalada pela presença de texturas diferentes na sua superfície. O resultado da operação está ligada ao acentuar das microtexturas presentes, ultimando a identificação da sua microestrutura. É de salientar que uma das características estruturais, dos elementos de uma textura, são a sua dimensão. Se uma máscara de convolução 3x3 for usada, é só possível falar do reconhecimento de microestruturas.

5.3 TEXTURA: PROCESSO ESTATÍSTICO

5.3-1 ESTATÍSTICA DO 1ºGRAU

Consideremos $\varnothing\ (i)$, com $i=0, \ldots, l-1$, o número de pontos numa imagem com nível de cinzento i e n o número total de pontos na imagem. Define-se $P(i)$, como a densidade de probabilidades dos níveis de cinzento na imagem I, estatísticas de 1ª ordem, com $P(i)$ definido por

$$P(i) \equiv P_R\{ I(l,k) = i \} \qquad \text{em que } 0 \leq i \leq l-1 \qquad (5.1)$$

O histograma da imagem pode ser estimado através de $P(i)$, como sendo igual a:

$$P(i) = \varnothing\ (i)/n \qquad (5.2)$$

A forma do histograma permite estimar muitas grandezas que caracterizem a imagem. Por exemplo um histograma muito estreito indica um baixo contraste, enquanto que um histograma bimodal indica a existência de duas regiões com brilhos diferentes. As medidas seguintes são utilizadas para caracterizar o histograma:

Média

$$\mu = \sum_{i=0}^{L-1} iP(i) \qquad (5.3)$$

Variância

$$\sigma_i^2 = \sum_{i=0}^{L-1} (i-\mu)^2 P(i) \qquad (5.4)$$

Coeficiente de assimetria (Skewness)

$$\alpha_3 = \frac{1}{\sigma^3} \sum_{i=0}^{L-1} (i-\mu)^3 P(i)$$

o terceiro momento é igual a (5.5)

$$\sum_{i=0}^{L-1} (i-\mu)^3 P(i)$$

Kurtosis

$$\alpha_4 = \frac{1}{\sigma^4} \sum_{i=0}^{L-1} (i-\mu)^4 P(i) - 3 \qquad (5.6)$$

Energia

$$E = \sum_{i=0}^{L-1} [P(i)]^2 \qquad (5.7)$$

O terceiro momento mede o "*skew*" (desvio da simetria) do histograma. Quando o histograma for simétrico o *skew*=0, quando tem um desvio à esquerda ou direita, o *skew* é negativo ou positivo respectivamente. A *energia* mede, por seu lado, a uniformidade do histograma. Quando o histograma é uniforme a *energia* toma o valor máximo, enquanto a existência de *clusters* na imagem leva a que o valor de E se aproxime de zero. Por outro lado a entropia mede a desordem duma imagem, e apresenta valores inversos do da energia.

As imagens das figuras 6.2 e 6.3, referenciadas por um e dois, visam tipificar valores estatísticos de medidas de textura propostas entre outros por, Tomita Fumiaki e Saburo[31-A], Besançon[7-A], etc.,. A tabela 6-1 referência precisamente essas medidas do primeiro grau, respectivamente: *média*, *variância*, *energia*, *entropia*, *"skewness"* e *"kurtosis"* das imagens[2].

[2]As medidas estatisticas do 1ºgrau, são retiradas da imagem através das funções thistogm(...), SKewness(...) , Kurtosis(...), em TFUNC.C.

fig. 5-2 Imagem 1 - madeira de pinho com nó e cardida

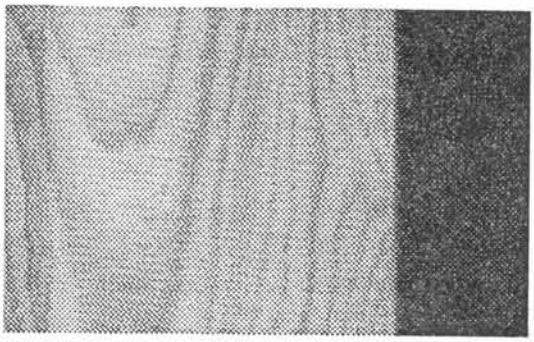


fig5.3 Imagem 2 - madeira de pinho *SDF* e *"background"*

Estatística do 1ºgrau	Imagem 1	Imagem 2
média	114. 99	129. 49
variância	1024. 95	4738. 87
energia	0. 01	0. 01
entropia	6. 95	6. 76
skewness	0. 59	0. 83
kurtosis	1. 31	0. 92

quadro 5-1 Estatística do 1ºgrau, para as imagens 1 e 2, das fig. 5.2 e 5.3

Uma imagem com um desvio padrão elevado, pode significar uma imagem com um contraste elevado. A variância, que é um momento de ordem 2, pode com eficácia discriminar texturas. É usual utiliza-la através do indicador *D*, que se calcula do seguinte modo

$$D = 1 - \frac{1}{1 - \sigma^2} \qquad (5.8)$$

O descritor *D* é nulo para texturas uniformes, e tende para o valor unitário em texturas caracterizadas por frequências largas e grosseiras das suas microestruturas, caracterizadas por fortes variações dos níveis de cinzento. Contudo é um indicador limitado, já que não nos dá nenhuma indicação sobre a posição relativa dos pixels. A média e a variância, dão-nos apesar de tudo, uma ideia do brilho e contraste da imagem, respectivamente.

Quando se pretende classificar imagens através de medidas estatísticas do 1º grau, as imagens são usualmente normalizadas, de modo a que tenham a mesma média e o mesmo desvio padrão. Saliente-se que as imagens são afectadas pelas condições de aquisição iniciais, respectivamente nível e forma de iluminação, além da sua resolução.

Para poder testar duas imagens, uma com média μ e outra com média μ_{nv}, temos de garantir que ambas possuam a mesma média. Para tal adicionamos $|\mu - \mu_{nv}|$, a diferença entre as médias existentes, a cada um dos pixels da primeira imagem. A nova média é igual a:

$$\mu_{nv} = \frac{1}{n}\sum_{i=0}^{L-1}(I_i + \mu_{nv} - \mu) \qquad (5.9)$$

em que n é o número de pixels na imagem.

Se o objectivo pretendido consistir em trabalhar com imagens que apresemtem o mesmo desvio padrão, faz-se o acerto multiplicando o valor de cada pixel, pelo quociente entre o desvio padrão pretendido e aquele que existir na imagem. O novo valor do *pixel* R_i é calculado pela expressão seguinte:

$$R_i = I_i\,(\sigma_{nv}/\sigma) \qquad (5.10)$$

O valor do R_i resultante da transformação da imagem I_i, afecta contudo a média da imagem. Para conseguir uma imagem com a mesma média e o mesmo desvio padrão, multiplico a diferença, entre o nível de cinzento de cada ponto da imagem e média μ pelo quociente (σ_{nv}/σ)

$$R_i = \mu + (I_i - \mu)\frac{\sigma_{nv}}{\sigma} \qquad (5.11)$$

Estas operações correspondem a um remapeamento da imagem, através de operações lineares.

5.3-2 MÉTODO *SGLDM* ("SPACIAL GRAY LEVEL DEPENDENCE MATRICES")

Um método de descriminação de texturas que segue uma aproximação estatística, é o método *SGLDM* [*"Spacial Gray Level Dependence Matrices"*]., que se baseia na análise de uma série de histogramas direccionais (2D), dependentes da

relação espacial entre pixels na imagem, tomados dois a dois. O processo consiste na extracção de medidas, com base na determinação estatística de 2ª ordem, da função densidade de probabilidade condicional $f(i,j/d,\theta)$. São as matrizes de co-ocorrência $S_{d,\theta}[i,j]$, que representam estas densidades de probabilidade. Cada coeficiente de uma matriz de co-ocorrência representa o valor de $p_\delta(i,j)=p(i,j/d,\theta)$, que mede a probabilidade de ocorrerência na imagem de um par de pixels com os níveis de cinzento *'i'* e *'j'*, com a separação δ entre eles. O vector $\delta=(d,\theta)$ representa, em coordenadas polares, a posição de dois pixels numa imagem, separados pela distância *d*, medidas segundo a direcção θ .

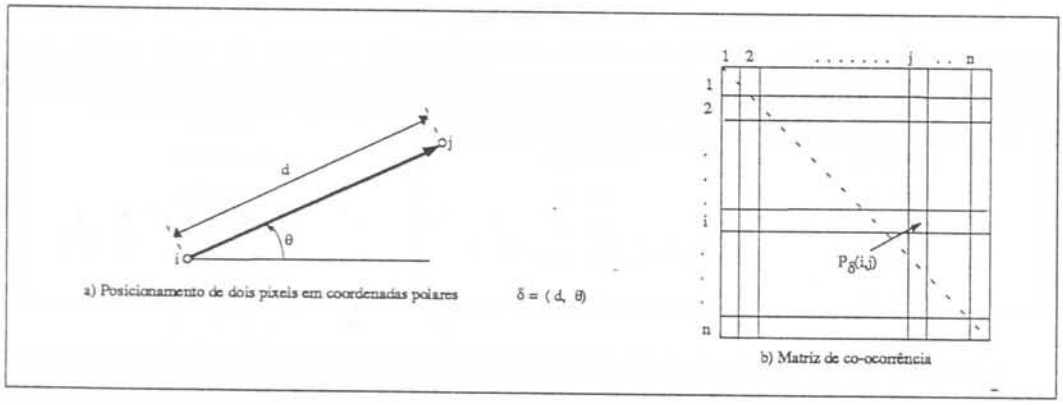


fig. 5.4 Posicionamento de pixels e matriz de co-ocorrência

$I(l,k)$ - nível de cinzento de um *pixel*, medido na linha l e coluna k, da imagem I

L - número de níveis de cinzento

$$0 \leq I(l,k) < L \qquad (5.12)$$

Consideram-se dois parâmetros de orientação:

d - medidas de distância espacial entre dois pixels (imagem digitalizada)

θ - medida do ângulo de separação entre dois pixels

Normalmente os valores mais usados para *'d'*, são: d= 1, 2, 4, 8, 16 e para θ, os valores mais usados são: θ= 0º, 45º, 90 º e 135º

Os ângulos θ' =180º , 225º , 270º , 315º, não são considerados, já que as matrizes de co-ocorrência $s_{d,\theta}\ [i,j]$ e $s'_{d,\theta}\ [i,j]$, que é a sua transposta, apresentam os mesmos valores para os seus elementos.

O método *SGLDM* é baseado num conjunto de matrizes $s_{d,\theta}\ [i,j]$, cada uma com dimensão de $L*L$. Os valores de $s_{d,\theta}\ [i,j]$ representam o número de pares de pixels na imagem $I(l,k)$, com uma distância entre eles igual a *'d'* e uma orientação θ , em que o primeiro pixel tem um nível de cinzento igual a *'i'* e o segundo pixel tem um nível de cinzento igual a *'j'*.

A matriz de co-ocorrência é construída do seguinte modo:

1 - Escolha de um conjunto de valores adequados para *'d'* e 'θ ';

2 - Colocam-se a zero os elementos das matrizes $s_{d,\theta}\ [i,j]$

3 - Para cada valor $(d,\theta\ )$, varre-se a imagem $I(l,k)$. Para cada pixel r localizado nas coordenadas (l,K), encontram-se as coordenadas (l',k') do seu parceiro de teste r', de tal modo que a distância (r,r') seja igual a d, para um ângulo θ .

4 - Incrementar de uma unidade o valor de:

$$s_{d,\theta}\ [I(l,k),I(l',k')] \tag{5.13}$$

5 - Encontrar o valor da matriz normalizada:

$$S_{d,\theta}[i,j] = \frac{S_{d,\theta}[i,j]}{\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} S_{d,\theta}[i,j]} \tag{5.14}$$

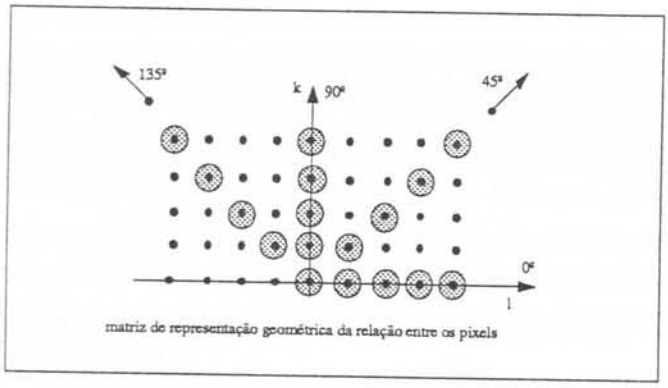


fig. 5.5 Construção da matriz de co-ocorrência

As figuras 5.6 e 5.7 representam a densidade espacial, da probabilidade de ocorrência das duas variáveis *l* e *k*, para *d=1* e θ *=0º*. Correspondem aos valores encontrados para as matrizes de co-ocorrência, obtidas das imagens das fig. 5.2 e 5.3, realçando as dependências entre pixels/níveis de cinzento na imagem. É visível pelos gráficos das fig.5.6 e 5.7, e para estes valores dos parâmetros *d* e θ , que os níveis de cinzento dos pixels na imagem estão próximos, observável nos gráficos 2D das figuras pela distribuição dos valores mais frequentes, próximos da diagonal, denotando imagens com baixo contraste. Os picos do histograma 2D correspondem a zonas de cor uniforme, enquanto os picos distintos, correspondem a zonas diversas na imagem.

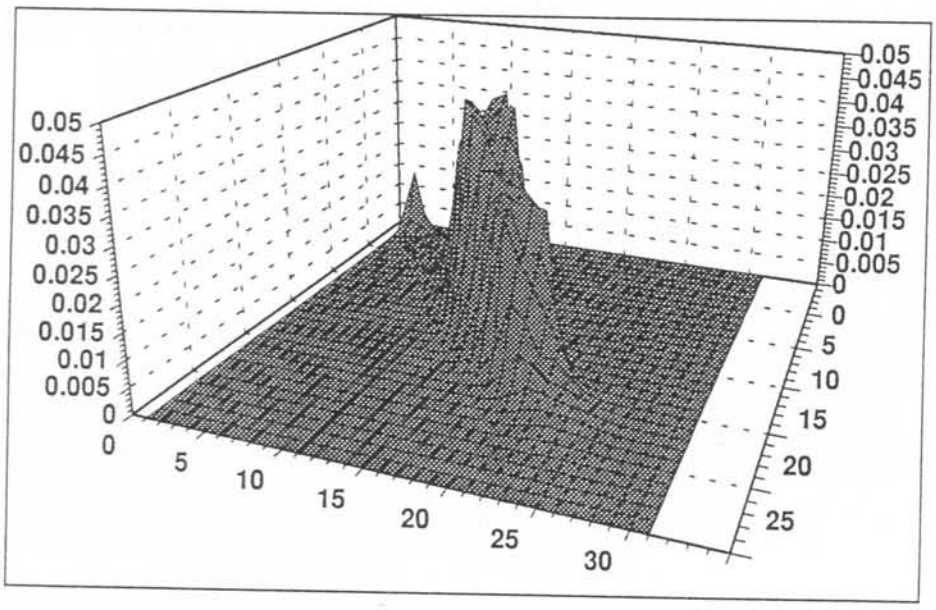


fig. 5.6 Densidade de probabilidade de duas variáveis, para a imagem da figura 5.2 com d=1 e θ=0º.

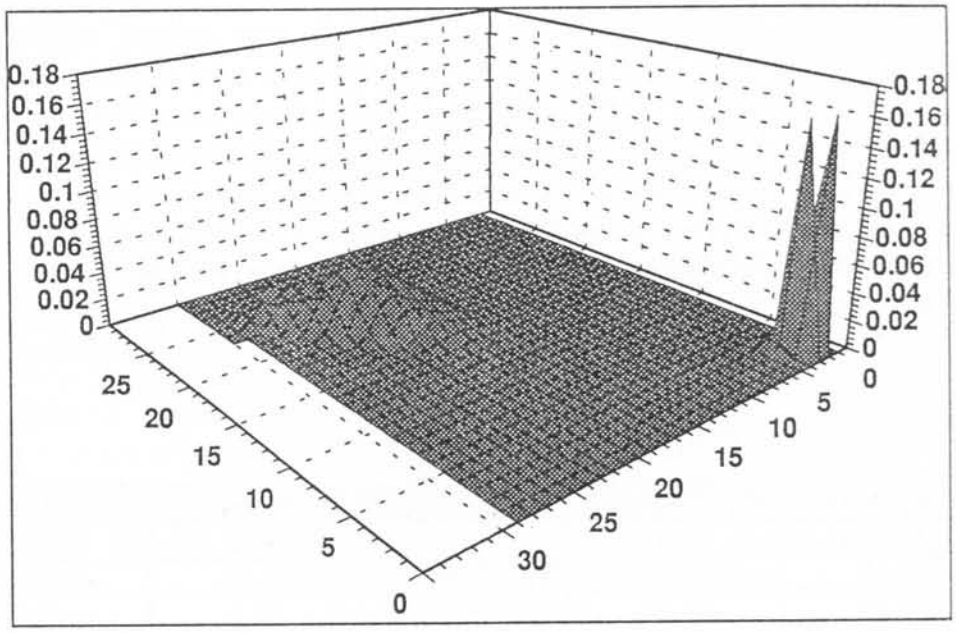


fig. 5.7 Densidade de probabilidade de duas variáveis, para a imagem da figura 5.3 com d=1 e θ=0º

É possível agora analisar a imagem em termos de textura, através da utilização da matriz de co-ocorrência obtida pelo método *SGLDMs*, através de um conjunto de medidas características. A descriminação de texturas, é conseguida deste modo, no espaço das características (através de um número reduzido de variáveis), em vez de ter de o fazer no espaço de padrões de imagem. As medidas estatísticas utilizadas, mais usuais, são as seguintes:

Energia ou Segundo Momento Angular:

$$E(S_{\theta,d}) = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} P_{\delta}(i,j)^2 \qquad (5.15)$$

Entropia:

$$H(S_{\theta,d}) = -\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} P_{\delta}(i,j)\log P_{\delta}(i,j) \qquad (5.16)$$

Inércia ou Contraste:

$$I(S_{\theta,d}) = \sum_{k=0}^{L-1} k^2 \sum_{\substack{i=0 \\ |i-j|=k}}^{L-1} \sum_{j=0}^{L-1} P_\delta(i-j) \tag{5.17}$$

Correlação:

$$C(S_{\theta,d}) = \frac{\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}(i-\mu_x)(j-\mu_y)P_\delta(i,j)}{\sigma_x\sigma_y}$$

onde μ_x, μ_y, σ_x e σ_y são dados por

$$\begin{aligned}
\mu_x &= \sum_{i=0}^{L-1} i \sum_{j=0}^{L-1} P_{\theta,d}(i,j) \\
\mu_y &= \sum_{j=0}^{L-1} j \sum_{i=0}^{L-1} P_{\theta,d}(i,j) \\
\sigma_x^2 &= \sum_{i=0}^{L-1} (i-\mu_x)^2 \sum_{j=0}^{L-1} P_\delta(i,j) \\
\sigma_y^2 &= \sum_{j=0}^{L-1} (j-\mu_y)^2 \sum_{i=0}^{L-1} P_\delta(i,j)
\end{aligned} \tag{5.18}$$

Homogeneidade Local:

$$L(S_{\theta,d}) = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} \frac{1}{1+(i-j)^2} P_\delta(i,j) \tag{5.19}$$

Em relação às medidas indicadas anteriormente, pode-se dizer que a *Energia* é uma medida da homogeneidade da imagem. Numa imagem homogénea há poucos níveis de cinzento diferentes, o que leva a que os coeficientes da matriz de co-ocorrência sejam grandes, e a maioria deles nulos. Em consequência o valor da entropia é grande para imagens homogéneas. Se a imagem tiver grandes variações dos níveis de cinzento o seu valor será pequeno.

A *correlação* é uma medida da linearidade da imagem. Um valor grande para a *correlação* implica uma linearidade elevada na imagem, assim como uma *correlação* elevada para matrizes de co-ocorrência com a direcção θ, implica a existência de estruturas lineares nessa direcção.

É necessário sublinhar que as medidas referenciadas são influenciadas pelos valores de *'d'* e de *'θ '* escolhidos para as matrizes de co-ocorrência, já que são eles que especificam o domínio local desta matriz. A tabela 6-2 representa quatro medidas estatísticas SGLDM[3], *energia*, *entropia*, *contraste* e *correlação*, referentes às imagens das fig. 6.2 e 6.3, para as direcções θ=0º, θ=90º e θ=45º, para distâncias entre pixels de *d=1* e *d=2*.

SGLDM(θ=θ)	Imagem 1(d=1)	Imagem 2(d=1)	Imagem 1(d=2)	Imagem 2(d=2)
Energia	0. 05	0. 1	0. 04	0. 09
Entropia	11. 59	10. 86	12. 36	11. 41
Contraste	6. 24	7. 40	10. 61	12. 67
Correlação	16. 58	22. 17	16. 71	22. 33
SGLDM(θ=90)	Imagem1(d=1)	Imagem2(d=1)	Imagem1(d=2)	Imagem2(d=2)
Energia	0. 04	0. 12	0. 06	0. 10
Entropia	12. 52	10. 09	11. 89	0. 75
Contraste	12. 35	3. 36	7. 25	5. 59
Correlação	16. 59	22. 36	17. 00	22. 70
SGLDM(θ=45)	Imagem1(d=1)	Imagem2(d=1)	Imagem1(d=2)	Imagem2(d=2)
Energia	0. 05	0. 09	0. 05	0. 09
Entropia	12. 30	10. 90	12. 53	11. 37
Contraste	10. 60	6. 40	11. 26	10. 12
Correlação	16. 82	22. 50	17. 35	23. 13

quadro 5-2 Medidas estatísticas do 2ºgrau, para as imagens 1 e 2, das fig5.2 e 5.3

Importa realçar a propósito destas medidas que um efeito perverso na aquisição de duas imagens com a mesma textura pode ser introduzido, se as suas condições de iluminação forem diferentes, podendo levar a que medidas de textura desajustadas sejam encontradas. Equanto que para a visão humana, tonalidades diferentes da mesma textura, não comprometem a avaliação da textura, sendo reconhecidas como a mesma, para a máquina isso pode não acontecer.

Procura-se na prática, des sensibilizar a avaliação (ou classificação) da imagem, em relação às condições de iluminação com que foi feita a aquisição. Duas

[3]São utilizadas as funções mCooVr10(...), mCooVr20(...), mCooVr90(...), mCooVr29(...), mCoVr150(...) e mCoVr245(...) que retiram as matrizes de coocorrência para θ de 0º, 90º, e 45º ,no arquivo TTXTURAS.C. As funções As_m(...), Entp(...), Contrast(...), correlaC(...) em TFUNC.C, que retiram os valores de energia, entropia, contraste e correlação nas imagens.

situações podem acontecer. Ou as imagens são invariantes em relação às transformações monotónicas dos níveis de cinzento, ou é necessário torna-las invariantes, por exemplo através da equalização do histograma. Acrescente-se por exemplo que a Entropia e o 2º Momento angular, são invariantes a este tipo de transformações.

5.3-3 MÉTODO *GLDM* (*"GRAY LEVEL DIFFERENCE METHOD"*)

Como no processo anterior, este método utiliza medidas estatísticas de segunda ordem. Consideremos $I(l,m)$ a imagem à qual foi aplicado o método, e δ um vector definido através de $\delta = [\Delta l, \Delta m]$, onde Δl e Δm são inteiros que definem o espaçamento entre pixels, tal que $I_\delta(l,m)$ é o valor absoluto da diferença entre os níveis de cinzento nos dois pontos respectivos.

$$I_\delta(l,m) = |I(l,m) - I(l + \Delta l, m + \Delta m)| = |i-j| \qquad (5.20)$$

i e j são os níveis de cinzento nesses pontos. A função densidade de probabilidades associada aos valores I_δ, pode ser determinada por:

$$P(I_\delta(l,m)=k) = P_\delta(i,j) = P_\delta(k) \quad \text{em que } k=|i-j| \qquad (5.21)$$

$$P_\delta(k) = \sum_{\substack{i=0 \\ |i-j|=k}}^{L} \sum_{j=0}^{L} P_\delta(i,j) \qquad (5.22)$$

Se na imagem existirem N níveis de cinzento, esta função assume a forma de um vector com dimensão N. Os coeficientes deste vector, determinam o número de vezes que os níveis de cinzento, com diferenças iguais a k, se encontram na imagem, para a relação espacial especificada por δ.

O valor da função $P_\delta(k)$ pode ser obtida da matriz de co-ocorrência, a partir dos coeficientes desta matriz, que se encontrem segundo as diagonais paralelas e simétricas à da diagonal principal da mesma.

Estes valores da diferença entre níveis de cinzento (entre pixels), para o desfasamento δ, contêm informação relevante sobre a textura. Para texturas finas e

desfasamentos δ na direcção dessa textura e dimensão comparável ao período da textura, são encontrados valores elevados de $I_\delta\ (l,m)$, com grande frequência, e portanto também valores da probabilidade $P_\delta\ (k)$ elevados.

Com os valores de $P_\delta\ (k)$, é possível determinar cinco medidas, caracterizadoras da textura da imagem:

Contraste

$$C = \sum_{k=0}^{L-1} k^2 P_\delta(k) \tag{5.23}$$

Segundo Momento Angular

$$ASM = \sum_{k=0}^{L-1} P_\delta(k)^2 \tag{5.24}$$

Entropia

$$ENT = -\sum_{k=0}^{L-1} P_\delta(k) \log P_\delta(k) \tag{5.25}$$

Momento Inverso às diferenças

$$IDM = \sum_{k=0}^{L-1} \frac{P_\delta(k)}{k^2+1} \tag{5.26}$$

Média às diferenças

$$M = \sum_{k=0}^{L-1} k P_\delta(k) \tag{5.27}$$

O quadro da tab. 5.3 representa valores estatísticos retirados das imagens das fig. 5.2 e fig. 5.3 pelo método GLDM[4], para as direcções $\theta = 0^{\circ}$ e $\theta = 90^{\circ}$ e desvios entre pixels de valor $\delta = 1$ e $\delta = 2$.

GLDM(θ=0°)	Imagem1(δ=1)	Imagem2(δ=1)	Imagem1(δ=2)	Imagem2(δ=2)
média	11. 48	11. 63	14. 75	14. 85
contraste	183. 57	219. 97	318. 51	382. 80
energia	0. 04	0. 04	0. 03	0. 03
entropia	4. 78	4. 92	5. 20	5. 31
momento inverso	0. 07	0. 12	0. 06	0. 10
GLDM(θ=90°)	Imagem1(δ=1)	Imagem2(δ=1)	Imagem1(δ=2)	Imagem2(δ=2)
média	14. 90	6. 51	10. 94	9. 07
contraste	374. 28	93. 75	213. 48	162. 70
energia	0. 03	0. 08	0. 04	0. 06
entropia	5. 30	4. 17	4. 91	4. 64
momento inverso	0. 07	0. 19	0. 11	0. 15

quadro 5.3 Medidas *GLDM* retiradas das imagens 1 e 2, representadas nas fig. 5.2 e fig. 5.3

5.3-4 MÉTODO *GLRLM* (*"GRAY LEVEL RUN LENGTH METHOD"*)

Uma cadeia mono-dimensional de níveis de cinzento, consiste num conjunto de pixels linearmente conexos, possuindo o mesmo nível de cinzento. O comprimento da cadeia é igual ao número de pixels que a constituem. O comprimento das cadeias são calculados para uma direcção específica, e por cada direcção especificada é construída uma matriz. O elemento $p(i, l/\theta)$ de níveis de cinzento de uma cadeia da matriz $R(\theta)$ - *"run length"*, é dado por:

$$R(\theta) = [\, p(i, l/\theta) \,] \qquad (5.28)$$

específica o número estimado de vezes que encontro na imagem a cadeia de comprimento l com o nível de cinzento correspondente a *'i'*, na direcção θ.

[4]São utilizadas no método GLDM as funções mdif10(...), mdif20(...), mdif190(...), mdif290(...) em TTXTURAS.C e difest1(...) em TFUNC.C .

Galloway[5], aplica o método a imagens de 64 níveis de cinzento, através da quantificação desses mesmos níveis de cinzento em intervalos de oito níveis de espaçamento, considerando uma partição logarítmica de base 2 para os comprimentos *'l'* de: 1, 2-3, 4-7, 8-15, 16-31, 32-64.

São usualmente utilizadas cinco medidas para caracterizar uma imagem, através do método do comprimento das cadeias dos níveis de cinzento.

Enfase das cadeias curtas ("Short Run Enphasis")

$$SRE = \frac{\sum_{i=0}^{N_g}\sum_{l=1}^{N_r}\frac{p(i,l|\theta)}{l^2}}{\sum_{i=0}^{N_g}\sum_{l=1}^{N_r}p(i,l|\theta)} \qquad (5.29)$$

Enfase das cadeias longas ("long Run Enphasis")

$$LRE = \frac{\sum_{i=0}^{N_g}\sum_{l=1}^{N_r}l^2 P(i,l|\theta\ )}{\sum_{i=0}^{N_g}\sum_{l=1}^{N_r}P(i,l|\theta\ )} \qquad (5.30)$$

Não uniformidade do comprimento das cadeias ("Run length Distribution")

$$RLN = \frac{\sum_{l=1}^{N_r}\left(\sum_{i=0}^{N_g}P(i,l|\theta\ )\right)^2}{\sum_{i=0}^{N_g}\sum_{l=1}^{N_r}P(i,l|\theta\ )} \qquad (5.31)$$

[5]Citado por Tomita Fumiaki e Saburo Tsuji, em [31-A].

Não uniformidade dos níveis de cinzento ("Gray level Distribution")

$$NNG = \frac{\sum_{i=0}^{N_g}\left(\sum_{l=1}^{N_r} P(i,l|\theta\ )\right)^2}{\sum_{i=0}^{N_g}\sum_{l=1}^{N_r} P(i,l|\theta\ )} \qquad (5.32)$$

Percentagem das cadeias ("Run Percentage")

$$RP = \frac{\sum_{i=0}^{N_g}\sum_{l=1}^{N_r} P(i,l|\theta\ )}{\sum_{i=0}^{N_g}\sum_{l=1}^{N_r} l.P(i,l|\theta\ )} \qquad (5.33)$$

As grandezas representadas por N_g e N_r representam respectivamente o número de níveis de cinzento para as cadeias e o número de cadeias definidas.

Em imagens com texturas grosseiras, as cadeias de dimensão importante são mais frequentes, e têm maior valor se forem medidas na direcção da textura.

O quadro 5.4, representa as medidas estatísticas retiradas das imagens referenciadas nas fig. 5.2 e fig. 5.3 pelo método GLRLM[6], respectivamente *SRE - enfase nas cadeias curtas*, *LRE - enfase nas cadeias longas*, *RLN - não uniformidade das cadeias*, *NNG - não uniformidade dos níveis de cinzento*, e *RP - percentagem da cadeia.*

GLRLM	Imagem1(θ=0)	Imagem2(θ=0)	Imagem1(θ=90)	Imagem2(θ=90)
SRE	0. 83	0. 74	0. 79	0. 46
LRE	0. 85	0. 85	0. 87	0. 64
RLN	1. 70	3. 69	2. 39	8. 69
NNG	0. 69	0. 67	0. 73	0. 41
RP	0. 08	0. 08	0. 08	0. 08

quadro 5.4 Quadro de medidas características obtidas das imagens representadas nas fig. 5.2 e fig. 5.3, obtidas pelo método *GLRLM*

[6]São utilizadas neste método GLRLM, as funções mrlgth0(...) , mrlgth0(...) em TTXTURAS.C e rlgthmd(...) em TRCMDPNH.C.

Um exemplo adicional de medidas retiradas de amostras de defeitos existentes na madeira, pode ser sugerido como forma de realce das diferenças existentes entre eles (defeitos diferentes e comuns em pranchas de madeira). Esses defeitos podem ser identificados através de classificação, baseada em vectores de características obtidos por processos de medição estatística. A tabela 5.5 mostra medidas, escolhidas ao acaso, e obtidas por registo directo sobre pranchas de madeira. Foram utilizados os métodos estatísticos estudados até aqui, evidenciando alguma diferença entre elas, o que lhes dá potencialidades para poderem integrar um processo de classificação de defeitos em madeira.

ESTATÍSTICA	CASCA	BOLSA DE RESINA	NÓ
Média(1ºgrau)	96	114	88
Média(GLDMd=1/0º)	15,36	18. 68	10. 92
Contraste(GLDM)	321. 72	497. 62	156. 23
SRE(GLRLM)	0. 89	0. 91	0. 86
RLN(GLRLM)	0. 75	0. 79	0. 70

quadro 5.5 Exemplo de medidas características de natureza estatísticas, encontradas para os defeitos, casca, bolsa de resina e nós em prancha de madeira de pinho

5.4 TEXTURA: MÉTODOS ESPECTRAIS. ANÁLISE DO ESPECTRO DE POTÊNCIA DE FOURIER

Até aqui estudámos os métodos ditos estatísticos. Cabe agora apenas uma referência aos métodos baseados na frequência.

As texturas podem ser analisadas no domínio da frequência, através da transformada de Fourier da imagem. Se $f(i,j)$ for o valor de uma imagem discretizada no ponto (i,j), imagem de dimensão MxN, a transformada discreta de Fourier (no domínio discreto a duas dimensões) é definida por:

$$F(u,v) = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} f(i,j)\exp^{-2\pi\sqrt{-1}(ui/M+vj/N)} \qquad (5.34)$$

O espectro de potência que representa o valor do módulo de cada frequência espacial, pode ser obtida por

$$P(u,v) = |F(u,v)|^2 \qquad (5.35)$$

O espectro de Fourier é um bom indicador da direcionalidade da textura numa imagem.

As características de Fourier, que nos dão essa informação, são:

. Análise da amplitude máxima (indica a direcção dos padrões de textura)

. Localização dos máximos (indica a periodicidade espacial associada com o padrão de textura).

Texturas finas, apresentam uma concentração espectral espacial, concentrada nas altas frequências. Enquanto que texturas grosseiras, têm o seu espectro concentrado mais nas baixas frequências.

Se uma textura for periódica, o espectro de potência terá valores elevados para as frequências espaciais correspondentes. Se uma textura for homogénea o espectro terá valores elevados concentrados à volta da origem. Pelo contrário os valores do espectro de potência serão mais dispersos para uma textura fina.

Na determinação da transformada discreta de Fourier, a imagem é trata como periódica. Se os bordos inferior e superior, bem como o esquerdo e direito forem diferentes, implicará que se comportem como se de facto existissem arestas abruptas na imagem de entrada, o que influencia fortemente a forma do espectro e distorce a informação que ela contém.

5.5 TEXTURA BASEADA NAS PROPRIEDADES ESPACIAIS DA GEOMETRIA

5.5-1 CARACTERÍSTICAS GEOMÉTRICAS LOCAIS

Um método descrito por Tomita Fumiaki e Saburo[30-A], baseia-se na medição de características genéricas locais. Se *f(i,j)* representar o valor da intensidade luminosa no ponto *(i,j)* na imagem, a dimensão de uma aresta (*"edge"*) no ponto *(i,j)* é calculado por:

$$|e(i,j)| = \sqrt{f_i(i,j)^2 + f_j(i,j)^2} \qquad (5.36\text{-a})$$

e o argumento representado por

$$\angle e(i,j) = \tan^{-1} \frac{f_j(i,j)}{f_i(i,j)} + \frac{\pi}{2}$$

onde (5.36-b)

$$f_i(i,j) = f(i+1,j) - f(i-1,j)$$
$$f_j(i,j) = f(i,j+1) - f(i,j-1)$$

Um conjunto de medidas estatísticas podem ser calculadas, a partir da imagem gerada por estes operadores.

"Coarseness" - a densidade dos *'edges'* é uma medida de textura. Quanto mais fina for a textura maior é a densidade.

Contraste - A média da dimensão dos *edges* é uma medida do contraste na imagem

"Randomness" - A entropia calculada do histograma de mapeamento dos *edges*, é uma medida de aleatoriedade da textura.

Directividade - É determinada no histograma da direcção dos *edges*. A entropia retirada deste histograma, dá uma medida da directividade da textura. A determinação de *clusters* no histograma indica directividade na textura, enquanto que uma distribuição uniforme, indica ausência de directividade na imagem.

A partir da direcção dos *'edges'*, também se podem extrair propriedades estruturais a partir de entidades de 2ª ordem, que relacionem *edges* em posições contíguas na imagem, em direcções especificadas.

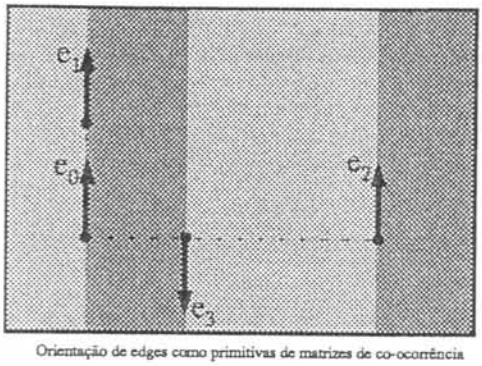


fig5.8 Texturas de 2ªordem determinadas pela distribuição da orientação dos *edges* na imagem

Linearidade - A probabilidade de co-ocorrência de dois *"edges"* com a mesma direcção e separados pela distância k na direcção do *edge*, dá-nos uma medida da linearidade na imagem.

A figura mostra quais os *"edges"* que podem servir como medida de linearidade da imagem (e_0 e e_1 com a mesma direcção, e separados da distância k, com a mesma direcção na sua orientação).

Periodicidade - É a probabilidade de co-ocorrência de *"edges"* com a mesma direcção, separados por uma distância igual a k, ao longo de uma direcção perpendicular à direcção do *edge* (e_0 e e_2) e indicam a periodicidade da textura.

Espaçamento (*size*) - A probabilidade de co-ocorrência de dois *"edges"* com direcções opostas, separados de uma distância k, na direcção perpendicular à direcção do *edge* (e_0 e e_3) dão uma medida do espaçamento entre elementos da textura.

5.5-2 MEDIDAS DE EXTREMOS RELATIVOS

Há muita informação sobre texturas, que se concentra na frequência relativa dos extremos locais da intensidade de cada pixel. Podemos considerar dois tipos de extremos locais: *Unidimensionais* e *bidimensionais*.

Utiliza-se o cálculo do número de extremos por unidade de superfície como medida de textura. No caso *unidimensional* podemos definir os extremos do modo seguinte:

- Numa linha, um pixel *i* é mínimo relativo se se verificar a relação seguinte:

$$f(i,j) \leq f(i+1,j) \quad \text{e} \quad f(i,j) \leq f(i-1,j) \tag{5.37}$$

- será máximo relativo se

$$f(i,j) \geq f(i+1,j) \quad \text{e} \quad f(i,j) \geq f(i-1,j) \tag{5.38}$$

Depois de marcar os máximos, centra-se uma janela sobre cada um dos pixels e contam-se o número de pixels marcados.

No caso unidimensional duas propriedades podem ser associadas a cada extremo, a sua altura e a sua largura. A altura dum máximo pode ser definido pela diferença entre o valor do máximo e do mínimo adjacente mais elevado. A largura como a distância entre os dois mínimos que lhe são adjacentes.

Sendo a textura um fenómeno bidimensional, podem ser definidos também máximos e mínimos bidimensionais, em função de vizinhos segundo as direcções horizontal e vertical. Contudo eles apresentam um maior peso de computação no seu cálculo.

5.5-3 ARESTAS POR UNIDADE DE SUPERFÍCIE

Neste caso, toma-se o número de arestas por unidade de superfície, como uma medida de textura. Calcula-se o gradiente sobre toda a imagem, e com uma janela centrada sobre cada pixel, determina-se a distribuição dos valores do gradiente pela imagem. A média dessa distribuição, é uma medida associada a essa janela, relacionada com a quantidade média de arestas ou *"edges"* na imagem.

5-6 MEDIDAS DE ENERGIA DAS TEXTURAS

Este método proposto por Laws[25-A] é um método espacial estatístico, já que se baseia em cálculos de medidas baseadas em funções locais da imagem. As medidas são espaciais porque as funções locais tomam em consideração vários pixels em simultâneo, de acordo com:

- As funções locais são determinadas efectuando a convolução das imagens através de janelas, de dimensões que podem variar de; 3x3; 5x5 ou 7x7. São as micro-janelads.

- A avaliação é efectuada através de medidas estatísticas como, médias, ou momentos estatísticos de ordem vária, sobre a imagem resultante da aplicação da convolução anterior. São utilizadas no cálculo das medidas estatísticas janelas de maior dimensão, de um dos valores seguintes: 15x15, 31x31, etc., a cada um dos pixels da imagem.

5-7 AUTO-CORRELAÇÃO

A função auto-correlação espacial, pode funcionar como medida de textura. Se $I(m,n)$ designar a função imagem, a auto-correlação pode ser calculada por:

$$A(\varepsilon,\eta;j,k)=\frac{\sum_{m=j-w}^{j+w}\sum_{n=k-w}^{k+w}I(m,n)I(m-\varepsilon,n-\eta)}{\sum_{m=j-w}^{j+w}\sum_{n=k-w}^{k+w}I(m,n)^2} \qquad (5.39)$$

A função é calculada, por aplicação de janelas de dimensão $(2w+1)$x$(2w-1)$ em cada ponto da imagem (j,k) para valores do desvio de $\varepsilon,\eta = 0, \pm1, \pm2, \ldots, T$.

A função auto-correlação da imagem mede o grau de semelhança, entre a imagem e uma versão de si própria. A auto-correlação pode fornecer dois tipos de informação: O grau de rudeza da imagem(*"coarseness"*) da textura e a *periodicidade*.

Se a textura for grosseira, as primitivas serão relativamente grandes e o valor da auto-correlação elevada para desvios (ε,η) fixos, diminuindo com a distância. Se a textura for fina o valor diminuirá rapidamente com a distância.

Se a textura for periódica a função de auto-correlação terá variações periódicas.

5.8 CONCLUSÕES

Depois de uma introdução sobre os conceitos teóricos relacionados com o significado de textura, separou-se o estudo das texturas, naquelas que apresentavam predominância estrutural, a que se chamamos determinísticas, e as que apresentavam organização aleatória, estudadas através de análise estatística.

Sendo colocada a incidência neste último tipo de texturas, abordaram-se métodos estatísticos de análise, de primeira e segunda ordem. Dentro destes, analisaram-se vários métodos: método *SGLDM* ("*Spacial Gray Level Dependence Matrices*"), método GLDM ("*gray level difference method*") e o método *GLRLM* ("*gray level length method*"). Foram estudados ainda outros processos, que utilizam medidas estatísticas de segunda ordem, cujas propriedades da textura não derivam directamente do valor de cinzento dos pixels da imagem, mas de processos de leitura indirectos, como são o caso da medida dos *"edges"* ou da sua orientação na imagem, ou ainda medidas relativas ou ligadas ao número de máximos e mínimos na imagem.

Sendo um dos objectivos deste trabalho, o *reconhecimento de defeitos em madeira*, uma das tarefas mais importantes do processo, prende-se com a extracção de medidas características das superfícies ou regiões segmentadas ou delimitadas pelo defeito. Muitas dessas medidas, basearam-se nas características das micro-estruturas dessas mesmas superfícies a que estão ligadas propriedades de textura.

Não sendo possível tirar conclusões imediatas, sobre determinados valores de textura retirados de amostras de madeira, foram apesar disso apresentados exemplos com essas medidas, fundamentalmente com dois objectivos: o de mostrar que essas grandezas apresentavam variações importantes que podem ser capitalizadas num processo de medida e o de mostrar que se a textura não for retirada de regiões com alguma homogeneidade, esses valores podem perder completamente o significado.

CAPÍTULO VI

RECONHECIMENTO DE PADRÕES

CLASSIFICAÇÃO

6 RECONHECIMENTO DE PADRÕES

6.1 CONCEITOS BÁSICOS

Um sistema de reconhecimento de padrões pode ser representado esquematicamente da forma como é ilustrado na fig. 6.1. O diagrama de blocos, referência três sistemas: o sistema de aquisição de imagem, o sistema de processamento de imagem, e o sistema de decisão/classificação. A fase de processamento envolve processos de análise/tratamento e extracção de características da imagem.

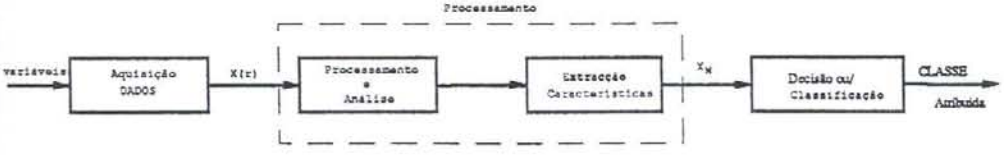


fig. 6.1 Representação por blocos, dum processo de reconhecimento de padrões

Pretende-se com a técnica do reconhecimento de padrões, discriminar através de dados de informação gerados por um sistema, entre várias classes de padrões de atribuição possível. A procura de atributos invariantes entre os membros duma população pertencente ao mesmo tipo de familia, permitirá garantir a estabilidade da classificação.

Segundo Tou e Gonzalez [32-A], o reconhecimento de padrões pode ser entendido como a categorização de dados de entrada, em classes identificáveis, através da extracção de características ou atributos significativos para esses dados, desprezando um conjunto de características irrelevantes. Isto não traduz mais do que a recolha de características comuns a cada classe, que descreva claramente todos os padrões pertencentes a essa classe.

Cada quantidade medida descreve uma característica ou padrão do objecto/região. As características de uma classe de padrões, caracterizam os atributos comuns a todos os padrões pertencentes a essa classe. Se for assumido que o objecto é caracterizado por n medidas, teremos o vector de características dos padrões $x=(x_1,x_2, \ldots ,x_n)^T$, caracterizado por uma dimensão n e contendo a informação disponível sobre as características dos objectos/regiões.

Um simples exemplo permite ilustrar o que foi dito anteriormente. Considere-se um conjunto de padrões, classificáveis numa das duas classes w_1 e w_2 (classe dos homens e mulheres respectivamente, de um grupo particular) à custa de um vector de padrões constituído pela variável característica altura (componente x_1) e pela variável característica peso (componente x_2). A fig. 6.2, representa os dois agrupamentos desses padrões, distribuídos pelas duas classes referidas w_1 e w_2.

Observa-se através da figura 6.2, que estas duas classes, identificadas por estas duas características (x_1-altura e x_2-peso), podem ser facilmente separáveis.

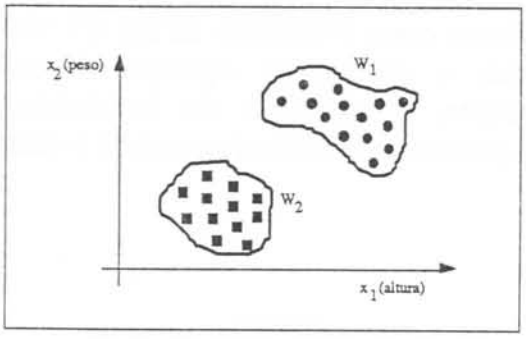


fig. 6.2 Classes de padrões separáveis

O segundo problema envolvido no reconhecimento de padrões relaciona-se com a selecção e extracção de características. A selecção de características procura encontrar o melhor e o menor conjunto de características, que melhor caracterize determinada classe ou conjunto de classes, enquanto a extracção de características procura diminuir a dimensão do vector de características, à custa de projecções sobre sistemas de menor dimensão.

Interessa resumidamente, encontrar um conjunto de características que caracterizem o melhor possível, as classes de padrões envolvidas na classificação. Essas características, devem apresentar uma grande margem de separabilidade, para

poderem oferecer individualmente, capacidade discriminante entre as classes, consideradas duas a duas. Depois de ter efectuado a selecção de características, procura-se diminuir o seu número, de modo a não ter de trabalhar com vectores de muitas componentes, ou seja em espaços de características multidimensionais, já que isso implicaria um peso de cálculo muito maior, e obscureceria a visualização espacial do problema (a representação bi ou tridimensional por vezes é vantajosa).

O terceiro problema envolvido no reconhecimento de padrões, prende-se com a determinação do procedimento de decisão, que possa conduzir aos melhores resultados de identificação (ou classificação) do padrão (ou conjunto de padrões).

Considere-se um conjunto de padrões pertencentes a diferentes classes. Esses padrões serão representados por esse motivo, em regiões diferentes do espaço de características. A separação entre classes é feita por superfícies de decisão, que consoante a dimensão do vector de características, serão: apenas um ponto (se n=1), uma linha (se n=2), um plano (se n=3), como está expresso em (6.1)

$$w_1 x_1 + w_2 x_2 + w_3 = 0 \tag{6.1}$$

e um hiperplano para o caso de n>2, caso geral representado em (6.2)

$$w_1 x_1 + w_2 x_2 + w_3 x_3 + \ldots + w_n x_n + w_{n+1} = 0 \tag{6.2}$$

na forma matricial a função de decisão pode ser expressa por

$$d_k(x) = w_k^T . x = 0 \tag{6.3}$$

onde

$$x = (x_1, x_2, \ldots, x_n, 1)^T \quad \text{e} \quad w = (w_1, w_2, \ldots, w_n, w_{n+1})^T \tag{6.4}$$

Para o caso da fig. 6.3, a função decisão $d(x)=d_k(x) - d_j(x)=0$ será uma equação que define a linha que separa as classes k e j. Podemos dizer que

$$\text{se } d_k(x) > d_j(x) \quad \text{então} \quad \forall x \in w_k \quad \forall\, j \neq k, \quad j=1,2, \ldots, M \tag{6.5}$$

A fronteira tomada entre as duas classes, estará em $d_1(x)=d_2(x)$, ou seja

$$d(x) = d_1(x) - d_2(x) = 0 \tag{6.6}$$

$$d(x) = w_1^T x_1 - w_2^T x_2 = (w_1 - w_2)^T x = 0 \tag{6.7}$$

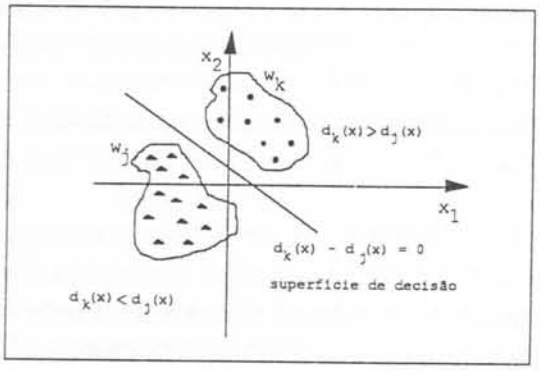


fig. 6.3 Espaço de características a duas dimensões e função de decisão entre duas classes de padrões

Num problema de classificação com *M* classes, *M-1* hiperplanos são suficientes para tomar a decisão. Para o caso particular de três classes, a fig. 6.4 mostra os dois hiperplanos de decisão, no caso duas rectas, já que a dimensão do espaço de características é igual a dois.

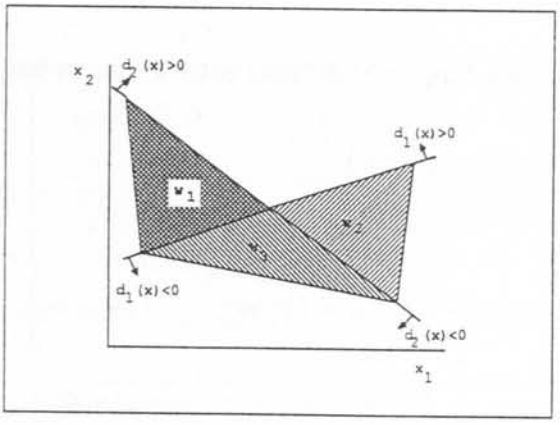


fig. 6.4 Superfícies de separação entre três classes num espaço bidimensional

Ao assumir que num problema de classificação, disponho de *M* classes; w_1 , w_2 . . . , w_M, estou também a considerar que estou a dividir o espaço de caracaerísticas, em *M* classes distintas. A atribuição dum padrão a uma destas regiões, consiste em classificar através duma função de decisão (ou função descriminante) $d_1(x)$, $d_2(x)$, . . . ,$d_M(x)$ o padrão numa dessas regiões do espaço, em que fica dividido o espaço de características.

se a condição $d_K(x) > d_j(x)$ para todo o k , $j = 1, 2, \ldots$,M e com $j \neq k$ for verdadeira, então o padrão x pertencerá à classe de padrões w_k.

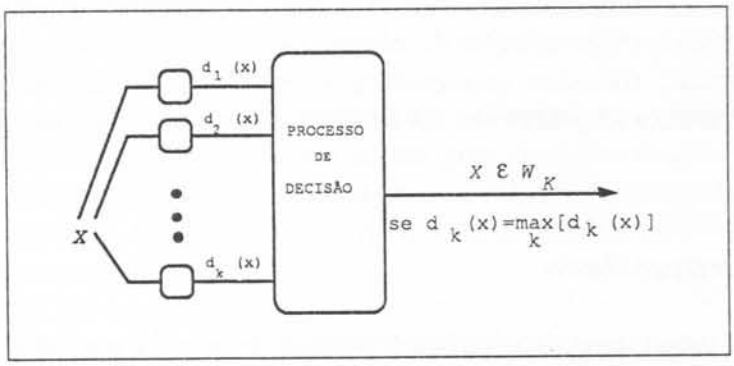


fig. 6.5 Diagrama de blocos dum sistema de classificação

Se houver conhecimento à priori do valor das características a atribuir aos padrões pertencentes a uma classe , pode-se esperar um melhor comportamento do classificador, uma vez que posso utilizar uma técnica de aprendizagem, que permita treinar previamente o classificador antes da sua utilização.

Existem conjuntos de técnicas, que permitem chegar às funções de decisão apropriadas a partir de um conjunto de padrões conhecidos, usando, quer processos determinísticos quer processos estatísticos. A esta operação chama-se aprendizagem supervisada.

Quer o processo de extracção de características quer o processo de obtenção das funções de decisão, pode ser melhorado se se dispuser de uma informação contextual dos padrões, que pode envolver componentes estatísticas, probabilísticas, variações geométricas, etc. Se poderem ser utilizadas estas informações contextuais, chega-se a um sistema de reconhecimento de padrões com características adaptativas, que pode ser descrito como mostra o diagrama da fig. 6.6

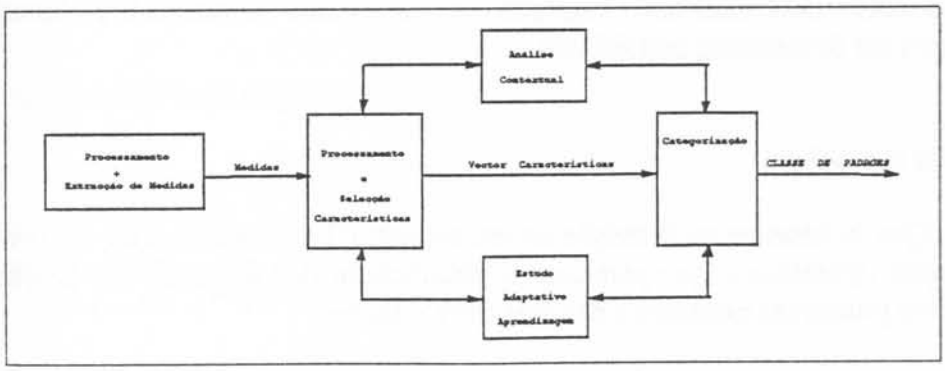


fig. 6.6 Diagrama blocos dum sistema de reconhecimentos de padrões, por um processo adaptativo

Existe um conjunto de metodologias, nos quais se baseia a construção, da maior parte dos classificadores. Essas metodologias assentam fundamentalmente em cinco tipos de técnicas: técnicas que estabelecem correspondências entre um padrão e a imagem; técnicas de aglutinação baseadas nas propriedades de aglomeração de alguma característica representativa da classe; técnicas heurísticas, dependentes do problema particular; métodos matemáticos e estatísticos e métodos sintácticos, baseados em gramáticas e primitivas. Os parágrafos seguintes clarificam melhor essas técnicas.

Técnicas de correspondência

Quando uma classe de padrões é caracterizada pela regularidade dos seus membros, os membros dessa classe podem ser caracterizados em termos da regularidade dessa propriedade. Dois processos podem ser utilizados - Correspondência de forma(*"matching template"*) ou correspondência de propriedade (*"matching features"*).

Técnicas de aglomeração

Quando a classe de padrões é caracterizado por um vector de números reais, a classe pode ser definida pelas suas propriedades de aglomeração em grupos (*"clusters"*). Quando a separação entre os grupos é mal definida, o classificador torna-se mais complexo, e difícil.

Técnicas heurísticas - também podem ser utilizadas, baseadas em conceitos de propriedades comuns, semelhanças entre os seus membros. São técnicas ad hoc, baseadas na experiência.

Métodos matemáticos

Baseados em técnicas e conceitos matemáticos com duas vertentes fundamentais: Determinístico (baseado em processos iterativos) e Estatístico (baseados em ferramentas estatísticas).

Métodos sintácticos

Que se baseiam na definição de um conjunto de primitivas e de um conjunto de regras (gramática), que permitam a classificação dos objectos, em função dos elementos primitivos extraídos e das suas inter relações.

6.2 TEORIA DA DECISÃO POR VIA NÃO PARAMÉTRICA

6.2.1 FUNÇÕES DE DECISÃO LINEARES

Funções de decisão lineares podem ser utilizadas na separação de classes linearmente separáveis, se apresentarem algum grau de aglomeração em função de certas características seleccionadas e alguma distância interclasses, para essas mesmas características. Duda e Hart[13-A] e Tou e Gonzalez[31-A] referem-se a estas funções com esses objectivos.

A distância de um vector de características a uma classe caracterizada por um conjunto de padrões, pode ser encontrada através da regra do vizinho mais próximo. A função de decisão $d_k(x)$ expressa em (6.8), determina a distância mínima de um padrão x a cada uma das M classes definidas para um problema de decisão (através de todos os padrões representativos de cada classe).

$$d_k(x) = \min_{\text{para } m=1,\ldots,N_k} [d_k^m(x)] \qquad k = 1, \ldots, M \tag{6.8}$$

$d_k(x)$ pode ser desenvolvida através de (6.9)

$$d_k^m(x) = w_{k1}^m x_1 + w_{k2}^m x_2 + \ldots + w_{kn}^m x_n + w_{k,n+1}^m \tag{6.9}$$

ou na forma matricial, por (6.10)

$$d_k^m(x) = (w_k^m)^T x \tag{6.10}$$

em que w_k^m está expressa em (6.11)

$$\mathbf{w}_k^m = \begin{bmatrix} w_{k1}^m \\ w_{k2}^m \\ \vdots \\ w_{k,n+1}^m \end{bmatrix} \tag{6.11}$$

e x em (6.12)

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \\ 1 \end{bmatrix} \qquad 6.12)$$

três casos podem ser enumerados:

caso 1 - Cada classe de padrões, é separável das outras classes por uma única superfície de decisão

$$d_i(x) = w_i^T x \begin{cases} > 0 & \text{se} x \in w_i \\ < 0 & \text{em caso contrário} \end{cases} \qquad i = 1,2, \ldots, M \qquad (6.13)$$

caso 2 - Cada classe de padrões é separável de todas as outras classes, duas a duas, por uma superfície de decisão.

Por exemplo w_1 pode ser separado de w_2 por $d_{12}(x)=0$ e de w_3 por $d_{13}(x)=0$

$$\begin{aligned} d_{ij}(x) &= d_i(x) - d_j(x) \\ &= (w_i - w_j)^T x \\ &= w^T_{ij}\, x \end{aligned} \qquad (6.14)$$

A superfície de decisão é representada por $d_{ij}(x) = 0$

$$x \in w_i \qquad \text{se} \qquad d_{ij}(x) > 0 \qquad \forall\, j \neq i \qquad (6.15)$$

caso 3 - As classes são separáveis por uma superfície de decisão, duas a duas, mas sem zonas de indeterminação

$$d_k(x) = (w_k)^T x \qquad k = 1,2, \ldots, M \qquad (6.16)$$

em que

$$x \in w_i \qquad \text{se} \qquad d_i(x) > d_j(x) \qquad \forall\, j \neq i \qquad (6.17)$$

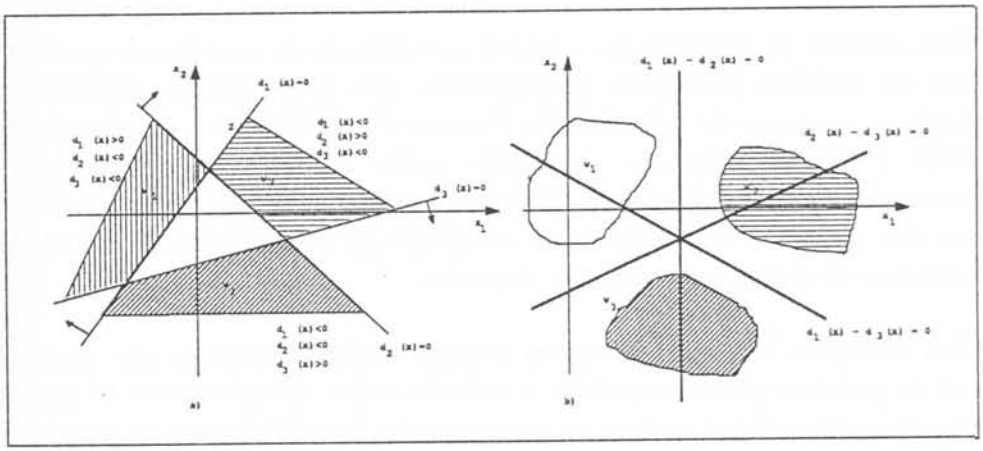


fig. 6.7.Classes de padrões separáveis - a) Cada classe separada por uma única superfície de decisão de todas as outras b) Classes separadas duas a duas por uma única superfície de decisão

Duas superfícies de decisão, permitem a separação no espaço de características, de qualquer uma das classes representadas em 6.7-b), se não houver regiões de indeterminação.

6.2-2 CLASSIFICAÇÃO DE PADRÕES POR MÍNIMA DISTÂNCIA

A utilização de funções distância para classificar padrões, é a maneira mais óbvia e acessivel de calcular um coeficiente de similitude entre vectores de padrões. No espaço Euclidiano essa distância é uma medida de proximidade. Através do exemplo da figura 6.8 observa-se que o ponto x pertence à classe w_i, com base na sua maior proximidade a esta classe.

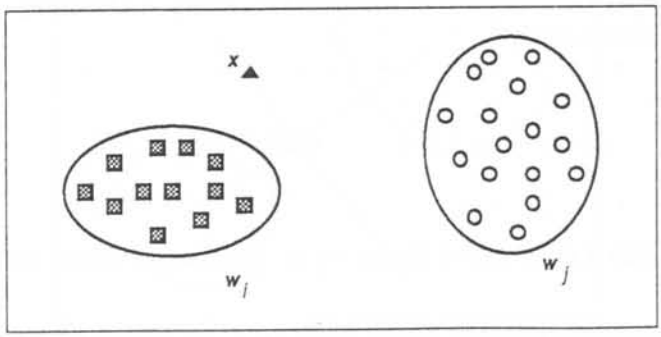


fig. 6.8 Classificação por proximidade

Este método de classificação, só pode ser utilizado de uma forma ajustada, se as classes de padrões possuírem propriedades, que lhe facultem agrupamentos representativos em torno de aglomerados (*"clusters"*). O algoritmo de classificação, que calcula a mínima distância ao padrão médio de cada uma das classes - *classificação de padrões pelo método da mínima distância*[CPDM], será aqui descrito como um dos métodos de classificação utilizados no programa de *localização e reconhecimento de defeitos em madeira de pinho.*

Em situações, em que as classes possam ser representadas por protótipos singulares de padrões, podemos utilizar o método *CPDM*. Considerando M classes e assumindo que estas classes podem ser representadas por protótipos de padrões Z_1, $Z_2, \ldots Z_m$, a distância entre dois padrões arbitrários, representados pelo vector x e pelo protótipo z_i é dada por

$$D_i = \|x - z_i\| = \sqrt{(x - z_i)^T (x - z_i)} \tag{6.18}$$

O classificador *CPMD* calcula a distância do padrão x de classe desconhecida em relação a cada um dos protótipos. Ou seja x é atribuído à classe w_i se $d_i < d_j$ para todo o $i \neq j$. Resolvendo a equação anterior, temos:

$$\begin{aligned} D_i^2 &= \|x - z_i\|^2 = (x - z_i)^T (x - z_i) \\ &= x^T x - 2x^T z_i + z_i{}^T z_i \\ &= x^T x - 2(x^T z_i - \frac{1}{2} z_i{}^T z_i) \end{aligned} \tag{6.19}$$

Escolher o mínimo d_i^2 é equivalente a escolher o mínimo de d_i já que todas as distâncias são positivas. Por outro lado $x^T x$ não depende de i, e escolher o mínimo de (6.19) é equivalente a escolher o máximo de $(x^T z_i - (1/2) z_i^T z_i)$. A função de decisão pode ser definida como:

$$d_i = x^T z_i - \tfrac{1}{2} z^T z \qquad \text{com i=1,2,. . . ,M} \tag{6.20}$$

sendo o padrão x atribuído à classe w_i se $d_i(x) > d_j(x)$ para todo o $i \neq j$

A função de decisão linear é $d_i(x)$, e z_{ij} são as componentes de z_i, com $j=1,2,. . . ,n$.

se fizermos

$$w_{ij} = Z_{ij} \qquad j=1,2, \ldots ,n$$

$$w_{i,n+1} = -\frac{1}{2} Z_i^T Z_i$$

$$X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \\ 1 \end{pmatrix} \tag{6.21}$$

Podemos representar a expressão (6.20) na forma matricial

$$d_i(x) = w^T_i \, x \,, \qquad \text{com } i=1,2, \ldots ,M \tag{6.22}$$

onde

$$w_i = (w_{i1}, w_{i2}, \ldots , w_{i,n+1})^T \tag{6.23}$$

A figura 6.9 mostra um exemplo de separação linear entre duas classes, através de uma recta. As classes são caracterizadas pela circunstância dos protótipos aparecerem em aglomerados únicos (grupos), um de cada lado da fronteira de decisão. Estas situações correspondem aos casos mais fáceis de classificar, mas também as mais teóricas, já que as situações reais são mais complexas.

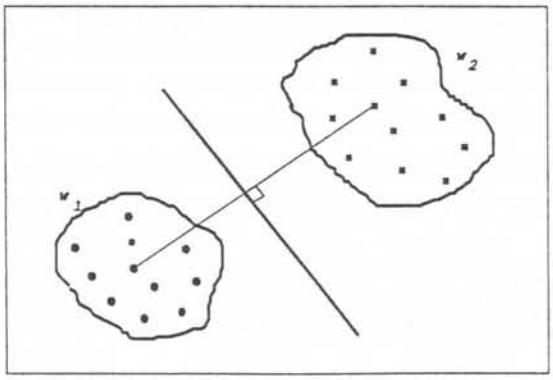


fig. 6.9 Fronteira de decisão entre duas classes, caracterizadas por possuírem protótipos com aglomeração simples.

O diagrama de blocos da fig. 6.10, ilustra a forma como poderia ser implementado um algoritmo de classificação por mínima distância.

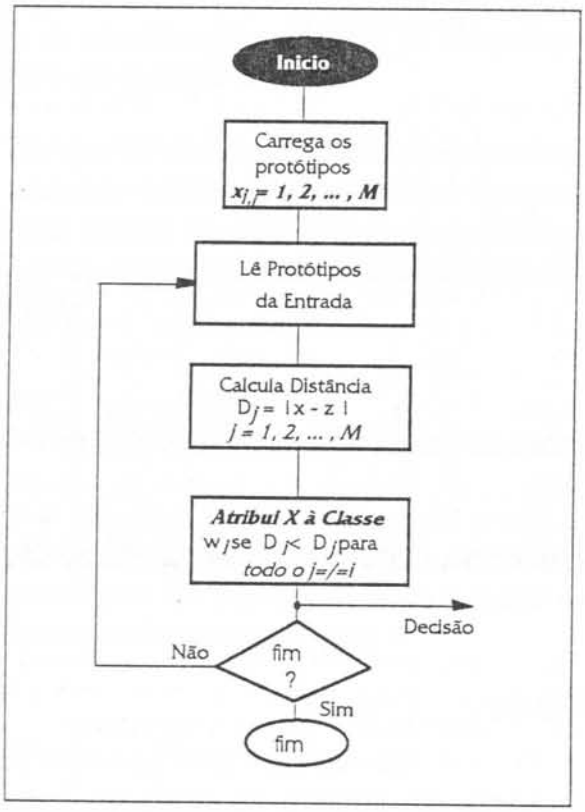


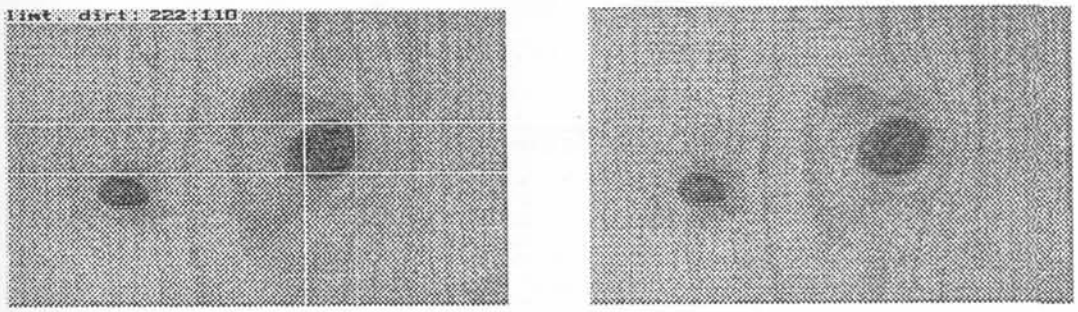
fig. 6.10 Diagrama de blocos dum algoritmo de classificação por mínima distância

6.2-3 UM EXEMPLO DE CLASSIFICAÇÃO POR *CPDM*

Este exemplo retrata um caso de classificação, através do método *CPDM* (*classificação de padrões por mínima distância*). Com esse objectivo o quadro 6-1 mostra alguns dados (padrões) extraídos de pranchas de madeira. Essas características representam um vector aumentado de quatro componentes $(x_1,x_2,x_3,x_4)^T$, relativas aos valores estatísticos extraídos por ***GLDM*** (cap. V, 5.3-3): média às diferenças, contraste às diferenças e 2ºmomento às diferenças, respectivamente.

As medidas foram retiradas de imagens de pranchas de madeira, do tipo indicado na fig. 6.11. As linhas marcadas a branco na imagem a), através de processo interactivo[1], marcam o defeito onde vão ser retiradas as características. Uma função[2] adequada retira por via estatística as componentes do vector de características, $x = (x_1, x_2, x_3, x_4)^T$.

Os protótipos z_1,z_2,z_3 e z_4 representam as componentes de um vector de medidas centrais características, face às quais vai ser medida a distância.



a) b)

fig6-11 Imagens de pranchas de madeira, com um defeito do tipo nó
a)Defeito assinalado na prancha, com objectivos de aprendizagem (extracção de medidas do protótipo)
b) Prancha de madeira, correspondente a a).

Com estes valores z_1,z_2,z_3 e z_4 foram classificados um conjunto de padrões representados no quadro 6-2. Foram utilizadas seis amostras de padrões com classes conhecidas, e para os quais se pretendeu testar o classificador. As classes que foram atribuídas às amostras, estão referenciadas nas quatro colunas à direita da tabela, e os defeitos assinalados a cinzento. Essas classes estão referenciadas por: *SDF* (madeira sem defeitos); *NO* (nós de madeira); *BCH* (brechas de resinagem em madeira); e *BRN* (bolsas de resina). Saliente-se, que segundo o algoritmo apresentado em 6.2.2, a classe é atribuída ao maior valor encontrado para o classificador.

---

[1]O programa que efectua a aprendizagem das características dos defeitos previamente seleccionados, utiliza a função A_spvisd(aprendizagem supervisada), e encontra-se no ficheiro TFUNC.C. Esta função actualiza um ficheiro, possuidor das características do defeito respectivo - RELATRIO.TXT (um ficheiro relatório).

[2]Uma função utilizada na obtenção de características por via estatística, é a função difest1(...), em TFUNC.C, que entre outras medidas retira o valor da média às diferenças, o contraste às diferenças e o 2º momento angular às diferenças (consultar capítulo V).

Amostra/vector	x1(a)	x2(a)	x3(c)	tipo defeito
1	10,38	14,14	4,76	SDF
2	10,07	13,61	4,75	SDF
3	9,86	12,92	4,93	SDF
4	9,7	12,8	4,8	SDF
5	10,62	14,46	4,91	SDF
6	10,45	13,92	5,07	SDF
7	10,92	16,62	4,22	SDF
8	10,84	15,24	4,7	SDF
1	9,54	13,53	5,02	NÓ
2	8,08	10,38	5,68	NÓ
3	8,9	10,52	5,5	NÓ
4	7,96	8,75	5,89	NÓ
5	7,46	7,76	6,24	NÓ
6	7,91	8,77	5,87	NÓ
7	7,73	8,11	6,22	NÓ
8	9,36	11,66	5,34	NÓ
9	7,3	7,55	6,39	NÓ
10	9,51	12,05	5,26	NÓ
11	6,11	5,3	8,19	NÓ
12	8,78	11,68	5,37	NÓ
13	7,13	7,12	6,4	NÓ
14	7,27	7,3	6,46	NÓ
15	10,29	13,79	4,94	NÓ
16	8,41	9,52	5,7	NÓ
17	8,79	9,96	6	NÓ
18	9,98	12,9	5,11	NÓ
19	10,13	13,2	5,11	NÓ
20	8,31	9,28	5,78	NÓ
1	15,87	34,54	2,94	BCH
2	15,87	34,54	2,94	BCH
3	15,71	34,32	2,95	BCH
4	12,58	23,61	3,62	BCH
5	12,4	22,54	3,68	BCH
6	16,56	37,84	2,82	BCH
7	10,29	13,79	4,94	BCH
8	11,79	21,48	4,04	BCH
1	17,13	41,97	2,75	BRN
2	17,53	43,89	2,75	BRN
3	16,01	33,32	3,22	BRN
4	17,39	42,9	2,81	BRN
5	15,27	30,93	3,3	BRN
6	16,4	42,09	2,76	BRN
7	18,94	55,71	2,53	BRN
8	23,46	83,26	2	BRN
9	19,09	51,04	2,54	BRN
10	24,08	81,99	2,16	BRN
11	17,67	51,4	2,9	BRN
12	19,99	61,16	2,49	BRN
13	20,43	68,22	2,45	BRN
14	17,51	47,85	2,63	BRN
15	16,03	42,09	2,87	BRN
16	16,77	44,04	2,79	BRN
Z1	10,36	14,21	4,77	Protótipo 1
Z2	8,45	9,96	5,82	Protótipo 2
Z3	13,88	27,83	3,49	Protótipo 3
Z4	18,36	51,37	2,68	Protótipo 4

quadro 6.1-Amostra dos valores das características x_1,x_2,x_3 e x_4, a partir das quais se definem as medidas centrais dos protótipos z_1,z_2,z_3 e z_4. Medidas extraídas por processo de ensaio em pranchas de madeira

x	x1	x2	x3	x4	d1=w1^.x	d2=w2^.x	d3=w3^.x	d4=w4^.x
1	10,38	14,14	4,76	1	165,17	153,74	64,51	-492,58
2	9,92	14,39	4,64	1	163,39	151,64	64,67	-475,47
3	12,41	22,54	3,68	1	300,41	248,27	322,68	90,26
4	17,67	51,4	2,91	1	761,34	575,68	1196,19	2100,60
5	8,79	10,38	5,68	1	99,66	108,21	-58,99	-752,30
6	17,51	47,85	2,63	1	707,90	537,34	1094,19	1852,31
•••					SDF	NÓ	BCH	BRN

quadro 6-2 Tabela de exemplificação de classificação de defeitos por *CPDM*

Os gráficos das fig. 6.12, fig. 6.13 e fig. 6.14 representam respectivamente, o valor central das característica x_1, x_2 e x_3 por classe, o valor por característica x_1, x_2 e x_3 em cada uma das classes, e uma representação mapeada da distribuição espacial das características, mostrando a sua dispersão.

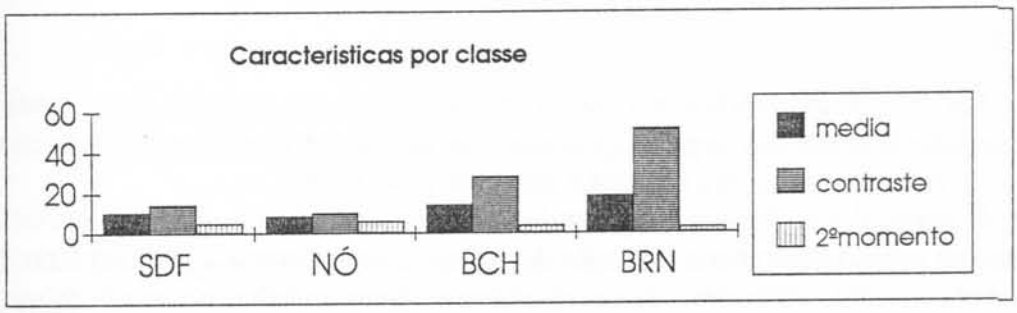


fig. 6. 12 Valor das características x_1, x_2 e x_3 por classe de defeito

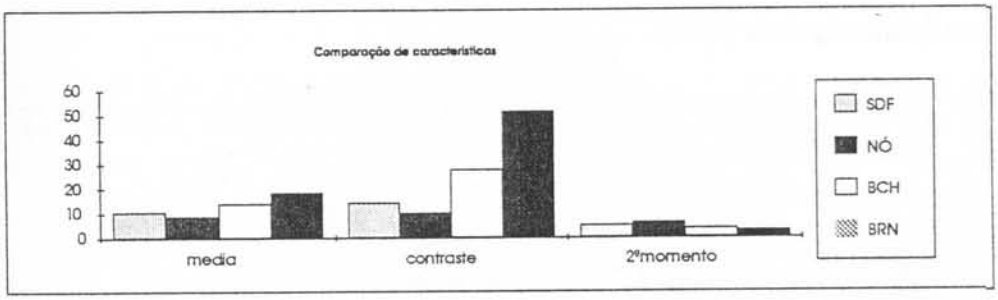


fig. 6.13 Valor por característica para as diferentes classes

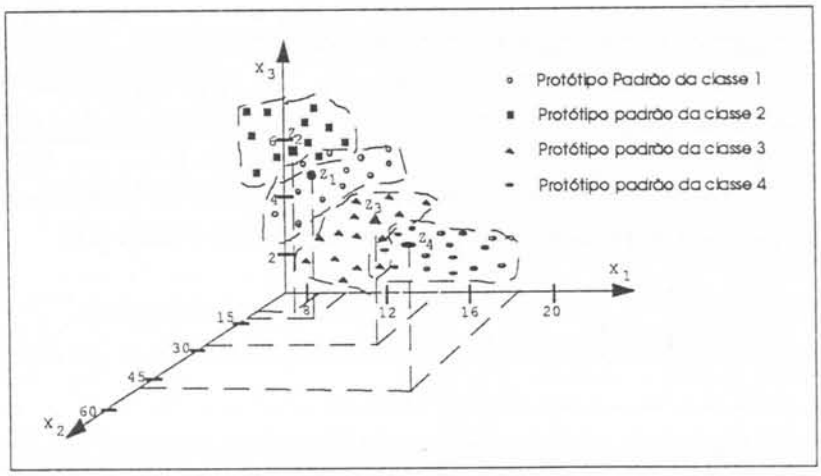


fig. 6.14 Dispersão de características por classe

6.3 Funções de decisão obtidas com aprendizagem. Técnicas não paramétricas

6.3-1 Generalidades

Um vector de padrões x é um ponto no espaço das padrões. Esse espaço pode ser dividido em sub espaços, aos quais se atribui um conjunto de protótipos padrão. A cada um desses sub espaços é atribuida uma classe.

É possível através dum conjunto de protótipos de padrões, definidos em cada um dos sub espaços, encontrar a função de decisão correspondente. O hiperplano de decisão encontrado, pode servir como classificador duma amostra, nesse sub espaço. As hiper superfícies de decisão encontradas, podem ser lineares ou não lineares. Só iremos abordar aqui as funções de decisão lineares.

Posso escrever duma forma sintética, uma função de decisão, na forma seguinte (6.24)

$$d(x)=f(w,x) \tag{6.24}$$

com

$$x = (x_1,x_2, \ldots, x_n,1)^T \quad \text{e} \quad w = (w_1,w_2, \ldots, w_n, w_{n+1})^T$$

em que x representa o vector de características dum padrão e w o vector pesado da função de decisão.

Num problema de classificação, o treino ou aprendizagem, visa encontrar um conjunto de vectores w (com componentes $w_1, w_2, \ldots, w_n, w_{n+1}$) que constituam as funções de decisão, que permitam dividir o espaço Euclidiano em n+1 dimensões.

Para cada protótipo z_k^m, com $k = 1,2, \ldots, M$ e $m = 1,2, \ldots, N_k$ em que M representa o número de classes e N_k representa o número de protótipos em cada classe k. No espaço w a expressão do hiperplano de decisão é dado por:

$$w^T z_k^m = 0 \tag{6.25}$$

Qualquer vector w no lado positivo do hiperplano, torna $w^T z > 0$ e no lado negativo $w^T z < 0$.

Para o caso de um problema de classificação com duas classes, supondo haver N_1 padrões pertencentes à classe w_1 e N_2 padrões pertencentes à classe w_2, supondo ainda que w_1 e w_2 são separáveis, o vector w deve ser procurado de tal modo que $d_1(w,z) > 0$,de tal modo que:

$$w^T z_1^m > 0 \qquad \forall z_1^m \in w_1,\ m = 1,2,\ldots,N_1 \tag{6.26}$$

$$w^T z_2^m < 0 \qquad \forall z_2^m \in w_2,\ m = 1,2,\ldots,N_2 \tag{6.27}$$

onde z_1^m e z_2^m representam protótipos em w_1 e w_2, respectivamente.

Com N protótipos, podemos definir N hiperplanos. Supondo três protótipos z^1, z^2, z^3 , todos pertencentes à classe w_1, esses três hiperplanos podem ser representados no espaço de padrões, como se mostra na fig. 6.15, e nesse caso qualquer w teria de classificar os três protótipos em w_1, de acordo com (6.28)

$$d_1(w,z) > 0 \qquad d_2(w,z) > 0 \qquad d_3(w,z) > 0 \tag{6.28}$$

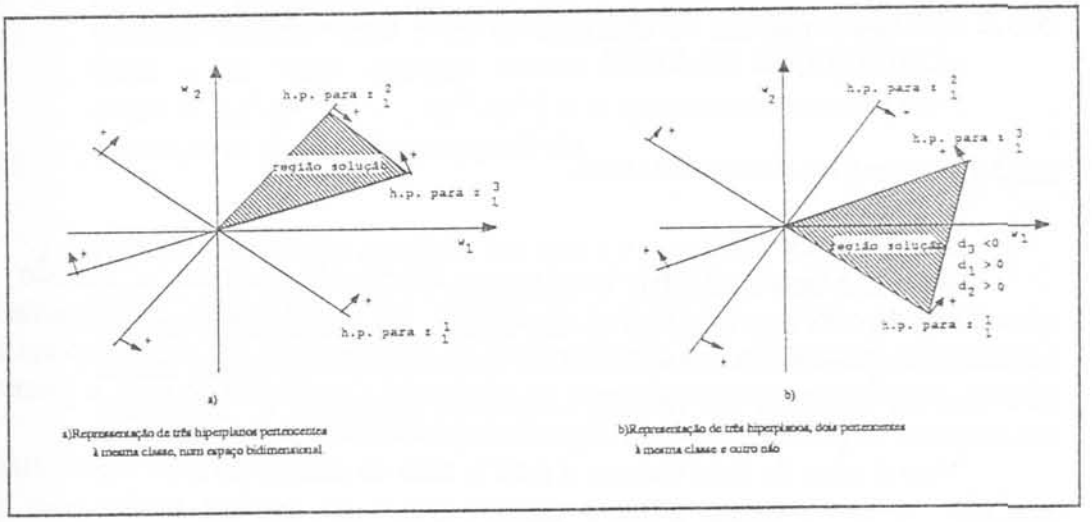


fig. 6.15 Representação de hiperplanos num problema de duas classes

Se tiver três padrões z^1, z^2, z^3, com z^1, z^2 pertencentes à classe w_1 e z^3 pertencente à classe w_3 teremos nesta situação

$$d_1(w,z) > 0 \qquad d_2(w,z) > 0 \qquad d_3(w,z) < 0 \tag{6.29}$$

Para o caso dum problema de classificação com duas classes, $d(w,x)$ deve tomar um valor positivo, para padrões pertencentes a uma das classes, e menores que zero para padrões pertencentes à outra das classe. A solução pode ser generalizada às M classes se os padrões pertencentes às classes restantes, forem substituídos por valores negativos. Para duas classes seria,

$$w^T z > 0 \qquad \forall z = z_1^m , - z_2^m \tag{6.30}$$

Se pretender dispor de uma margem de segurança, escolho um intervalo de segurança *("threshold")* T maior que zero, que satisfaça a expressão seguinte:

$$w^T z > T \qquad \forall z = z_1^m , - z_2^m \tag{6.31}$$

No processo de aprendizagem, qualquer w que satisfizer a inequação será solução do problema.

6.3.2 Procedimento de aprendizagem através da correcção do erro

6.3.2-1 Regras de Aprendizagem .

Tou e Gonzalez[32-A] bem como Bow[9-A], abordam a questão da aprendizagem e treino das funções de decisão, em problemas de classificação de natureza não paramétrica. Os procedimentos de aprendizagem dos classificadores são obtidos através de processos diversos de adaptação e correcção do erro, a partir de um conjunto de protótipos padrão disponíveis para cada uma das classes w.

Para o caso de duas classes, e para o caso de dois protótipos representados por z_1^m e z_2^m , vamos testar o classificador w do seguinte modo: caso seja encontrado um dos protótipos mal classificado, deslocar um dos hiperplanos no sentido de classificar as duas amostras correctamente. A função de decisão, há-de verificar as condições expressas em (6.32):

$$w^T z_1^m < 0 \quad \text{e} \quad w^T z_2^m > 0 \tag{6.32}$$

A forma mais simples de efectuar a correcção, é deslocar o hiperplano no sentido de classificar as duas amostras correctamente, ou seja consiste em deslocar o w na direcção perpendicular ao hiperplano. A condição $w^T z = 0$ implica que z seja

perpendicular ao hiperplano, e portanto satisfazendo a condição anterior. Isso corresponde a deslocar o hiperplano na direcção z_1^m ou na direcção $-z_2^m$. A correcção de w é feita por:

$$w(k+1) = w(k) + cz_1^m \quad \text{se} \quad w^T(k)z_1^m < 0(T) \tag{6.33}$$

$$w(k+1) = w(k) - cz_2^m \quad \text{se} \quad w^T(k)z_2^m < 0(-T) \tag{6.34}$$

$$w(k+1) = w(k) \quad \text{se} \quad \text{classificação for correcta}$$

O processo de treino é efectuado para todos os padrões $N=N_1+N_2$ que vão sendo integrados no cálculo, um de cada vez de uma forma iterativa. O processo é repetido até que todos os padrões tenham obtido classificação correcta. A correcção por seu turno é efectuada sempre que acontece uma classificação incorrecta, e depende do valor escolhido para 'c'.

O valor de c pode ser escolhido, utilizando um de três critérios possíveis:

1 - ***Regra do incremento fixo*** em que c é uma constante positiva qualquer (muitas vezes c=1). O algoritmo é iniciado com $w(0)$ igual a um valor qualquer, muitas vezes zero. A sequência de treino é $[z_1^1, z_2^1, \ldots, z_1^{N1}, z_2^{N1}]$ e o ajustamento termina em k etapas, se as classes forem separáveis.

2 - ***Regra de correcção absoluta***, em que c é escolhido como o menor inteiro, que faça $w(k+1)$ ultrapassar o hiperplano dos protótipos padrão mal classificados, sempre que na iteração, uma classificação errada aconteça. Considerando $\mathbf{z}^*$ a média dos vectores que não satisfazem a inequação $wz \geq 0(T)$, c deverá ser escolhido de tal modo que :

$$w^T(k+1)z_i^* = [\, w(k) + cz_i^*]' \, z_i^* > 0(T) \tag{6.35}$$

e

$$cz_i^{*T} z_i^* > T - {}^T(k)\, z_i^* \tag{6.36}$$

resultando

$$c > \frac{T - w^T(k)z_i^*}{z_i^{*T} z_i^*} \tag{6.37}$$

e se T=0, fica

$$c > \frac{\left| w^{T}(k) z_i^{*T} \right|}{\left| z_i^{*} \right|^2} \tag{6.38}$$

prova-se que o vector solução w, é encontrado em k iterações, para o caso das classes serem separáveis.

3 - ***A regra de correcção fraccional*** consiste em: partindo do pressuposto que z é perpendicular ao hiperplano de decisão e que aponta no sentido positivo, pela fig. 6.16 observa-se que a distância ao hiperplano é dada por:

$$D = \Delta + \delta = \frac{T}{\left| z_i^{*} \right|} + \left| w(k) \frac{z_i^{*}}{\left| z_i^{*} \right|} \right| \tag{6.39}$$

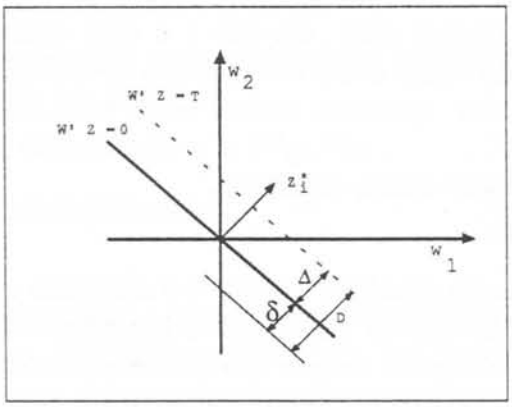


fig. 6.16 Hiperplano de decisão, correcção da sua posição na direcção de z_i^*

quando $w(k)$ está do outro lado do hiperplano D, apresenta o valor seguinte:

$$D = \frac{T - \left| w^{T}(k) z_i^{*} \right|}{\left| z_i^{*} \right|} \tag{6.40}$$

na correcção fraccional c é escolhido de tal modo que w é movido de uma fracção de distância, na direcção perpendicular ao hiperplano desejado;

$$c = \lambda \frac{D}{z_i^*} \qquad \text{com} \quad \lambda > 0 \tag{6.41}$$

$$w(k+1) - w(k) = \lambda D \frac{z_i^*}{|z_i^*|} \tag{6.42}$$

com $T(limiar)=0$

$$c = \lambda \frac{|w^T(k) z_i^*|}{|z_i^*|^2} \quad \text{com} \quad \lambda > 0 \tag{6.43}$$

prova-se que para $0>\lambda>2$, a solução para encontrar w correcto, através de aprendizagem por processo de correcção fraccional, converge num número de iterações limitado.

6.3.2-2 ALGORITMO DE APRENDIZAGEM

O algoritmo de aprendizagem pode ser descrito pelos passos seguintes: [Foi assumido inicialmente que M=2].

1. Para cada amostra **z** do conjunto de treino, verificar a categoria a que z pertence.

2. Se uma resposta correcta for obtida, passar à próxima amostra **z.**

3. Se a resposta for incorrecta, achar um novo w(k+1) à custa de w(k)

4. Depois de todos os **z**'s terem sido examinados, repetir o processo pela mesma ordem.

Se os todos z's forem linearmente separáveis, qualquer dos três algoritmos estudados (*c* escolhido de acordo com 6.3-2) convergirá para um *w* correcto num número limitado de iterações.

Para um número de classes maior que dois ($M>2$) o processo de cálculo é semelhante. Assumindo o conjunto de treino válido para todas as classes de padrões w_i, com $i=1,2,\ldots,M$ se:

se $d_i(z) > d_j(z)$ *e* $z \in w_i \quad \forall \quad j \neq i \quad \therefore$ os vectores *w* não são ajustados

se $d_j(z) > d_i(z)$ *e* $z \in w_i \quad \forall \quad j \neq i \quad \therefore$ os ajustamentos serão feitos do seguinte modo:

$w_i(k+1) = w_i(k) + cz$ para o *w* a que pertence o padrão

$w_j(k+1) = w_j(k) - cz$ para as classes em que se verifica $d_j(z) > d_i(z)$

$w_l(k+1) = w_l(k)$ para as classes em que se verifica $d_i(z) > d_j(z)$

para as iterações *k* e *k+1* respectivamente.

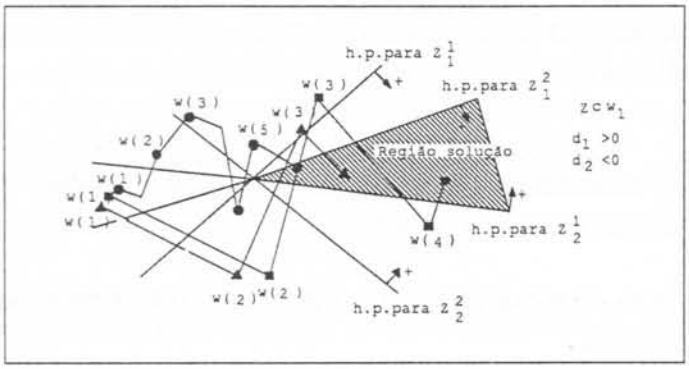


fig. 6.17 Procedimentos de ajuste num algoritmo de:

- INCREMENTO FIXO
- CORRECÇÃO ABSOLUTA
- CORRECÇÃO FRACCIONAL

6.3.3-TÉCNICAS BASEADAS EM GRADIENTE

6.3.3-1 INTRODUÇÃO

As técnicas baseadas em gradientes, descritas por Sing-Tze Bow[9], são outro processo utilizado para treinar um sistema de amostras, através de um processo de aproximação gradual à função de decisão correcta. O vector gradiente tem a particularidade de apontar na direcção da máxima variação de crescimento de w, quando o argumento aumenta. O ajustamento pode ser formulado do seguinte modo:

$$w(k+1) = w(k) - \rho_k \nabla j(w)|_{w=w(k)} \tag{6.44}$$

Em que $j(w)$ é uma função que estabelece um critério, que deve ser minimizada por ajuste de w. A aproximação pode ser feita movendo w na direcção negativa à do gradiente. Pode ser sumarizada do seguinte modo:

1 - Começar com a escolha arbitrária do vector w(1) e calcular o vector gradiente$\nabla\ j[w(1)]$

2 - Obter o novo valor $w(2)$ movendo de uma certa quantidade $w(1)$ na direcção descendente.

O factor ρ_k é um número positivo que marca a fracção de aproximação. $J(w)$ pode ser aproximado do seguinte modo:

$$J[w(k+1)] = J[w(k)] + [w(k+1) - w(k)]' \nabla\ j[w(k)] + 1/2[w(k+1) - w(k)]' . D\ [w(k+1) - w(k)] \tag{6.45}$$

onde

$$D = \left.\frac{\partial^2 J}{\partial w_i \partial w_j}\right|_{w = w_k} . \tag{6.46}$$

substituindo (6.44) em (6.45) obtém-se:

$$J[w(k+1)] = J[w(k)] - \rho_k / \nabla J[w(k)]/^2 + 1/2\rho_k^2 \nabla J' D. \nabla J \quad (6.47)$$

fazendo $\partial J[w(k+1)]/\partial \rho_k = 0$ para obter o mínimo de $J[w(k+1)]$, resulta:

$$/\nabla J/^2 = \rho_k \nabla J' D \nabla J \quad (6.48)$$

ou

$$\rho_k = \left. \frac{|\nabla J|^2}{\nabla J' D \nabla J} \right|_{w = w(k)} \quad (6.49)$$

que é equivalente a fazer $\rho_k = D^{-1}$

Dado que D^{-1} pode não existir, e se existir pode ser complexo e moroso o seu cálculo, geralmente toma-se ρ_k por uma constante qualquer positiva.

6.3.3-2 Determinação da função critério *"Perceptron"*

A função critério $J_p(w)$, pode ser escolhida como sendo definida por

$$J_p(w) = \sum_{z \in P} -w^T z \quad (6.50)$$

onde o somatório envolve as amostras mal classificadas. $J_p(w)$ é proporcional à soma das distâncias dos padrões mal classificados ao hiperplano de decisão. Tomando a derivada de $J_p(w)$ em função de $w(k)$, obtém-se:

$$\nabla J_p.[w(k)] = -\sum_{z \in P} z \quad (6.51)$$

$w(k)$ é o valor de w na iteração k

O algoritmo de aproximação (*"perceptron"*) seria formulado do seguinte modo:

$$w(k+1) = w(k) - \rho \; {}_k\nabla \; j_p(w) \qquad (6.52)$$

ou

$$w(k+1) = w(k) + \rho \; k \sum_{z \in P} z \qquad (6.53)$$

Em que P é o conjunto de amostras mal classificadas por $w(k)$. Determina-se o produto $w^T z$ para todos os padrões $z \in P$ e ajusta-se somente depois de ter classificado todos os padrões.

6.3-4 EXEMPLO DE CLASSIFICAÇÃO, ATRAVÉS DE HIPERPLANOS CALCULADOS PELO ALGORITMO DE CORRECÇÃO DO ERRO

No exemplo seguinte procura encontrar-se através de um conjunto de amostras pertencentes a três classes de padrões separáveis, os hiperplanos de decisão capazes de as separar. Os padrões estão representados no quadro 6.3 por vectores de características de dimensão três aumentados.

Amostra/vector	Z1 a)	Z2 b)	Z3 c)	Z4	tipo defeito
1	10,38	14,14	4,76	1,00	SDF
2	10,07	13,61	4,75	1,00	SDF
3	9,86	12,92	4,93	1,00	SDF
4	9,70	12,80	4,80	1,00	SDF
5	10,62	14,46	4,91	1,00	SDF
6	10,45	13,92	5,07	1,00	SDF
7	10,84	15,24	4,70	1,00	SDF
1	15,03	32,38	3,00	1,00	CRD
2	16,15	34,83	2,99	1,00	CRD
3	15,90	33,89	3,02	1,00	CRD
4	15,88	36,36	2,82	1,00	CRD
5	15,82	33,41	3,05	1,00	CRD
6	16,04	37,89	2,76	1,00	CRD
7	15,47	33,43	2,97	1,00	CRD
8	18,86	45,34	2,85	1,00	CRD
1	11,90	19,30	3,90	1,00	C_Z
2	14,12	25,55	3,70	1,00	C_Z
3	10,73	16,33	4,20	1,00	C_Z
4	11,45	18,41	3,98	1,00	C_Z

Quadro 6.3 Protótipos de classificação de amostras de madeira
a) Média às diferenças b) Contraste às diferenças
c) 2º momento angular às diferenças (Método GLDM)
Classes: *SDF* - madeira sem defeito *CRD* - cardido *C_Z* - Azulado da madeira

As três funções de decisão w_K, com $k=1,\ldots,M$ e M igual ao número de classes, neste caso igual a três, foram determinadas pelo processo de aproximação por correcção do erro. Pelo facto de as classes serem claramente separáveis, a convergência do algoritmo deu-se em apenas cinco iterações. O exemplo é meramente académico. Foi escolhido com uma dimensão reduzida, de modo a que os dados pudessem ser representados com maior facilidade.

O quadro 6.5 representa um resumo simplificado, de alguns valores intermédios assumidos pelos vectores de classificação w_1, w_2 e w_3 na implementação do algoritmo descrito em 6.3.2-2. Saliente-se que em cada iteração, todas as amostras são passadas pelo classificador, seguindo o algoritmo de aprendizagem[3] descrito em **6.3.2-2**.

[illegible]	[illegible]	x1	x2	x3	w1(0)	w1(1)	w1(2)	w1(3)	w2(0)	w2(1)	w2(2)	w2(3)	w3(0)	w3(1)	w3(2)	w3(3)
0	0	10,38	14,14	4,76	10,38	14,14	4,76	1,00	-10,38	-14,14	-4,76	-1,00	-10,38	-14,14	-4,76	-1,00
0	1	14,16	28,12	3,20	-3,78	-13,98	1,56	0,00	3,78	13,98	-1,56	0,00	-24,54	-42,26	-7,96	-2,00
0	2	11,90	19,30	3,90	-15,68	-33,28	-2,34	-1,00	-8,12	-5,32	-5,46	-1,00	-12,64	-22,96	-4,06	-1,00
0	0	9,86	12,92	4,93	-10,78	-39,13	4,34	0,00	-27,14	-25,02	-15,84	-3,00	-18,45	-23,94	-10,04	-2,00
0	1	16,15	34,83	2,99	-10,78	-39,13	4,34	0,00	-10,99	9,81	-12,85	-2,00	-34,60	-58,77	-13,03	-3,00
0	2	10,73	16,33	4,20	-21,51	-55,46	0,14	-1,00	-21,72	-6,52	-17,05	-3,00	-23,87	-42,44	-8,83	-2,00
1	0	10,38	14,14	4,76	-12,67	-73,08	14,53	1,00	-42,84	-11,26	-34,49	-6,00	-43,19	-65,21	-19,63	-4,00
1	1	14,16	28,12	3,20	-12,67	-73,08	14,53	1,00	-42,84	-11,26	-34,49	-6,00	-43,19	-65,21	-19,63	-4,00
1	2	11,90	19,30	3,90	-24,57	-92,38	10,63	0,00	-54,74	-30,56	-38,39	-7,00	-31,29	-45,91	-15,73	-3,00
1	0	9,86	12,92	4,93	-18,76	-91,40	16,61	1,00	-74,67	-57,09	-48,07	-9,00	-37,10	-46,89	-21,71	-4,00
1	1	16,15	34,83	2,99	-18,76	-91,40	16,61	1,00	-58,52	-22,26	-45,08	-8,00	-53,25	-81,72	-24,70	-5,00
1	2	10,73	16,33	4,20	-29,49	-107,73	12,41	0,00	-69,25	-38,59	-49,28	-9,00	-42,52	-65,39	-20,50	-4,00
2	0	10,38	14,14	4,76	-20,65	-125,35	26,80	2,00	-90,37	-43,33	-66,72	-12,00	-67,27	-110,60	-29,05	-6,00
2	1	14,16	28,12	3,20	-20,65	-125,35	26,80	2,00	-90,37	-43,33	-66,72	-12,00	-67,27	-110,60	-29,05	-6,00
2	2	11,90	19,30	3,90	-32,55	-144,65	22,90	1,00	-102,27	-62,63	-70,62	-13,00	-55,37	-91,30	-25,15	-5,00
2	0	9,86	12,92	4,93	-26,74	-143,67	28,88	2,00	-116,39	-88,18	-74,32	-14,00	-61,18	-92,28	-31,13	-6,00
2	1	16,15	34,83	2,99	-26,74	-143,67	28,88	2,00	-100,24	-53,35	-71,33	-13,00	-77,33	-127,11	-34,12	-7,00
2	2	10,73	16,33	4,20	-37,47	-160,00	24,68	1,00	-110,97	-69,68	-75,53	-14,00	-66,60	-110,78	-29,92	-6,00
3	0	10,38	14,14	4,76	-28,60	-151,15	30,41	2,00	-122,42	-88,09	-79,51	-15,00	-75,47	-119,63	-35,65	-7,00
3	1	14,16	28,12	3,20	-28,60	-151,15	30,41	2,00	-122,42	-88,09	-79,51	-15,00	-75,47	-119,63	-35,65	-7,00
3	2	11,90	19,30	3,90	-40,50	-170,45	26,51	1,00	-122,42	-88,09	-79,51	-15,00	-63,57	-100,33	-31,75	-6,00
3	0	9,86	12,92	4,93	-34,69	-169,47	32,49	2,00	-131,37	-94,18	-85,14	-16,00	-84,41	-133,69	-40,73	-8,00
3	1	16,15	34,83	2,99	-34,69	-169,47	32,49	2,00	-131,37	-94,18	-85,14	-16,00	-84,41	-133,69	-40,73	-8,00
3	2	10,73	16,33	4,20	-45,42	-185,80	28,29	1,00	-131,37	-94,18	-85,14	-16,00	-73,68	-117,36	-36,53	-7,00
4	0	10,38	14,14	4,76	-36,55	-176,95	34,02	2,00	-131,37	-94,18	-85,14	-16,00	-82,55	-126,21	-42,26	-8,00
4	1	14,16	28,12	3,20	-36,55	-176,95	34,02	2,00	-131,37	-94,18	-85,14	-16,00	-82,55	-126,21	-42,26	-8,00
4	2	11,90	19,30	3,90	-36,55	-176,95	34,02	2,00	-131,37	-94,18	-85,14	-16,00	-82,55	-126,21	-42,26	-8,00
4	0	9,86	12,92	4,93	-36,55	-176,95	34,02	2,00	-131,37	-94,18	-85,14	-16,00	-82,55	-126,21	-42,26	-8,00
4	1	16,15	34,83	2,99	-36,55	-176,95	34,02	2,00	-131,37	-94,18	-85,14	-16,00	-82,55	-126,21	-42,26	-8,00
4	2	10,73	16,33	4,20	-36,55	-176,95	34,02	2,00	-131,37	-94,18	-85,14	-16,00	-82,55	-126,21	-42,26	-8,00

quadro 6.4 Tabela com alguns valores intermédios, dos vectores de decisão w_1, w_2, w_3 sujeitos a processo de correcção pelo algoritmo descrito em 6.3.2-2

O quadro 6.5 apresenta uma amostra de classificação, para seis casos diferentes, através dos hiperplanos de decisão encontrados no algoritmo descrito anteriormente. A amostra é atribuída à classe que obtiver o valor mais elevado de classificação.

[3] A função Percetrn2( ) em TRCPAD.C (arquivo de Reconhecimento de padrões) executa o algoritmo , submetendo de cada vez, um protótipo de cada uma das classes alternadamente. Percetrn1( ) executa o mesmo algoritmo, mas submetendo primeiro ao teste todos os protótipos da 1ª classe, depois da todos os da 2ª classe, etc. O ficheiro FICH1.TXT recebe todos os dados intermédios de actualização dos w_i e PCTRTS.TXT recebe os vectores de decisão w_i finais.

w	w1	w2	w3	w4			
1	-36,55	-176,95	34,02	2,00			
2	-131,37	-94,18	-85,14	-16,00			
3	-82,55	-126,21	-42,26	-8,00			
x	x1	x2	x3	x4	d1=w1'x	d2=w2'x	d3=w3'x
1-C_Z	11,61	19,29	4,66	1,00	-3677,18	-3754,69	-3597,93
2-SDF	10,21	14,01	4,69	1,00	-2690,69	-3076,06	-2817,24
3-CRD	18,86	45,34	2,85	1,00	-8613,29	-7006,41	-7407,70
4-C_Z	10,73	16,30	4,20	1,00	-3131,58	-3318,32	-3128,48
5-CRD	12,09	26,12	3,83	1,00	-4931,53	-4390,33	-4464,49
6-SDF	10,51	15,32	4,46	1,00	-2941,29	-3219,26	-2997,62
***	.....	.....	.....	.....	SDF	CRD	C_Z

quadro 6.5 Classificação de seis amostras através dos hiperplanos de decisão w_1, w_2, w_3 obtidos pelo método de aproximações sucessivas

6.3-5 CONSIDERAÇÕES FINAIS SOBRE ALGORITMOS DE APRENDIZAGEM POR CORRECÇÃO DO ERRO

Os algoritmos apresentados determinam o peso dos vectores w, por processos de aprendizagem através de regras de correcção. Não permitem um erro de classificação sem fazerem um ajuste à função discriminante w.

Para o caso de duas classes com uma distribuição normal, por exemplo, ocorrerá sempre um erro, mesmo no caso de ter sido encontrado o hiperplano de decisão óptimo. Nesta situação a regra de correcção apresentará sempre um erro e w nunca estabilizará numa posição óptima. Nestes casos de sobreposição parcial de classes, a oscilação de w provoca problemas de indeterminação, e uma regra de paragem pode ter de ser definida. Esta regra tem de ser encontrada com muito cuidado para que o w obtido não se torne pouco consistente. A solução consistirá em utilizar métodos que não sejam baseados em técnicas de correcção do erro, como são as que utilizam processos de aproximação estocástica, função potencial ou procedimentos de minimização do erro médio quadrático. Para o caso de classes, que se distribuam por vários grupos no espaço de padrões, também se levantam problemas de classificação através destes processos.

Com este tipo de algoritmos de aprendizagem, os classificadores garantem bons resultados, num número de iterações razoável, na condição das classes serem linearmente separáveis.

6-4 Procedimentos através da Minimização do Erro Médio Quadrático (LMSE)

6.4-1 Método da Matriz Pseudo Inversa

Num algoritmo de aproximação por correcção do erro, o hiperplano de decisão só podia ser encontrada se as classes fossem separáveis. Em caso contrário, o algoritmo oscilava e não era possível prever, nem o número de iterações necessários para completar o algoritmo, nem a qualidade da solução final encontrada.

O método da matriz pseudo - inversa tem a vantagem de apresentar sempre uma solução, para casos de classes separáveis, e de fornecer também uma indicação de não separabilidade, nos casos em que essa situação aconteça. A única desvantagem deste algoritmo *LMSE*, reside no facto de terem que se efectuar operações com matrizes, que podem ser de elevada dimensão, principalmente a inversão da matriz $(x^T.x)^{-1}$, o que pode acarretar custos computacionais elevados.

O que se pretende com este algoritmo é resolver a equação $Z.w = b$, em vez de tentar resolver a inequação $Zw > 0$.

Isso significa ter de resolver um sistema de N equações ($N = \sum_{i=1}^{M} N_i$) lineares, onde N é o número total de protótipos, N_i é o número total de protótipos na classe w_i e M é o número de classes.

Z e b podem ser definidos respectivamente como,

$$Z = \begin{bmatrix} z_1^T \\ z_2^T \\ \vdots \\ z_n^T \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1n} \\ z_{21} & z_{22} & \cdots & z_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ z_{N1} & z_{N2} & \cdots & z_{Nn} \end{bmatrix} \tag{6.54}$$

e

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \tag{6.55}$$

Se Z for uma matriz não singular e quadrada, posso escrever que $w = Z^{-1}b$, e resolvê-la em ordem a w. Como Z geralmente é rectangular, mais linhas que colunas, o sistema é indeterminado e apresenta por isso várias soluções para a equação $Zw=b$. Definindo o vector erro $e=Zw - b$. Assumindo a função $J_S(w)$ como a soma dos erros médios quadráticos

$$J_s(w)=\frac{1}{2}|e|^2=\frac{1}{2}|zw-b|^2=\frac{1}{2}\sum_{i=1}^{N}(z_i w-b_i)^2 \qquad (6.56)$$

Tomando a derivada parcial de J_S em função de w, obtenho

$$\nabla J_s(w)=\sum_{i=1}^{N}(z_i w-b_i)z_i \qquad (6.57)$$

Na forma matricial

$$\nabla J_s(w)=\mathbf{Z}^T(\mathbf{Z}w-\mathbf{b})$$

fazendo $\nabla J_s(w)=0$, minimizando a expressão (6.58)

$$\mathbf{Z}^T w\mathbf{Z}=\mathbf{Z}^T\mathbf{b}$$

ou

$$w=\mathbf{Z}^{\#}\mathbf{b} \qquad (6.59)$$

onde $\mathbf{Z}^{\#}=(\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'$, chamada matriz pseudo inversa de Z. $\mathbf{Z}^{\#}$ tem a propriedade de ser igual a $\mathbf{Z}^{-1}$ se Z for quadrada e não singular. O valor de **b** pode ser escolhido arbitrariamente, na condição de b_i ser maior que zero para $\forall_i$. Uma possível escolha será $\mathbf{b}=[1\ 1\ 1\ \ldots\ 1]=\mathbf{u}^T$.

Esta técnica é diferente da anterior na medida que não calcula um w por cada z apresentado, através de mecanismos de aproximação, mas sim através da consideração de todas as amostras em conjunto. Por esse motivo o tempo de treino pode ser muito menor do que o dispendido com os algoritmos de correcção do erro mostrados anteriormente.

6.4-2 MÉTODO DE HO-KASHYAP

Quando $J(w)$ é considerada mínima não só em relação a w, mas também em relação a **b**, o algoritmo de aprendizagem toma o nome de Ho-Kashyap.

a função $J(w) = 1/2 \mid \mathbf{Z}w - \mathbf{b} \mid^2$ (6.60)

e as derivadas parciais de $J(w)$ em relação a w e **b** são respectivamente

$$\frac{\partial J(w)}{\partial w} = \mathbf{Z}^T(\mathbf{Z}w - \mathbf{b})$$

e (6.61)

$$\frac{\partial J(w)}{\partial b} = -(\mathbf{Z}w - \mathbf{b})$$

fazendo $\partial J / \partial w = 0$, ficamos com

$$w = (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{b} = \mathbf{Z}^{\#} \mathbf{b} \quad (6.62)$$

como **b** é positivo, o ajustamento de **b** pode ser feita de tal modo que

$$\mathbf{b}(k+1) = \mathbf{b}(k) + \delta \mathbf{b}(k) \quad (6.63)$$

$$\delta b_i(k) = \begin{cases} 2c[e(k)] & \text{quando } e(k) > 0 \\ 0 & \text{quando } e(k) \leq 0 \end{cases} \quad (6.64)$$

representando k, i, e c respectivamente o índice de interacção, o índice do vector e o incremento positivo de correcção

$$w(k+1) = \mathbf{Z}^{\#}\,\mathbf{b}(k+1) \tag{6.65}$$

por (6.62), (6.63) e (6.65)

$$\mathbf{w}(k+1) = w(k) + \mathbf{Z}^{\#}\,\delta\,\mathbf{b}(k) \tag{6.66}$$

sendo $\mathbf{b} = (b_1, b_2, \ldots, b_n)^T$ todos positivos, o algoritmo pode seguir o seguinte percurso de ajustamento, para w e $\mathbf{b}$

$$w(1) = \mathbf{Z}^{\#}\,\mathbf{b}(1) \qquad \mathbf{b}(1) > 0 \tag{6.67}$$

$$e(k) = \mathbf{Z}w(k) - \mathbf{b}(k) \tag{6.68}$$

O ajustamento do w e do $\mathbf{b}$ podem ser efectuados do seguinte modo

$$w(k+1) = w(k) + c\mathbf{Z}^{\#}\,[e(k) + |e(k)|] \tag{6.69}$$

$$\mathbf{b}(k+1) = \mathbf{b}(k) + c[e(k) + |e(k)|] \tag{6.70}$$

6.4-3 EXEMPLO DE CLASSIFICAÇÃO PELO ALGORITMO *LMSE*

Vai ser aplicado o algoritmo *LMSE*[4] ao conjunto de amostras/protótipos **z**, já utilizadas no exemplo de aplicação do algoritmo de determinação da função de decisão pelo processo de aproximação com correcção do erro. Esses protótipos estão apresentadas no quadro 6.3. Os vectores de características com três componentes, foram expandidos e são apresentados na forma $z = (x_1, x_2, x_3, 1)^T$. Estas amostras dizem respeito a ensaios efectuados em madeira de pinho, e assinalam três tipos de *classes/defeitos*: Picado/Cardido (*CRD*), cinzento da madeira (*C_Z*) e madeira sem defeitos (*SDF*). A razão de ser, para se considerar uma classe *SDF* na madeira, resulta da necessidade de distinguir numa amostra, as peças com defeito, das peças ou

[4]O algoritmo *LMSE* é implementado pela função com o mesmo nome, em TRCPAD.C. Os dados de entrada são recolhidos em RSPV.TXT (protótipos de classificação) e as informações de saída são recolhidas em MAT001.TXT. Os vectores de classificação são enviados para PCTR.TXT

segmentos, que apesar de seleccionados (pela segmentação), não constituem qualquer defeito.

As variáveis características seleccionadas empiricamente foram, a média às diferenças, o contraste às diferenças e o 2º momento angular às diferenças[5].

O valor dos três hiperplanos de decisão, são calculados através da expressão

$$w = Z^{\#} \cdot b \qquad (6.71)$$

em que o significado de $Z^{\#}$ e **b** está referida em 6.4-1.

Como se pretendem três hiperplanos de decisão, esta operação tem de ser repetida três vezes, já que se procuram hiperplanos que separem cada uma das classe de todas as outras.

Os quadros seguintes, representam as três matrizes Z^TZ, as três matrizes $(Z^TZ)^{-1}$ e as três matrizes $Z^{\#} = (Z^TZ)^{-1}.Z^T$, para o caso de aprendizagem por *LMSE*. Apenas são apresentados a título de demonstração, já que a dimensão do exemplo utilizado apenas se adapta a uma situação meramente teórica.

	0	1	2	3
0	3421,45	6650,06	916,17	249,27
1	6650,06	13439,28	1623,10	464,21
2	916,17	1623,10	295,72	73,16
3	249,27	464,21	73,16	19,00

Quadro 6.6 Matriz produto [Z'. Z][1]

	0	1	2	3
0	0,59	-0,17	-0,11	-3,26
1	-0,17	0,05	0,10	0,51
2	-0,11	0,10	0,90	-4,42
3	-3,26	0,51	-4,42	47,32

Quadro 6.7 Matriz inversa $[Z'. Z]^{-1}$[1]

N	0	1	2	3	4	5	6	7	8	9	10	11	12	14	15	16	17	18	19
0	-0,01	-0,10	-0,13	-0,19	0,06	0,04	0,09	0,10	-0,16	-0,16	0,24	-0,19	0,39	0,01	-0,02	-0,13	-0,42	0,10	0,00
1	0,00	0,03	0,04	0,05	0,00	0,01	-0,02	-0,02	0,04	0,05	-0,07	0,06	-0,11	0,00	-0,05	0,06	0,12	-0,01	0,03
2	0,09	0,06	0,18	0,07	0,23	0,34	0,10	0,18	0,07	0,11	0,04	0,12	-0,04	0,15	-0,55	0,33	0,14	0,23	0,30
3	-0,31	0,47	0,01	1,04	-1,59	-2,02	-0,98	-1,65	0,70	0,50	-1,72	0,61	-2,24	-0,89	3,53	-1,18	1,97	-2,15	-1,84

Quadro 6.8 Matriz $Z^{\#}=[Z'. Z]^{-1}Z'$[1] (matriz pseudo - inversa)

	0	1	2	3
0	3421,45	6650,06	916,17	249,27
1	6650,06	13439,28	1623,10	464,21
2	916,17	1623,10	295,72	73,16
3	249,27	464,21	73,16	19,00

Quadro 6.9 Matriz produto [Z'. Z][2]

	0	1	2	3
0	0,59	-0,17	-0,11	-3,25
1	-0,17	0,05	0,10	0,51
2	-0,11	0,10	0,90	-4,42
3	-3,25	0,51	-4,42	47,31

Quadro 6.10 Matriz inversa $[Z'. Z]^{-1}$[2]

[5]ver nota (2) e capítulo 4.

N	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	-0,10	0,16	0,16	-0,24	0,19	-0,39	-0,01	0,02	0,01	0,10	0,13	0,19	-0,06	-0,04	-0,09	-0,13	-0,42	0,10	0,00
1	0,02	-0,04	-0,05	0,07	-0,06	0,11	0,00	0,05	0,00	-0,03	-0,04	-0,05	0,00	-0,01	0,02	0,06	0,12	-0,01	0,03
2	-0,18	-0,07	-0,11	-0,04	-0,12	0,04	-0,15	0,55	-0,09	-0,06	-0,18	-0,07	-0,23	-0,34	-0,10	0,33	0,14	0,23	0,30
3	1,65	-0,70	-0,50	1,72	-0,61	2,24	0,89	-3,53	0,31	-0,47	-0,01	-1,04	1,59	2,02	0,98	-1,18	1,97	-2,15	-1,84

Quadro 6.11 Matriz $Z^{\#}=[Z'.\ Z]^{-1}Z'[2]$ (matriz pseudo - inversa)

N	0	1	2	3
0	3421,45	6650,06	916,17	249,27
1	6650,06	13439,28	1623,10	464,21
2	916,17	1623,10	295,72	73,16
3	249,27	464,21	73,16	19,00

Quadro 6.12 Matriz produto [Z'. Z][3]

N	0	1	2	3
0	0,59	-0,17	-0,11	-3,25
1	-0,17	0,05	0,10	0,51
2	-0,11	0,10	0,90	-4,42
3	-3,25	0,51	-4,42	47,31

Quadro 6.13 Matriz inversa $[Z'.\ Z]^{-1}[3]$

N	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	0	-0,1	0,42	0,13	0,02	0,01	0,39	-0,19	0,24	-0,16	-0,16	0,1	-0,09	-0,04	-0,06	0,19	0,13	0,1	0,01
1	-0,03	0,01	-0,12	-0,06	0,05	0	-0,11	0,06	-0,07	0,05	0,04	-0,02	0,02	-0,01	0	-0,05	-0,04	-0,03	0
2	-0,3	-0,23	-0,14	-0,33	0,55	0,15	-0,04	0,12	0,04	0,11	0,07	0,18	-0,1	-0,34	-0,23	-0,07	-0,18	-0,06	-0,09
3	1,84	2,15	-1,97	1,18	3,53	-0,89	-2,24	0,61	-1,72	0,5	0,7	-1,65	0,98	2,02	1,59	-1,04	-0,01	-0,47	0,31

Quadro 6.14 Matriz $Z^{\#}=[Z'.Z]^{-1}Z'[3]$ (matriz pseudo - inversa)

w	w1	w2	w3	w4
1	-0,48	0,23	2,14	-7,73
2	-0,42	0,19	-0,15	1,33
3	0,90	-0,41	-2,00	5,41

Quadro 6.15 Hiperplanos de decisão, obtidos por *LMSE*

x	x1	x2	x3	x4	d1=w1'x	d2=w2'x	d3=w3'x
1-SDF	10,38	14,14	4,76	1,00	5,53	3,14	-9,57
2-SDF	10,07	13,61	4,75	1,00	0,73	-1,03	-0,61
3-CRD	16,04	37,89	2,76	1,00	-0,81	1,38	-1,21
4-CRD	14,36	29,05	3,02	1,00	-1,18	0,34	0,10
5-C_Z	10,73	16,33	4,20	1,00	-0,14	-0,70	0,03
6-C_Z	11,45	18,41	3,98	1,00	-0,47	-0,58	0,21
7-CRD	18,86	45,33	2,85	1,00	-0,26	1,59	-1,90
8-C_Z	12,08	20,03	3,83	1,00	-0,72	-0,51	0,40
9-SDF	10,84	15,24	3,83	1,00	0,63	-1,03	-0,48
10-C_Z	11,89	19,29	3,90	1,00	-0,65	-0,58	0,40
***	...	...	...	...	SDF	CRD	C_Z

Quadro 6.16 Tabela de classificação de uma amostra diversificada de padrões, pertencentes a três classes diferentes.

O quadro 6.16 faz a descrição, daquela que seria uma tarefa de classificação, a partir dos hiperplanos de decisão calculados pelo processo implementado pelo algoritmo de aprendizagem ***LMSE***. As superfícies de decisão d_1, d_2, e d_3, com valores indicados na Quadro 6.15, são aplicadas a um conjunto de amostras x, das quais se obtém uma classificação através do produto desses vectores pelos diversos vectores classificadores w.

6.5 Funções de Decisão Estatística

6.5-1 Generalidades

Este tipo de classificadores são os mais utilizados quando as circunstâncias em que assenta a classificação, se baseia em categorias de padrões que se caracterizam por um conjunto de parâmetros que podem ser conhecidos à priori (classificação paramétrica). Num sistema de classificação estatístico, usa-se a função densidade de probabilidade conhecida/ou não, de um conjunto de variáveis conhecidas para cada uma das classes. Essa função pode assentar numa distribuição paramétrica com forma conhecida. Por exemplo podem ser usadas distribuições gaussianas, distribuições de t'Student, ou qualquer outra que se saiba seguir o comportamento dos padrões das amostras que caracterizam cada uma das classes.

- A formulação de um algoritmo de classificação de padrões, através de processo estatístico, parte do pressupostos seguintes:

  - A probabilidade à priori $p(w_i)$ da classe w_i é conhecida , ou pode ser estimada

  - A função densidade de probabilidade condicional de x, $p(x/w_i)$, é conhecida ou pode ser estimada pelo conjunto de treino.

  - A probabilidade à posteriori $p(w_i/x)$, probabilidade que o vector x tem de provir da classe w_i, pode ser determinada.

O objectivo de uma função de decisão estatística, consiste em minimizar as perdas médias de classificação incorrectas, e simultaneamente o risco médio condicional $r_k(x)$ de classificações mal feitas, que pode ser definido pela expressão (6.72). Na expressão do risco, a função L_{ij} é definida como uma função custo de

penalidade, atribuído à tomada de uma decisão errada. Por exemplo decidir que $x \in w_j$ quando de facto $x \in w_i$, representa um custo, variável com as consequências da decisão.

$$r_k(x) = \sum_{i=1}^{M} L_{ik} p(w_i / x) \qquad (6.72)$$

(6.72) representa a perda média esperada, para uma má classificação de x em w_k, quando deveria ter sido classificada numa outra classe w_i, $i=1,2, \ldots, M$ com $i \neq k$.

O objectivo dum classificador é encontrar sempre a decisão óptima, que permita minimizar o risco ou custos da decisão. A decisão deverá basear-se em:

1 - Cálculo de perda esperada $r_i(x)$, atribuída à decisão de $x \in w_i$ $\forall i$, $i=1,2, \ldots, M$

2 - Decidir por $x \in w_k$ se $r_k(x) \leq r_i(x)$ $\forall i$, $i \neq k$.

A função descriminante, pode simplesmente ser representada através da função $d_k(x) = -r_k(x)$, função que minimiza os riscos de classificação.

6.5.2 FUNÇÃO DE DECISÃO DE BAYES

A regra de decisão de Bayes pode ser escrita na forma seguinte:

$$p(w_i / x) = \frac{p(x / w_i) p(w_i)}{p(x)}$$

onde (6.73)

$$p(x) = \sum_i p(x / w_i) p(w_i), \quad \text{com } i = 1,2, \ldots, M$$

é a probabilidade que x ocorra, sem olhar à classe a que pertence.

Substituindo (6.73) em (6.72), obtém-se:

$$r_k(x) = \frac{1}{p(x)} \sum_{i=1} L_{ik} p(x/w_i) p(w_i) \tag{6.74}$$

verifica-se que $p(x)$ é comum a todo o $r_j(x)$, $j=1,2, \ldots, M$, podendo por isso ser eliminado da fórmula do risco médio condicional. A decisão de Bayes consiste portanto na minimização da expressão (6.74), equivalente a

$$min\, r_k(x) = min \sum_{i=1} L_{ik} p(x/w_i) p(w_i) \tag{6.75}$$

O classificador de Bayes baseia-se na minimização desta função, que pode ser traduzida pelo função de decisão

$$d_k(x) = -r_k(x) \tag{6.76}$$

6.5.3 Decisão por Máxima Probabilidade

A expressão que me dá as perdas esperadas, ao decidir que $x \in w_k$ é dada por:

$$r_k(x) = \sum_{i=1}^{M} L_{ik} p(x/w_i) p(w_i) \tag{6.77}$$

que pode ser utilizada, para encontrar por minimização da função, a classe w_k à qual pertence x.

Para o caso particular de duas classes, $x \in w_1$ se:

$$r_1(x) = L_{11} p(x|w_1) p(w_1) + L_{21} p(x|w_2) p(w_2) \tag{6.78}$$

similarmente, o custo de tomar a decisão por w_2 , $x \in w_2$ é dado por: :

$$r_2(x)=L_{12}p(x|w_1)p(w_1) + L_{22}p(x|w_2)p(w_2) \tag{6.79}$$

Na forma matricial,

$$\mathbf{R} = \mathbf{LP} \tag{6.80}$$

ou

$$\begin{vmatrix} r_1 \\ r_2 \end{vmatrix} = \begin{vmatrix} L_{11} & L_{21} \\ L_{12} & L_{22} \end{vmatrix} \begin{vmatrix} p(x/w_1)p(w_1) \\ p(x/w_2)p(w_2) \end{vmatrix} \tag{6.81}$$

A decisão de x pertencer a w_1 será tomada, na condição de:

$$L_{11}P(X|W_1)P(W_1) + L_{21}P(X|W_2)P(W_2) \; < \; L_{12}P(X|W_1)P(W_1) + L_{22}P(X|W_2)P(W_2) \tag{6.82}$$

ou

$$(L_{21}-L_{22})p(x/w_2)p(w_2) \; < (L_{12}-L_{11})p(x/w_1)p(w_1) \tag{6.83}$$

que pode ser posta na forma;

$$\frac{p(x|w_1)}{p(x|w_2)} = \frac{(L_{21}-L_{22})p(w_1)}{(L_{12}-L_{11})p(w_2)} \tag{6.84}$$

Se for usada a notação $L_{12}(x)$ para o quociente $p(x|w_1)/p(x|w_2)$ rácio de probabilidades, e a notação θ_{12} para o quociente $(L_{21}-L_{22})p(W_2)/(L_{12}-L_{11})p(W_1)$, como sendo um valor de limiar, o critério de decisão transforma-se em:

$$x \in w_1 \quad \text{se} \qquad L_{12}(x) > \theta_{12}$$

generalizando, a razão de probabilidades e o limiar de atribuição, podem ser escritos na forma seguinte:

$$l_{ki} = \frac{p(x|w_k)}{p(x|w_i)}$$

e

$$\theta_{ki} = \frac{(L_{ik} - L_{ii})p(w_i)}{(L_{ki} - L_{kk})p(w_k)} \tag{6.85}$$

O critério de decisão pode ser assim estabelecido do modo seguinte:

$$x \in w_k \text{ se } \quad L_{ki}(x) > \theta_{ki} \tag{6.86}$$

Através desta regra, a que se chama *regra da máxima probabilidade* pode ser implementado um classificador, através da condição

$$x \in w_k \text{ se } \quad l_{ki}(x) > \theta_{ki} \qquad \forall i, \quad i = 1, \ldots, M. \tag{6.87}$$

Em muitos problemas de reconhecimento de padrões os custos de má classificação, são nulos para as decisões correctas e iguais para todas as decisões incorrectas. Nestas condições a função de perdas pode ser expressa por $L_{ki}=1-\delta_{ki}$, onde $\delta_{ki} = 1$ quando $k=i$ e $\delta_{ki}=0$ quando $k \neq i$. A regra da máxima probabilidade para este tipo de perdas simétricas resulta na expressão:

$$\frac{p(x|w_k)}{p(x|w_i)} = \frac{p(w_i)}{p(w_k)} \tag{6.88}$$

se $p(w_i) = p(w_k)$ $\forall i,k$ a regra de máxima probabilidade, transforma-se em:

$$x \in w_k \text{ se } \quad L_{ki}(x) > 1 \tag{6.89}$$

Para o caso de perdas simétricas e $p(w_i) \neq p(w_k)$ podemos atribuir $x \in w_k$ se

$$p(x/w_k)p(w_k) > p(x|w_i)p(w_i) \qquad (6.90)$$

sendo agora a função descriminante dada por : $d_k(x) = p(x/w_k)p(w_k)$. a regra de decisão de Bayes reduz-se assim em atribuir x a w_k se $d_k(x) > d_i(x)$ para todo o $k \neq i$. O principal problema fica deste modo reduzido à determinação da densidade de probabilidade $p(x/w_i)$. O classificador de bayes , assumindo perdas nulas para classificações correctas e perdas iguais para classificações incorrectas, possui a propriedade de que, a decisão óptima minimiza a probabilidade do erro da classificação.

O desenho dum classificador de bayes, requer o conhecimento da probabilidade à priori e da densidade de probabilidade das variáveis para cada classe, além dos custos de decisão. Se $p(w_i)$ não for conhecido, o critério da máxima probabilidade, consiste em encontrar a decisão que minimize as perdas médias, na pior das condições.

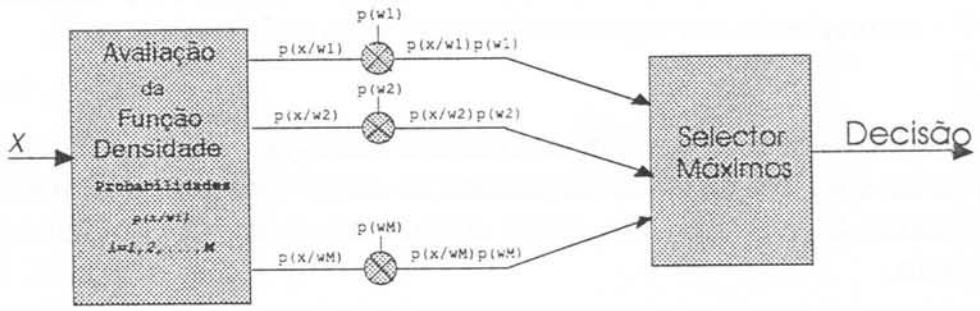


fig. 6.18-Classificador de Bayes

6.5-4 FUNÇÕES DE DECISÃO PARA DISTRIBUIÇÕES NORMAIS DE PADRÕES

Se for assumido que a densidade de probabilidade $p(x/w_i)$ é uma distribuição normal (Gaussiana) de multivariável (ou univariável), o classificador de Bayes toma uma forma particular.

Para o caso de uma variável x, a função densidade de probabilidades é escrita na forma seguinte:

$$p(x)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)^2\right] \tag{6.91}$$

que é completamente definida pela média m e pela variância σ^2, em que

$$m=E(x)=\int_{-\infty}^{+\infty}xp(x)dx \tag{6.92}$$

e

$$\sigma^2=E\{(x-m)^2\}=\int_{-\infty}^{+\infty}(x-m)^2\,p(x)dx \tag{6.93}$$

sendo $E\{\ .\ \}$ a esperança matemática de x, e $p(x)$ a probabilidade de ocorrência de x, representado muitas vezes por $p(x) \sim N(m,\sigma^2)$.

As distribuições normais tendem a juntar os valores de igual probabilidade à volta da média, com uma dispersão proporcional ao desvio quadrático σ. Cerca de 95% dos valores de uma distribuição normal caiem dentro de um intervalo $[2\sigma]$, centrado na média.

Para o caso de uma distribuição de multivariável a função densidade de probabilidades, para a distribuição normal é dada por:

$$p(\mathbf{x}/w_i)=\frac{1}{(2\pi)^{n/2}|\mathrm{Ci}|^{1/2}}\exp\left[-\tfrac{1}{2}(\mathbf{x}-\mathbf{m}_i)'\mathrm{C}_i^{-1}(\mathbf{x}-\mathbf{m}_i)\right] \qquad i=1,2,\ldots,M \tag{6.94}$$

sendo a densidade de probabilidade completamente determinada pela média $\boldsymbol{m}_i$ e pela matriz de covariância $\boldsymbol{C}_i$, definidas do modo seguinte:

$$\boldsymbol{m}_i=E_i\{\boldsymbol{x}\} \tag{6.95}$$

$$\mathrm{C_i}=\mathrm{E_i}\{(\boldsymbol{x}-\boldsymbol{m}_i)(\boldsymbol{x}-\boldsymbol{m}_i)^T\} \tag{6.96}$$

em que $E\{\ .\ \}$ é a esperança matemática para cada uma das classes w_i. , n a dimensão do vector de padrões e $|\ C_i\ |$ o determinante da matriz C_i, sendo C_i simétrica e positiva semidefinida. Os elementos da diagonal c_{kk} representam a variância do elemento k do vector de padrões. Os elementos fora da diagonal principal c_{jk} representam a covariância entre x_j e x_k. Quando estes elementos forem independentes $c_{jk}=0$.

Os padrões tomados de uma distribuição normal, tendem a aparecer em aglomerados, cujo centro é determinado pelo vector média de padrões, e a forma do aglomerado, determinada pela matriz de covariância.

A fig. 6.19 representa três exemplos de aglomerados para cada uma das variáveis, respectivamente para as seguintes situações:

a) $\mathbf{m}=0$ e $\mathbf{C}=\mathbf{I}$

b)
$$m=\begin{bmatrix}1\\0\end{bmatrix} \quad \text{e} \quad c=\begin{bmatrix}c_{11} & 0\\ 0 & c_{22}\end{bmatrix} \tag{6.97}$$

c)
$$m=\begin{bmatrix}0\\1\end{bmatrix} \quad \text{e} \quad c=\begin{bmatrix}c_{11} & c_{12}\\ c_{21} & c_{22}\end{bmatrix} \tag{6.98}$$

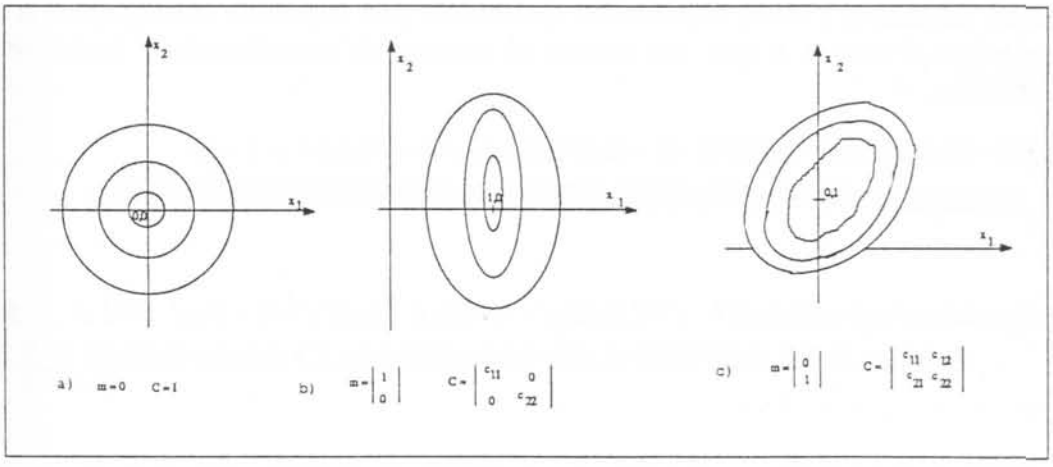


fig. 6.19 Formas diferentes de aglomeração das características para a mesma probabilidade de ocorrência

Os eixos principais dos hiperelipsoides (contornos de igual probabilidade) são dados pelos *"eigenvectores"* de C em que os *"eigenvectores"* determinam os comprimentos relativos desses eixos.

Uma medida de semelhança de um padrão x, em relação à média pode ser determinado por:

$$r^2=(x - m)^t C^{-1}(x - m) \qquad (6.99)$$

que é chamado a distância de Mahalanobis, e que entre duas classes pode ser expressa por:

$$r_{kj}=(m_k - m_j)^t C^{-1}(m_k - m_j) \qquad (6.100)$$

6.5-5 Função discriminante, para distribuições de multivariável

A função de decisão $d_k(x)=p(x/w_k)p(w_k)$ (6.101) para a classe k, pode ser escrita na sua forma logarítmica neperiana por, $d_k(x)=lnp(x/w_k) + lnp(w_k)$, que permite simplificar a forma exponencial apresentada pela expressão $p(x/w_k)$ quando a distribuição é normal, e que em termos de resposta de classificação, é totalmente equivalente.

Aplicando a expressão da distribuição normal (6.94) a (6.101), obtém-se:

$$d_k(x)=lnp(w_k)-(n/2)ln2\Pi \ -(1/2)ln|C_i| - (1/2)[(x - m_i)C_i^{-1}(x - m_i)] \quad i=1,2,\ldots,M \qquad (6.102)$$

como o termo $(n/2)ln2\Pi$ não depende de k, pode ser eliminado da expressão, ficando:

$$d_k(x)=\ln p(w_k) - (1/2)\ln|C_k| - (1/2)[(x - m_k)C_k^{-1}(x - m_k)] \quad k=1,2,\ldots,M \tag{6.103}$$

Esta equação representa a função de decisão de bayes para a classificação padrões, com uma distribuição normal. Esta função de decisão é hiperquadrática, não contendo termos de ordem superior ao 2º grau. O melhor que o classificador de bayes pode fazer, em termos de superfície de decisão, assumindo a distribuição normal, é colocar uma superfície quadrática entre cada par de classes de padrões.

Se as matrizes de covariância forem iguais $C_k=C$ para todo o $k=1,2,\ldots,M$, a função de decisão fica reduzida a uma função linear, na forma:

$$d_k(x)=lnp(w_k) + x^TC^{-1} m_k - (1/2)m^T_kC^{-1} m_k \quad k=1,2,\ldots,M \tag{6.104}$$

Se além disso, $C=I$, sendo I a matriz identidade e $p(w_k)=1/M \quad k=1,2,\ldots,M$ a expressão reduz-se a:

$$d_k(x)= x^Tm_k - (1/2)m^T_km_k \quad k=1,2,\ldots,M \tag{6.105}$$

Decorre de (6.104) que a equação de decisão entre duas classes w_k e w_j é dada por:

$$d_k(x) - d_j(x)= \\ =lnp(w_k) - lnp(w_j) + x^TC^{-1}(m_k-m_j) - (1/2)m^T_kC^{-1}m_k + (1/2)m^T_jC^{-1} m_j \tag{6.105}$$

que é a equação de um hiperplano de decisão entre as duas classes, com distribuição normal dos seus padrões pelas duas classes.

6.5.6 EXEMPLO DE CLASSIFICAÇÃO PELO MÉTODO DE BAYES

É dado um conjunto de protótipos classificados, e conhecendo à partida o número de classes e as probabilidades à priori de ocorrência das classes w_i (evento i). Neste exemplo foi atribuído a probabilidade de ocorrência, da classe 1-SDF - 60%, 20% para a probabilidade de ocorrência da classe 2 - CRD, e 20% a

probabilidade de ocorrência para a classe C_Z. Procura-se em primeiro lugar determinar a probabilidade de ocorrência $p(x/w_i)$ de cada classe que é a função densidade de probabilidade do vector de padrões x, para a classe w_i. Admitiu-se que a distribuição seguia um comportamento normal $N(m,\sigma^2)$. A determinação de $p(x/w_i)$ resume-se assim à determinação de m_i (vectores de médias dos padrões da classes w_i) e à determinação de C_i (matrizes de covariância de cada uma das amostras de protótipos, pertencentes cada uma das classes).

Os protótipos utilizados na determinação dos vectores de médias, e das matrizes de covariância, foram os mesmos dos já utilizados no exemplo descrito em 6.2-3 e 6.3-4.

Os hiperplanos de decisão resumem-se assim à determinação do valor da expressão $d_k(x)= p(x/w_k).p(w_k)$. No cálculo recorreu-se à expressão deduzida no ponto anterior, dada por:

$$d_k(x)=\ln p(w_k) - (1/2)\ln|\mathbf{C}_k| - (1/2)[(x - m_k)\mathbf{C}_k^{-1}(x - m_k)] \qquad k=1,2,\ldots,M \tag{6.107}$$

m_i é determinado por

$$m_i = E_i \{ x \} \tag{6.108}$$

e C_i é determinado por

$$C_i = E_i \{ (x - m_i) (x - m_i)^T \} \tag{6.109}$$

Os quadros 6.15 a 6.17 especificam respectivamente os valores encontrados pelo algoritmo que implementa a função de Bayes1( )[6], para os vectores de média, covariância, e inversa da matriz de covariância, para as três classes em consideração. O quadro 6.18 indica os resultados de uma amostra de classificação, efectuada pela aplicação da expressão do classificador (6.107).

De salientar que os protótipos, as classes e os dados objecto de classificação são os mesmos que foram utilizados no algoritmo de classificação ***LMSE***.

---

[6]Bayes1 é uma função implementada em TFUNC.C, que através de um processo de decisão estatístico desenvolvido por Bayes, atribui a um vector de características, uma classe. A decisão é uma função do valor tomado pelas equações de decisão, em número igual ao número de classes. Bayes1( ) abre vários ficheiros, sendo RSPV.TXT o arquivo onde vais buscar os protótipos de cada uma das classes, RBAYS.TXT o ficheiro onde lança os resultados finais (objecto da classificação), MBAYCV.TXT ficheiro onde lança os valores dos vectores de: médias, matrizes de covariância e respectivas inversas, e finalmente DADBAYS.TXT onde vai buscar os dados para classificação.

m1	0
0	10,27
1	13,87
2	4,85

m2	0
0	16,14
1	35,94
2	2,93

m3	0
0	12,05
1	19,90
2	3,95

Quadro 6.17 Vectores de médias das classes 1, 2 e 3

C1	0	1	2
0	0,15	0,30	0,00
1	0,30	0,63	-0,03
2	0,00	-0,03	0,01

C2	0	1	2
0	1,16	4,02	-0,04
1	4,02	15,39	-0,25
2	-0,04	-0,25	0,01

C3	0	1	2
0	1,60	4,35	-0,21
1	4,35	11,81	-0,58
2	-0,21	-0,58	0,03

Quadro 6.18 Matrizes de Covariância para as classes 1, 2 e 3

1/C1	0	1	2
0	4653,13	102,17	607,71
1	-2321,78	-33,83	-215,18
2	-3250,09	-431,2	173,46

1/C2	0	1	2
0	-2321,78	-33,83	-215,18
1	1160,21	11,31	76,95
2	1624,73	145,61	-47,78

1/C3	0	1	2
0	-3250,09	-431,2	173,46
1	1624,73	145,61	-47,78
2	2344,55	1992,51	327,28

Quadro 6.19 Matrizes inversas da Covariância de cada uma das classes

Os valores encontrados para o logaritmo, dos determinantes das matrizes de covariância, e de $-(1/2)\ln|C_k|$, são dados respectivamente por 6.61, 3.52 e 4.78 para as classes um, dois e três, respectivamente.

Classes/x	*x1*	*x2*	*x3*	*d1*	*d2*	*d3*
1-SDF	10,38	14,14	4,76	3,62	-2200,47	-147,44
2-SDF	10,07	13,61	4,75	5,49	-2255,98	-101,58
3-PCD	16,04	37,89	2,76	-53271,91	0,07	-2269,80
4-PCD	14,36	29,05	3,16	-12646,96	-11,67	-421,26
5-C_Z	10,73	16,33	4,20	-249,12	-1019,00	1,67
6-C_Z	11,45	18,41	3,98	-574,60	-617,51	1,66
7-PCD	18,86	45,33	2,85	-76864,13	-1,50	-1938,55
8-C_Z	12,08	20,06	3,83	-811,09	-384,76	0,48
9-SDF	10,84	15,24	4,70	3,71	-1971,53	-166,45
10-C_Z	11,89	19,29	3,90	-463,23	-455,94	1,84

Quadro 6.20 Tabela de demonstração, duma operação de classificação pelo método de Bayes.

6.6 Conclusões

Este capítulo procura fazer um resumo, de algumas das técnicas mais utilizadas na classificação de padrões - *reconhecimento de padrões*.

Dessa forma, o primeiro processo de classificação apresentado, foi o método de classificação por mínima distância. Seguiram-se métodos de classificação, utilizando classificadores lineares obtidos por processos de pesquisa determinística, baseada em gradiente e correcção do erro. Foi também descrito um método de obtenção dos classificadores através do *LMSE "least-mean-square-error algorithm"*. Finalmente foram aflorados métodos estatísticos, baseados nomeadamente nos processos de decisão de Bayes.

Apresentaram-se exemplos de classificação de defeitos em madeira, sempre baseados no valor de três características seleccionadas empiricamente (média às diferenças, contraste e média inversa às diferenças), utilizando os métodos referenciados anteriormente. As características foram extraídos de protótipos de amostras de madeira, previamente classificadas, com base nas quais foram determinados os classificadores (aprendizagem supervisada). Um conjunto de amostras foi depois sujeito a classificação, seguindo abordagens de mínima distância, classificação linear determinística, e metedologia de Bayes.

O capítulo tem por principal objectivo, introduzir conceitos teóricos ligados aos processos de *classificação/reconhecimento de padrões*, utilizados no *Capítulo VII* desta dissertação, na identificação de defeitos presentes em pranchas de madeira de pinho.

PARTE 3

LOCALIZAÇÃO E RECONHECIMENTO
DE DEFEITOS EM MADEIRA DE PINHO

CAPÍTULO VII

TRATAMENTO DE DEFEITOS EM MADEIRA IMPLEMENTAÇÃO E RESULTADOS

RUI BEBIANO

7- LOCALIZAÇÃO E IDENTIFICAÇÃO DE DEFEITOS EM MADEIRA DE PINHO.

IMPLEMENTAÇÃO E RESULTADOS

7.1 INTRODUÇÃO

O sistema apresentado, descreve um conjunto de processos, que têm por objectivo último, a localização e identificação de defeitos em pranchas de madeira. Mais precisamente, essa análise debruça-se sobre a detecção de defeitos em pranchas aplainadas, já que as propriedades de reflexão das superfícies, são influenciadas pelo estado das mesmas (rugosidade, polimento, limpeza), o que implica, consequências nos resultados e na forma de tratamento dado às imagens. Por suposição, a aplicação do processo de reconhecimento, a pranchas de madeira aplainada ou por aplainar, apresentaria resultados provavelmente divergentes.

O resultado de qualquer sistema de visão, de natureza supervisada, está completamente dependente da caracterização que for feita dos objectos a identificar (neste caso defeitos em madeira) e portanto da qualidade das amostras protótipos que tiverem sido utilizadas, ou seja da relevância que tiverem para a categorização do problema. Nesse sentido as variáveis características envolvidas no processo de classificação, ou de categorização, só terão eficácia se tiverem sido bem escolhidas, o que implica necessariamente a sua consistência absoluta.

Outro factor essencial a ter em conta, é que as imagens dos protótipos, em primeiro lugar, e depois as imagens das amostras em classificação, tenham sido obtidas nas mesmas condições de iluminação. Para um bom desempenho, as imagens devem ainda apresentar as mesmas características de resolução, e no que toca à disposição/orientação da prancha, as imagens deverão ter sido adquiridas mantendo a mesma disposição relativa em relação à câmara.

Dada a natureza, e dimensão do problema em apreciação, e à necessidade consequente de o restringir, optou-se pelo estudo do problema da *localização e identificação de defeitos em madeira*, limitado a uma única espécie vegetal. Essa escolha recaiu sobre o *'pinho bravo'* dado ser essa uma das espécies vegetais mais frequentes no nosso coberto florestal. Isso representou uma maior facilidade de recolha de amostras, do que a que se verificaria se se tivesse optado por outra espécie de madeira, por exemplo uma madeira dura (nogueira, carvalho, etc) ou por uma qualquer espécie exótica. Além disso a vulgaridade com que é utilizada, justificaria só por si essa escolha.

Neste estudo não se procuram soluções definitivas, ou soluções que visem uma aplicação industrial imediata, mas apenas estabelecer um contributo experimental, e se possível, pontos que ancorassem e favorcessem o lançamento eventual, de um estudo futuro mais profundo.

Dada a variedade de aspectos com que uma prancha de madeira de pinho se pode apresentar, mesmo na hipótese de não apresentar qualquer defeito, impõe sempre que se resolva em primeiro lugar, o problema da presença ou não de defeito na peça de madeira. Ou seja, encontrar um processo de decisão consistente, que permita com segurança, saber se a prancha em causa é portadora ou não de qualquer defeito. A primeira tentação para resolver o problema, seria considerar sem defeito, toda a imagem que apresentasse um histograma unimodal. Sucede que isso não acontece de todo, já que as imagens de madeira de pinho têm texturas variáveis, apresentam-se geralmente com muitos veios, ou com tonalidades variáveis bastante acentuadas, e outras vezes com veios de cerne dominantes que não se constituem como defeito.

As imagens das figuras 7.1 e 7.2 são imagens de pranchas de pinho, sem quaisquer defeitos visíveis. A aplicação do programa de reconhecimento RDMP (*Reconhecimento de Defeitos em Madeira de Pinho*) identifica a segunda imagem como não tendo defeitos, mas não é capaz de o fazer de uma forma imediata em relação à segunda.

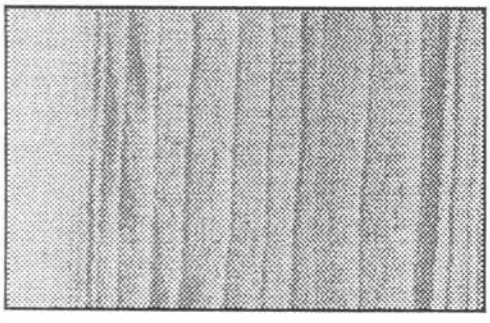


fig. 7.1 Imagem sem defeitos com veios bastante marcados

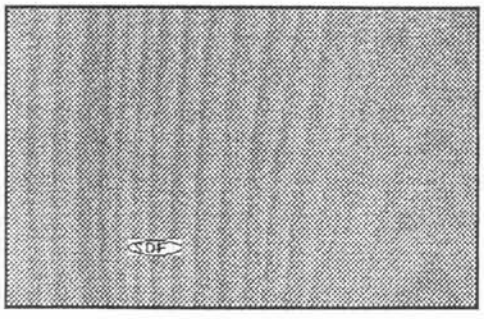


fig. 7.2 Imagem sem defeitos (SDF), com veios muito suaves, identificada pelo RDMP/ad-hoc

A imagem da figura 7.2, foi identificada como não tendo defeitos, através da análise do seu histograma, através da constatação de que a sua variância era reduzida, como se pode concluir da observação do histograma da fig. 7.3. A inspecção do histograma da imagem não possibilita contudo, informação suficiente para se poder concluir se uma imagem de madeira é portadora de defeitos ou não. O caso da fig. 7.1 é um caso complicado, não se podendo tirar conclusões sem se recorrer a técnicas adicionais. É notório e significativo que apesar da imagem da figura 7.2 apresentar uma variância elevada, ela não é portadora de qualquer defeito, que possa diminuir o seu valor ou as suas características como material.

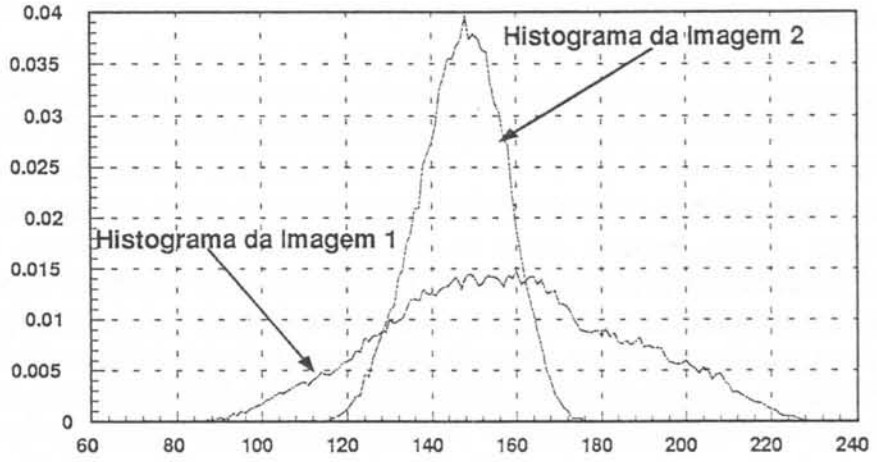


fig. 7.3 Histogramas das imagens das fig. 7.1 e fig. 7.2

Deste modo a primeira identificação que deve ser feita pelo sistema de reconhecimento (nas pranchas serradas e aplainadas), prende-se em saber se têm ou não defeitos. Essa indicação é referenciada nos algoritmos pela designação *SDF[sem defeito]*.

A segunda tarefa consiste em identificar se a imagem contém ou não um fundo (*"background"*), não constituído por madeira. A aquisição da imagem da peça de madeira é feita sobre uma base escura, condicionada a um fundo negro para fazer realçar a madeira, sempre de cor muito mais clara. Esse *"defeito"* é assinalado pelo sistema sob a designação de *B_K(background)*.

De um conjunto vasto de defeitos que podem ser considerados na análise em causa, tal como está referido no capítulo I, optou-se por um conjunto de defeitos mais limitado, reduzindo assim também por esta via a dimensão do problema. Os defeitos considerados para além de *SDF* e de *B_K*, foram os seguintes: *casca(CSC)*, *nó(NO)*, *brecha/fenda(bch)*, *azulado/cinzento da madeira(C_Z)*, *bolsa de resina(BRN)* e *ardido/cardido/picado(CRD)*.

Estes defeitos além de serem os mais vulgares, são também os que mais importa identificar em termos das consequências que provocam nas características mecânicas e físicas das peças de madeira em que estão presentes, e nas implicações estéticas que motivam.

O problema da *identificação de defeitos em madeira*, pode para o sistema implementado, ser dividido em três partes (fases um, dois e três), abstraindo duma inicial que podemos chamar *fase zero*. Esta fase zero, a que chamámos de pré-tratamentos (usualmente filtragens), quase sempre pode ser ultrapassada[1].

Assim, a *fase 1* é a fase de identificação global. Nesta fase pretende-se saber se estamos perante uma prancha serrada e aplainada defeituosa, ou se pelo contrário estamos perante uma prancha de boa qualidade - *SDF*. O processo utiliza o método de *Bayes* (cap. 6.5.2), para através de um vector de características (formado pela variância e características de projecção) decidir se estamos ou não, na presença de boa madeira. Esta fase de decisão deve ser robusta (não se deve enganar na classificação, para o caso da decisão ser tomada pela classe de madeira sem defeitos, *SDF*) e em caso de dúvida, o algoritmo deverá prosseguir.

A *fase 2* é a fase seguinte, e diz respeito à localização dos defeitos através de segmentação e extracção dos contornos da região. Nesta fase é feita a divisão da imagem em zonas homogéneas e leitura/delimitação dos seus contornos. De salientar que os defeitos na madeira aparecem geralmente com uma tonalidade mais escura do que a madeira clara. Pelo facto de a madeira de pinho apresentar muitos veios, aparecem na imagem segmentada, regiões que não correspondem a defeitos reais, e que é necessário eliminar. Isso obriga a que na fase posterior, se testem essas mesmas regiões, com base em hiperplanos de decisão (cap. 6.3) capazes de as poder identificar segundo a classe a que pertencem. Utiliza-se para tal uma árvore de decisão hierárquica, que numa ordem pré-definida, vai eliminando as várias hipóteses sucessivamente.

A *fase 3* é a fase de classificação propriamente dita, em que os vários blocos de imagem são aceites como defeitos, ou como madeira de boa qualidade*(SDF)*. A teoria da decisão e um classificador hierárquico, são os grandes instrumentos que esta etapa utiliza.

[1]As operações de pré-tratamento são muitas vezes úteis. Por exemplo, o efeito conseguido numa segmentação por *"threshold"*, através da utilização prévia de uma filtragem de média, é a obtenção de patamares de segmentação mais adequados, e a consequente delimitação de regiões de segmentação na imagem, mais bem definidas. No algoritmo de *localização* e *identificação de defeitos em madeiras*, não é contudo utilizada, porque isso aumentaria bastante o tempo de execução do processo, e o efeito pode ser compensado por um processo de extracção de regiões conexas na imagem, operação de que não se pode prescindir na implementação do algoritmo.

O sistema possibilita um acompanhamento visual, já que permite fazer o *"display"* da imagem em diversas fases do processo de execução. O objectivo pretendido, é dar ao utilizador, a ideia de progressão em que opera e se desenvolve o algoritmo.

A consistência da classificação não é absoluta, e dificilmente o podia ser, já que a base de classificação, assenta em características, que não são estáveis para o caso da *identificação de defeitos em madeiras*. A grande dificuldade prende-se efectivamente com a selecção das características, ajustadas à tarefa de classificação. Essas características podem basear-se em propriedades decorrentes da geometria dos defeitos, ou basearem-se nas propriedades de micro-textura das superfícies dos defeitos. Sucede que estas propriedades não são estáveis, nem dentro da mesma *classe/defeito*, nem de amostra para amostra, introduzindo por essa via, dificuldades muito difíceis de ultrapassar.

Resumidamente podemos dizer que no capítulo se analisa e descreve um conjunto de processos utilizados na *localização e identificação de defeitos em peças serradas e aparelhadas de pinho*.

Em primeiro lugar é apresentado um processo *'ad hoc'*, assim designado por basear a escolha das suas características em testes directos sobre a imagem e na comparação empírica dos resultados obtidos. Procura primeiro identificar se a imagem é globalmente defeituosa ou não, e se contém ou não *background*. Se a peça de madeira contiver defeitos, a análise prosseguirá *defeito* a *defeito*, através de um processo inicial de localização do defeito potencial, seguido da sua classificação.

Faz-se ainda uma abordagem, a alguns processos de selecção de características e referências ao conceito de extracção de características. O poder de separação de uma característica é medido de várias formas, entre elas, o da medição da distância Euclidiana inter-classes, para essa(s) característica(s), ou através da medição da distância inter-classes pelo método de Jeffries-Matusita (J-M).

Seguem-se três processos de classificação e teste de defeitos: um através de *decisão singular de etapa única*, outro, um *processo de decisão binária global (SDF/CDF)* e finalmente um *processo de decisão binária hierarquizada*. Estes processos foram avaliados e permitem concluir que o processo de decisão binária hierarquizada apresenta resultados melhores que os anteriores. É possível também concluir pela avaliação feita, que existe um conjunto de defeitos mais faceis de identificar: *backup, nós, bolsas de resina e brechas de resinagem*. Estes defeitos apresentam taxas de êxito maiores (maior eficiência) do que os quatro defeitos restantes. A baixa taxa de sucesso dos defeitos *SDF, CRD, AZLD e CSC* compromete a taxa de sucesso global. No que se refere à localização dos defeitos, os algoritmos comportam-se de uma forma robusta.

Um processo de reconhecimento de defeitos pré-marcados a giz é também apresentado e testado com resultados aceitáveis.

7.2 Recursos Informáticos e Estruturas Utilizadas

O sistema foi totalmente implementado num PC compatível, dotado de um microprocessador 386SX a 20MHz e 2MBytes de RAM. A utilização do sistema operativo MS-DOS, acarretou sérias limitações ao sistema, principalmente a nível da gestão da memória e da disponibilidade para a utilizar. Todas as funções foram implementadas em C standard, tendo sido utilizado como sistema de desenvolvimento o C++ da Borland, versão 2.0, e a sua biblioteca de referência. Não foi usado na implementação qualquer sistema de imagem ou biblioteca de funções de imagem. Apenas as imagens de pranchas de madeira serrada e aplainada, foram previamente gravadas através de uma câmara CCD, e da utilização do sistema de imagem Vision*plus*-AT e da biblioteca ITEX MFG, através da utilização de algumas das suas funções de aquisição de imagem.

As imagens foram pré gravadas no formato 512x512 bytes(sem cabeçalho) e posteriormente recortadas para o formato de 320x200 (colunas x linhas), às quais foi acrescentado um cabeçalho de 4bytes, com a indicação do número de colunas e do número de linhas respectivamente[2]. Isso foi uma consequência da necessidade de ajustar a dimensão da imagem, à dimensão de um segmento de memória de endereçamento directo (64kbytes). As limitações do sistema operativo e do equipamento em particular, obrigaram a uma preocupação permanente em relação à melhor forma de gerir, os 640kbytes de memória disponível do sistema (MsDOS), dos quais subjavam escassos 200kbytes, que tinham de ser guardados para utilização de memória dinâmica, onde eram carregados todos os *"buffers"* de imagem. Os algoritmos de processamento e análise de imagem, sendo grandes consumidores de memória (mesmo para imagens de pequena dimensão), obrigaram à reformulação de alguns algoritmos, e à utilização preferencial de estruturas dinâmicas (árvores e listas ligadas), que em determinadas situações permitiram a economia de espaço, através da libertação do espaço ocupado por essas estruturas, quando o seu limite de utilidade era atingido. O aumento de velocidade destes algoritmos é na maior parte das vezes conseguido, pelo facto de se trabalhar com ponteiros, e pelo facto destas estruturas terem libertado muito espaço redundante (apresentando por isso uma dimensão de tratamento menor), do que as equivalentes estruturas indexadas. A utilização de *"arrays"* de dimensão rígida, para situações de volume de dados elevado, em condições de subutilização desse espaço, criam muitas vezes apesar do indexamento directo, problemas de velocidade aos algoritmos.

[2]A função que executa o recorte das imagens encontra-se em FORMTIMG.C

7.3 Processo ad hoc de Reconhecimento de Defeitos

7.3.1 Introdução

Descreve-se neste ponto do capítulo, um processo heurístico que efectua um conjunto de operações, que visam como resultado responder às seguintes perguntas sobre a imagem:

1º A imagem tem ou não defeitos?
2º A imagem tem ou não *"background"*?
3º Se a imagem tem defeitos, qual a sua localização e a que classe pertencem?

Por outras palavras, o processo deverá ser capaz de fazer a segmentação da imagem, delimitar essas regiões segmentadas e fazer a sua identificação correctamente.

O algoritmo identifica seis tipos de defeitos, referenciados no quadro 7.1, e que são: o defeito *"background"*, o defeito *nó*, o defeito *bolsas de resina*, o defeito *SDF* - madeira escura sem constituir defeito, o defeito *brechas de resinagem*, o defeito podridões do tipo *cardidos* e o defeito *azulado* da madeira.

madeira escura(não defeito)	SDF
Nós	NO
Brechas de resinagem	BCH
Bolsas de resina	BRN
Picado(cardido)	CRD
Acizentado(Azulado da madeira)	C Z

Quadro 7-1 Listagem de defeitos em madeira

Chamámos a este processo de localização e identificação de defeitos em madeira de *ad hoc*, pelo facto da classificação implementada, assentar em medidas efectuadas sobre pranchas de madeira, baseadas em escolhas empíricas. As características foram seleccionadas com base na consistência dos resultados de classificação obtidos, tendo em conta a variação das características ensaiadas durante a fase dos testes. A inspecção e teste dessas medidas foi conseguida, através da utilização de diversas amostras préviamente adquiridas.

Os ensaios de classificação basearam-se nos pressupostos seguintes:

1 - As imagens com *"background"* apresentam sempre uma faixa dinâmica dos níveis de cinzento mais larga, devido à presença do fundo negro na imagem, e uma variância inerente a esse fundo negro reduzida.

2 - As imagens sem defeito apresentam uma variância interna menor do que as imagens com defeitos.

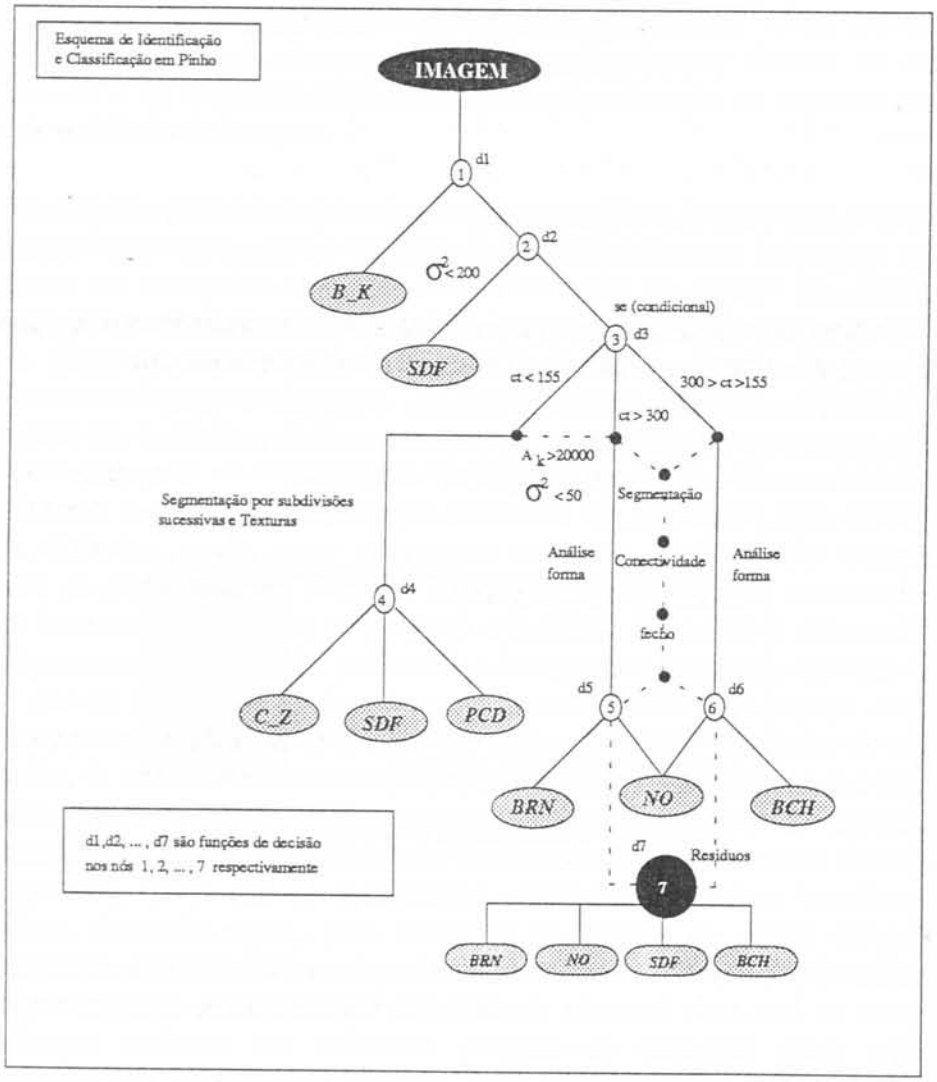


fig. 7.4 Esquema de decisão do algoritmo *RDMP/AD-HOC*
[Reconhecimento de defeitos em madeira de pinho-ad hoc]

3 - A identificação de defeitos, tais como, madeira escura *sem defeito* (*SDF*), *azulados* (*AZLD*), *cardidos* (*CRD*), podem ser identificados através da utilização de medidas estatísticas, retiradas pelo processo *GLDM* (descrito no cap. 5.3-3). Essas medidas são: *média das diferenças*, *contraste das diferenças* e *segundo momento angular das diferenças*, retiradas das regiões segmentadas na imagem.

4 - Os *nós*, as *brechas* e as *bolsas de resina*, podem ser identificados pelas suas características geométricas, em função da sua maior proximidade (semelhança) em relação a um círculo ou a um rectângulo.

7.3.2 ARQUITECTURA DO SISTEMA *RDMP/ad-hoc* (ALGORITMO DE RECONHECIMENTO EM MADEIRA DE PINHO)

O processo de aquisição, processamento, análise e classificação das imagens, obedece à implementação da seguinte arquitectura de processamento:

1 - Aquisição: As imagens das superfícies de madeira, são obtidas por intermédio de um suporte óptico - do tipo matriz CCD, e convertidas para a correspondente imagem discretizada e digitalizada, através da utilização de 256 níveis de cinzento, e uma resolução de 320x200 pontos.

2 - Detecção dos defeitos: Procura-se com o processo de segmentação diferenciar e localizar as regiões escuras na madeira. Estas regiões correspondem geralmente a defeitos. Nas regiões de madeira clara, é pouco provável encontrar estes tipos de defeitos (ref. em 7.3.1). Um processo de segmentação por *"thresholding"* (descrito no cap3.2.2-1) ou um processo de segmentação por subdivisão sucessiva da imagem, utilizando um critério estatístico (algoritmo descrito em 3.6) é utilizado na identificação das regiões, em função do critério de decisão hierárquico aplicado ao nó <3>.

A procura de independência, em relação às condições de iluminação com que foi adquirida a imagem, leva a que o método utilizado por este algoritmo para recolha de características, utilize o

método de extracção de medidas estatísticas *GLDM* (processo descrito em 5.3-3).

As figuras 7-5, mostram respectivamente uma prancha de madeira-*a*, a prancha segmentada-*b* e o resultado depois de aplicação do algoritmo *rdmp/ad-hoc-c*. Isso envolve a segmentação da prancha de madeira e a identificação dos contornos dessas regiões, operação que corresponde ao reconhecimento da localização dos defeitos.

a) b) c)

fig. 7.5 Prancha de pinho com nós *a*, segmentada *b* e com os defeitos localizados e reconhecidos *c*

3 - Análise de imagem: Um processo de recolha de características de imagem é processada para cada uma das regiões com localização identificada. Essas características consistem em medições estatísticas do 2º grau - *GLDM*, média, contraste e 2ºmomento, às diferenças.

$$M = \sum_{k=0}^{L-1} kP_{\delta}(k) \qquad C = \sum_{k=0}^{L-1} k^2 P_{\delta}(k) \qquad ASM = \sum_{k=0}^{L-1} P_{\delta}(k)^2$$

ou características de natureza geométrica, respectivamente a área e a relação de forma Q_f. No cálculo de Q_f são considerados os códigos de cadeia(*"chain code"*). Para cada uma das direcções do contorno $i=0,1,2,3,4,5,6,7$ é medida a frequência f_i, de tal modo que se verifique $f_0>f_1>f_2>f_3>f_4>f_5>f_6>f_7$. A variável Q_f é definida do seguinte modo:

$$Q_f = (f_0 + f_1) / \sum_{i=0}^{7} f_i \qquad (7.1)$$

e a área definida por $A_k = \sum_i \sum_{j \in T_k} f_{ij}$ em que T_k é um conjunto dos pontos da região k, com k=1,2, . . . , n (sendo n o número de regiões).

4 - Classificação: As funções de decisão estatística nos nós 1,2,3,4,5,6 e 7 (esquema da fig. 7.4) são baseados nos algoritmos de aprendizagem de correcção do erro (descritos em 6.3), ou em decisões exclusivas de natureza condicional. No nó sete da estrutura de decisão, a opção pela classe é tomada por mínima distância.

A tabela Quadro 7-2 descreve as características utilizadas em cada um dos nós, e o tipo de função de decisão, em que se baseia a determinação da classe de decisão em cada nó.

nó da estrutura de decisão	características	função de decisão/descrição
d1	$f1,\sigma^2$	if condicional
d2	σ^2	if condicional
d3	ct,σ^2,A_K	if condicional
d4	md,ct,Asm	decisão linear/correcção do erro
d5,d6	Qf	if condicional
d7	md,ct,Asm	mínima distância

Quadro 7.2 Quadro de características por nó de decisão e tipo de classificador

em que f_1, σ^2, A_k, Q_f, *md*, , e *Asm* representam respectivamente
f_1 - Faixa dinâmica dos níveis de cinzentos.
σ^2 - Variância, para a estatística *GLDM*
A_k - Área de uma região
Q_f - Factor de forma
md - Média, para a estatística *GLDM*
ct - Contraste,para a estatística *GLDM*
Asm-2ºmomento,para a estatística *GLDM*

O processo pode ser esquematizado pelo diagrama de fluxos, apresentado no esquema da fig.7.4.

7.3-3 DESCRIÇÃO DO SISTEMA *RDMP/ad-hoc* (ALGORITMO DE RECONHECIMENTO EM MADEIRA DE PINHO)

Através da função raizmadeira(...), o algoritmo de *reconhecimento de defeitos em madeira de pinho*, determina da existência de *"background"* ou não na

imagem. A finalidade é poder separar o *"background"* caso ele exista. O *"background"* beneficia do facto da sua cor ou textura poder ser controlada pelo utilizador, ou mais propriamente pelo desenhador do sistema. Nessa base podemos aceitar que o *bakground* seja formado por um fundo escuro (negro preferencialmente), constituído por cartão ou pano preto. Essa representação vai implicar que o *"background"* na imagem, altere a largura da faixa dinâmica dos níveis de cinzento no histograma. O histograma passa a ser caracterizado por uma maior extensão à esquerda, devido à presença de maior quantidade de pixels, com níveis de cinzento de valor mais baixo na imagem. O *"background"* é também caracterizado pela sua baixa variância interna, o que constitui outro objectivo a medir, nas operações de imagem implementadas por esta função.

Depois de separado o *"background"* da imagem, se existir, é medida a variância[3] na imagem restante. Não é uma medida infalível, já que depende da quantidade e natureza dos veios que a imagem contiver, da tonalidade e das condições em que tiver sido feita a aquisição dessa imagem. É seguro contudo que, se a variância for muito baixa, a madeira (segmento da prancha visualizada) não conterá qualquer defeito. Neste caso o algoritmo terminará o seu desempenho, com a indicação de imagem sem defeito(SDF).

Através de um conjunto de medidas características retiradas da imagem (referidas no quadro 7.2), respectivamente: variância, área da região segmentada e contraste pelo método *GLDM*, é possível tomar a decisão, por qual dos processos de segmentação da imagem optar. Pode ser feita por um processo de segmentação por *"threshold"* normal(método das *concavidades*, ou de *OTSU* - cap. 3.2) ou através da segmentação por subdivisão estatística da imagem (cap.3.6).

No nó quatro (fig.7.4), um processo de escolha binária é efectuado através de decisão condicional implementada por d_4, com base nas características extraidas pela função anterior, e com o objectivo de escolher um dos dois processos de segmentação já mencionados - normal ou por divisão estatística da imagem. Baseia-se a decisão no facto de podermos supor que, uma região de segmentação de imagem com contraste, variância elevada e área reduzida, poder significar a presença de defeitos do tipo: *nós*, *bolsa de resina* ou *brechas de resinagem*, e áreas de segmentação grandes (maiores que 20000 pixels em imagens de 64000 pixels), variâncias baixas e contrastes moderados, poder significar região defeituosa com defeitos do tipo *cardidos*, *azulados* ou *madeira escura não defeituosa*.

É de salientar que o processo de tratamento feito até este ponto do algoritmo de imagem, foi processado globalmente sobre toda a imagem, enquanto depois da segmentação, operada no nó quatro - fig. 7.4, os processos de decisão são aplicados sobre as regiões de segmentação.

---

[3]A função inspcregV1(...) inspeciona a região em função da sua uniformidade à custa da variância, utilizando como argumentos o patamar de segmentação (*threshold*) da imagem e o nivel de separação do *background*, retornando a àrea da imagem separada por segmentação. Determina ainda o contraste existente na região da imagem segmentada. A função raizmadeira(...) determina se há ou não *bakground* na imagem. Estas funções encontram-se no arquivo TSEGMT2.C.

A função le_t(void)[4] (função que constrói e lê uma árvore quaternária) é utilizada na implementação da segmentação por subdivisão sucessiva da imagem, à custa da medição do seu grau de uniformidade textural em cada região específica dessa mesma imagem. A imagem é dividida em quatro partes, sempre que a uniformidade de textura nessas quatro partes de imagem, apresente um gradiente elevado. As medidas com que foram avaliados o gradiente de imagem, foram a média *GLDM* e o contraste *GLDM*, tendo-se optado pela média (*md*), pelo facto do algoritmo apresentar uma execução mais simples (rápida - a determinação do contraste exige o cálculo prévio da média e operações de produtos em vírgula flutuante).

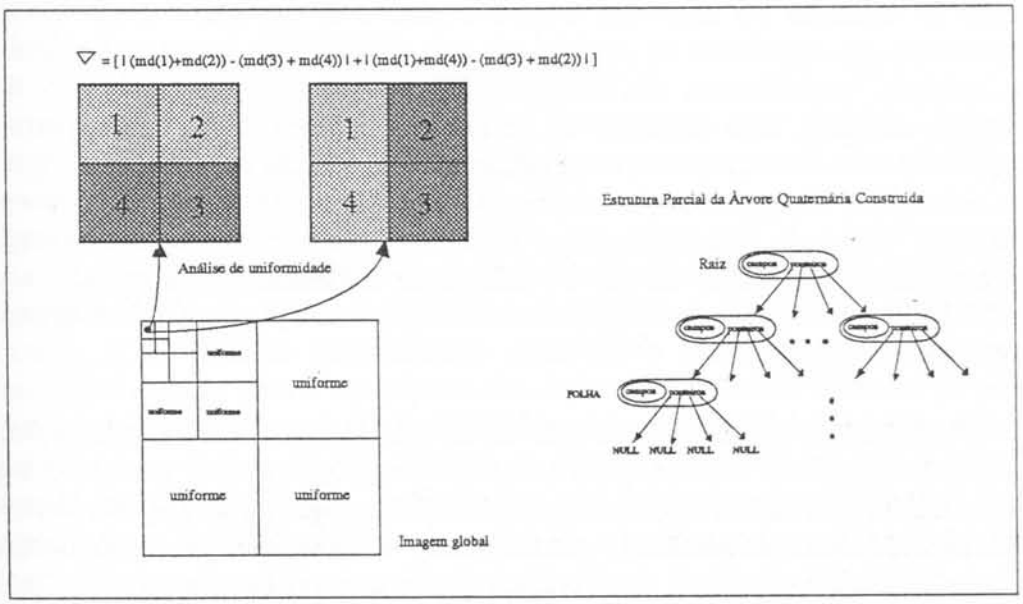


fig. 7.6 Construção da árvore quternária, através do algoritmo de segmentação por subdivisão sucessiva da imagem

[4]As funções que executam a segmentação por divisão da imagem em zonas homogéneas, acha o gradiente da imagem em função do contraste/ou média *GLDM*, e armazena os dados numa estrutura em àrvore, envolvendo o arquivo RBSEGQT.C (segmentação por quadtree). Os argumentos da função que constroi a àrvore são, o centroide da subimagem (imagem dividida em quatro partes), nivel de profundidade da àrvore e o valor do contraste nessa região. A descrição das tarefas desempenhadas pelas funções são as seguintes: le_t(void) - chama as funções de construção da àrvore, de leitura da àrvore, e de eliminação (apagar) da àrvore; Arv_q2(struct quadt_e *arvore)-constroi a àrvore quaternária (versão 2), analisando as condições de homogeneidade através da análise do gradiente descrito na fig. 7.6; readtree(struct quadt_e *arvore) - lê a imagem e faz a sua aquisição para um *buffer;* dlttree(struct quadt_e *arvore)-apaga a estrutura àrvore(delete tree). A estrutura do nó da àrvore é a seguinte:

```
struct quadt_e
{
        unsigned char md, nivel, coordyi;
        int coordxj;
        struct quadt_e  *esqlt, *esqbx, *drtlt, *drtbx;
};
```

Com este algoritmo mede-se o gradiente, por intermédio das propriedades de textura e através da relação entre as quatro regiões em que é partida a imagem. Se a diferença for superior a um determinado *"threshold"* (obtido empiricamente) a região é considerada não uniforme e novamente subdividida. Através deste processo é criada uma dependência entre regiões, representada através de uma estrutura arborescente, com a particularidade de as suas folhas serem constituídas por regiões homogéneas, em que a informação contida, é nem mais nem menos do que a média dos níveis de cinzento nessa região.

Sendo a imagem, depois desta operação, constituída exclusivamente por zonas homogéneas, a binarização por *"threshold"* torna-se muito simples, devido ao facto do seu histograma identificar claramente os patamares de segmentação. Este processo foi utilizado no algoritmo com o objectivo de facilitar a segmentação, principalmente em superfícies de madeira em que os veios são vistos pela máquina como defeitos, prejudicando ou dificultando a localização e identificação dos verdadeiros defeitos. Este processo ao uniformizar regiões homogéneas, permite ultrapassar de certa forma algumas destas dificuldades. A razão pela qual não é esta a forma sistemática de segmentação empregada, deve-se ao facto de este processo utilizar mais tempo de processamento, e ao facto dos resultados de segmentação serem tendencialmente piores do que os obtidos por segmentação normal, em casos de madeira clara com poucos veios presentes. Este processo ofusca em determinadas situações algumas regiões.

Pelos motivos apontados a decisão da escolha do processo de segmentação, é crítica e pode conduzir a erros globais de classificação, já que se explora o lado esquerdo da arquitectura do algoritmo representado na fig. 7.4, que apenas permite identificar dois tipos de defeitos, *cardidos(CRD)* e *azulados(C_Z)* e obviamente madeira escura não defeituosa. Se a entrada por este ramo do algoritmo for falsa, alguns erros acabarão por ser cometidos na classificação final. Na decisão de via prosseguida pelo algoritmo, entram características como a variância, contraste e área da região previamente definida como fazendo parte da segmentação. A decisão é identificada no nó três - fig. 7.4, e pelo critério d_3 descrito segundo o Quadro 7-2.

A segmentação pela via normal, baseia-se numa binarização, efectuada por um ou dois *"threshold(s)"*, em função da existência ou não de *"background"*, através de bnthresd1(...) ou bnthresd(...) respectivamente.

Após a segmentação as operações que se seguem no *rdmp/ad-hoc*, visam retirar o contorno[5] das regiões definidas na imagem. Uma primeira operação explora os pixels conexos na imagem, eliminando desse modo ilhas de pixels desconexos

[5]As funções utilizadas na extracção do contorno são: void conectividade( ); que determina as regiões com *conectividade oito*, no arquivo TSEGMT3.C, void Fecho( ); que fecha uma superfície dos seus espaços ocos, em TSEGMT3.C, void limpz1(...); que efectua um conjunto de operações binárias na imagem, em TCLEAN.C, struct lista1 *segct1(...); que determina o contorno e as componentes de *"chain code"* e unsigned int fill1(...); que enche determinada região com uma cor identificativa, ambas em TBINCT.C.

(fora de regiões uniformes). Uma operação de fecho elimina da imagem os buracos e restaura as dimensões iniciais dos segmentos. A função limpez1(...) é uma operação que completa estas operações e vai facilitar as condições em que a função de extracção do contorno vai operar. Esta função baseia-se numa operação binária, que explora a imagem através de uma janela de dimensão *kxk* (5x5 ou 3x3) e que verifica a seguinte condição:

$$f(m,n)=\begin{cases} 0 \quad se \quad \left(\sum_{m=i-k/2}^{m=i+k/2} \sum_{n=j-k/2}^{n=j+k/2} f(m,n)=0 \right) > kxk/2 & \forall i,j \\ 1 \quad se \quad \left(\sum_{m=i-k/2}^{m=i+k/2} \sum_{n=j-k/2}^{n=j+k/2} f(m,n)=1 \right) > kxk/2 & \forall i,j \end{cases} \tag{7.2}$$

As operações seguintes do *rdmp/ad-hoc*, constroem uma lista ligada de que fazem parte todas as regiões segmentadas na imagem, e que contêm a indicação do contorno, e de algumas características geométricas desse contorno. Armazenam, informações como a área e o perímetro das regiões de contorno (defeitos ou pseudo defeitos). As funções segmct1(...) e fill1(...), que formam as listas de todas as regiões segmentadas e com contorno definido na imagem, são executadas a partir da chamada da função topb_n3(...), função que tem por argumentos o *"buffer"* onde a imagem vai ser tratada(e onde se encontra) e a via de segmentação seguida (opção "0" ou "1" do *rdmp/ad-hoc*) e da qual depende a forma de tratamento dada ao defeito. Esta função faz ainda a eliminação de pontos da cadeia de contornos, que tenham sido perdidos. Com esta operação o sistema fica a conhecer várias medidas características sobre os defeitos (ou regiões defeituosas), caso dos: *"chain code"* de cada contorno (informação armazenada no vector chncd[]) e a partir dos quais se calcula um factor de forma Q_f, da *área* e do *perímetro*, além de características de natureza estatística do 2ºgrau, pelo método *GLDM*, como *contraste*, *segundo momento* e *média* (cap. 5.3-3).

Se a opção de segmentação seguida no *rdmp/ad-hoc* tiver sido zero(*case 0*), o processo será conduzido da forma seguinte:

1 - Abre o ficheiro pctr.txt e lê os dados para o vector *w*, vector pesado de decisão linear, obtido por aprendizagem.

2 - A função de decisão w é aplicada a um vector de características retirado dos segmentos de imagem, e determinada a qual classe pertence esse defeito (função de decisão w determinada por processo determinístico supervisado, através de processo de correcção do erro, cap. 6.3).

3 - O contorno é representado no écran juntamente com a classe a que pertence. Se a classe pertencer a SDF o contorno não é representado no écran e o defeito não é assinalado.

Se a opção de segmentação seguida no *rdmp/ad-hoc* tiver sido um (*case 1*), o processo será conduzido da forma seguinte:

1 - Abre o ficheiro pdt. txt onde se encontra a função de decisão, determinada por mínima distância, e guarda os valores dum vector w.

2 - Decide a classe através de uma escolha condicional, baseada num vector de características, *factor de forma* (Q_f), *entropia(ent)* e *contraste(ct)*, para o caso do nó cinco. Para o nó seis a decisão é baseada num vector com características tais como *factor de forma* (Q_f), *média(md)*, e *contraste(ct)*. Se a escolha não puder ser efectuada condicionalmente por d_5 (para o nó cinco) ou por d_6 (para o nó seis), o processo residual será desencadeado.

3 - O processo residual utiliza uma classificação baseada em mínima distância, pela utilização dum vector w de características médias. As atribuições serão feitas às classes c_1, c_4, c_5 ou c_6 (*SDF, NÓS, BRN e BCH*) através de d_7.

4 - O contorno do defeito é representado no écran juntamente com a classe a que pertence. Se a classe pertencer a *SDF* o contorno não é mostrado no écran e o defeito não é assinalado.

O diagrama de fluxos do algoritmo pode ser esquematizado do seguinte modo:

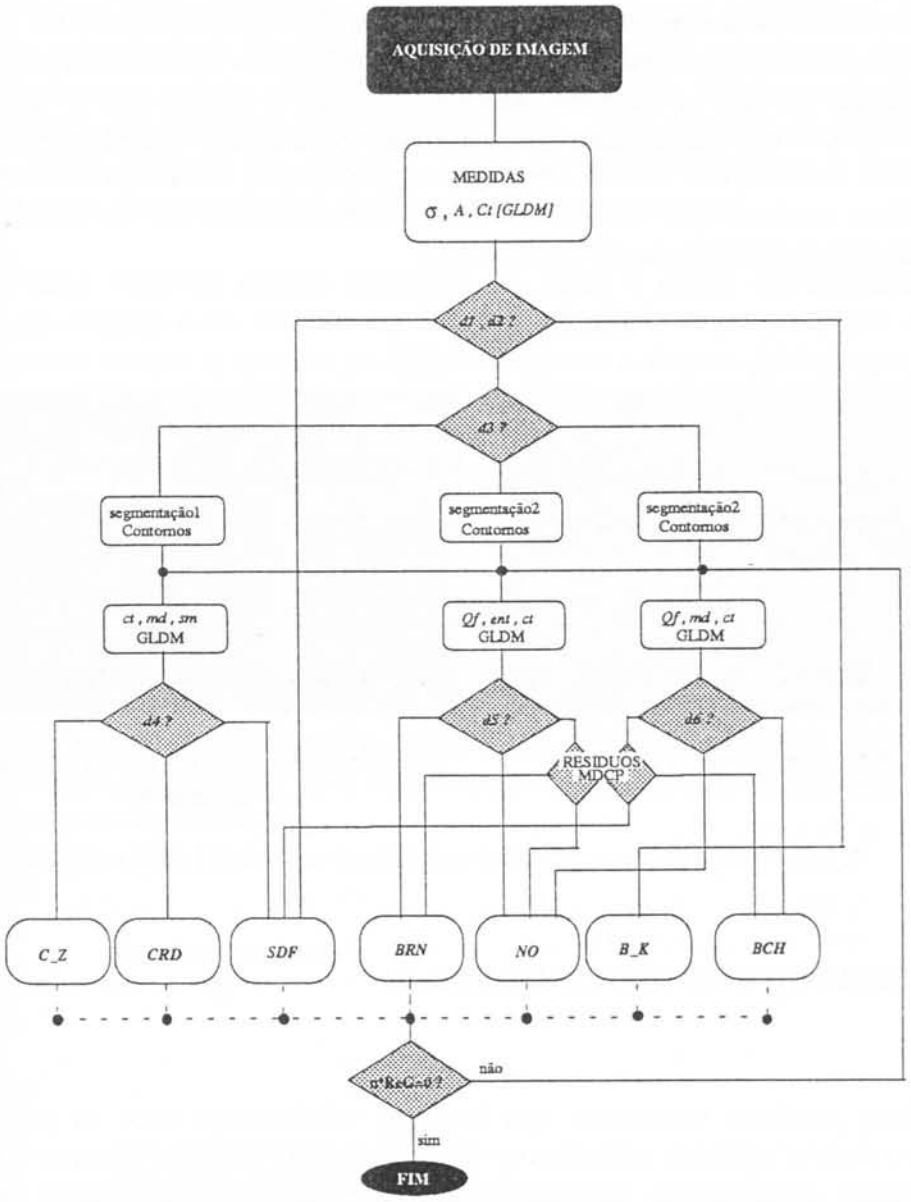


fig. 7.7 Diagrama de fluxos do programa de imagem *rdmp/ad-hoc*

As imagens apresentadas nas figuras 7.8 e 7.9 representam a imagem e o resultado da classificação obtida, pela aplicação do programa de reconhecimento *rdmp/ad-hoc*. Nota-se que apesar de serem reconhecidos correctamente os defeitos do tipo *nós*, aparecem na imagem subregiões de menor dimensão mal classificadas

(classificadas como *brechas de resinagem-BCH*). Uma forma possível para ultrapassar este problema, seria tentar classificar apenas as regiões de segmentação, de dimensão razoável, uma vez que a informação sobre a área das regiões segmentadas é conhecida.

Nem sempre a classificação correcta das imagens (de superfícies naturais como no caso das madeiras) é fácil de obter. Uma das muitas dificuldades que se deparam num processo de classificação correcta dos defeitos em madeira, prende-se com a variância elevada do valor das características para o mesmo tipo de defeito, e consequentemente nas dificuldades em caracterizar os defeitos com rigor. É quase sempre muito difícil, senão mesmo impossível nalguns casos, obter essa consistência. Pode-se dizer contudo que dentro das dificuldades relativas, são os nós os que apresentam menos dificuldades de classificação.

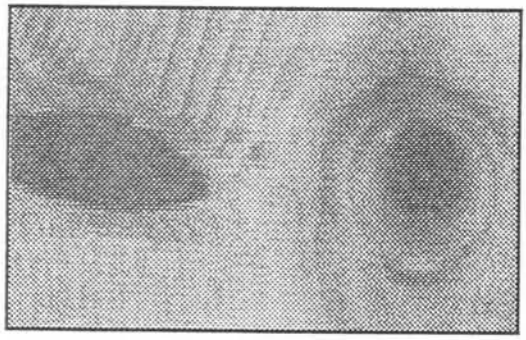


fig. 7.8 Imagem de pinho com nós

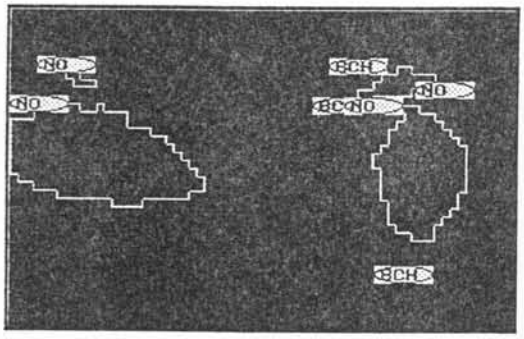


fig7.9 Classificação obtida da fig. 7.8, por *RDMP/ad-hoc*

7.3-4 ESTRUTURAS DE DADOS UTILIZADAS

Neste subtítulo trataremos dos assuntos relacionados com as principais estruturas e com as variáveis utilizadas no sistema *rdmp/ad-hoc*. Em todas as funções implementadas, nesse sistema *rdmp/ad-hoc* para o *Pinhobravo* e no sistema mínimo de imagem, são utilizados três *"buffers"* de imagem, que correspondem à utilização de três vectores de ponteiros **buf1[200]*, **buf2[200]*, **buf3[200]*, que apontam para regiões de memória alocada dinamicamente. Todos estes ponteiros apontam para dados do tipo *"unsigned char"*, correspondendo cada unidade a um pixel de imagem. Globalmente cada região alocada tem capacidade para armazenar uma imagem de 320x200 (nº colunas x nº linhas) pixels. À excepção do **buf1[200]*, que armazena a imagem original da operação, com *display* na região **'1'** do écran (o monitor foi dividido em duas regiões para display de imagens de madeira, e uma para troca de informação), e que tem o espaço de imagem alocado permanentemente na

memória enquanto o programa estiver activo, os vectores de ponteiros **buf2[200]* e **buf3[200]*, têm memória alocada em função das necessidades do sistema (para armazenar imagens ou operações de imagem intermédias). Estes ponteiros apontam para NULL, caso não haja alocação de memória atribuída. A memória é libertada em qualquer altura da execução do programa, sempre que a informação contida por estes *"buffers"* já não seja necessária. Isto representa uma vantagem fundamental, já que as operações sobre imagem, requerem manipulação de grandes quantidades de dados e consequentemente espaço para os armazenar, o que põe problemas de memória, principalmente em ambientes limitados de trabalho, como é o caso de operações em MS-DOS. Estes *"buffers"* são globais e podem ser acedidos (criados e libertos) de qualquer ponto do programa (ou das funções que o componham).

Outras variáveis globais utilizadas são *maxx* e *maxy*, correspondentes ao número de colunas e ao número de linhas no *"buffer"* de imagem, que no caso correspondem sempre a imagens de 320x200(colunas x linhas). Inicialmente previa-se, que o sistema pudesse suportar imagens de qualquer formato (qualquer número de linhas e colunas, dentro dos limites do *sistema gráfico*).

A variável global *chext* indica se o sistema está a operar em dezasseis cores ou em 256 níveis de cinzento. É de salientar que o *display* das imagens é sempre feito em dezasseis cores (ou dezasseis níveis de cinzento), impostos pelas limitações do *drive gráfico* EGAVGA.BGI em que o sistema foi implementado, apesar das imagens estarem gravadas e serem processadas em 256 níveis de cinzento. A variável global *iM_g* indica ao sistema, se o *"buffer"* três está ou não com memória alocada.

Apesar do sistema utilizar memória dinâmica em muitas das funções que utiliza, vamos fazer apenas uma síntese das variáveis utilizadas pela função segct1(...)[6](por ser uma das mais significativas e das mais utilizadas) e que tem a

---

[6]Esta função utiliza listas ligadas com as estruturas indicadas a seguir, lista1 e lstpc1

```
struct lista1
        {
                struct lista1 *pximo;
                int       i,j;
        };

struct lstpc1
        {
                struct lstpc1   *prox;
                struct lista1   *proximo;
                unsigned char   flag0:1,flag1:3,flag2:4;
                unsigned char   ai,bi;
                int             area,prt,num,aj,bj;
        } ;
```

e utiliza várias funções de suporte para editar, inserir e apagar estas estruturas. Essas funções estão representadas em baixo.

```
struct lista1 *nova1(...) /*função que retorna uma struct*/
struct lista1 *insere1(struct lista1 *pt)
struct lstpc1 *nov1(...) /*função que retorna uma struct*/
struct lstpc1 *isre1(struct lstpc1 *pt)
```

finalidade de estruturar o contorno de um segmento de imagem, armazenando numa lista ligada todos os segmentos que constituem esse contorno, além de um conjunto de grandezas geométricas do contorno, ou da superfície desse contorno. Esta função recebe como argumentos, a indicação do *"buffer"* onde a imagem está armazenada, e as coordenadas *(j,i)* do ponto de arranque desse contorno. Esta função procura todo o contorno e retorna um ponteiro para o inicio de uma lista ligada com a estrutura mostrada na fig. 7.10.

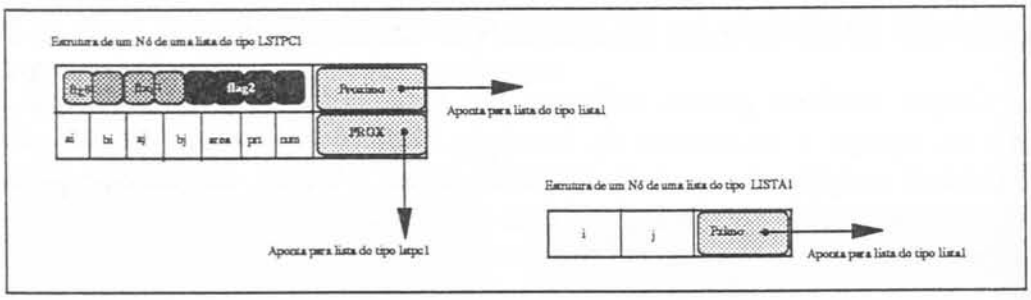


fig. 7.10 Estrutura dos nós das listas ligadas: *lstpc1* e *lista1*

A função segct1(...) retorna também um vector com as frequências de ocorrência dos códigos de cadeia (*"chain code"*), que ocorrem no contorno. A função fill1(...) complementa o processo de exploração dos contornos das regiões segmentadas, enchendo o contorno a que respeita com uma cor que o identifique e distinga dos restantes.

Uma função que utiliza recursividade, é utilizada na homogeneização de uma imagem, através da subdivisão sucessiva da imagem, até que se encontre essa região homogénea não divisível. A função arv_q2(...)[7], constrói esses dados e armazena os dados numa estrutura do tipo árvore.

---

struct lista1 *apglista1(struct lista1 *ppt)
struct lstpc1 *apglst1(struct lstpc1 *pt)
unsigned int fill1(struct lista1 *llig,int num,char buff_er,unsigned int *ar,int *prt,int lmt[4])
struct lista1 *segct1(char buff_er,int i,int j,int *n,int chncd[8])-função objecto
Todas estas funções se encontram no arquivo TBINCT.C

[7]Consultar capítulo 3, parágrafo 3.6 e a nota 4 no capítulo 7. A estrutura do nó da àrvore é a seguinte:

```
struct quadt_e
{
        unsigned char md,nivel,coordyi;
        int coordxj;
        struct quadt_e  *esqlt,*esqbx,*drtlt,*drtbx;
}
```

O processo é recursivo, já que o mecanismo de construção da árvore é repetitivo e baseia-se sempre na subdivisão da árvore em quatro partes, estabelecendo-se sempre uma hierarquia em relação ao nó que está a ser tratado. Cada nó da estrutura da árvore armazena o centróide (ponto no centro da sub imagem a que pertence) o nível de profundidade da árvore e a média da subregião da imagem. Para o caso de um nó terminal (folha) a imagem pode ser reconstruída à custa do centróide e da profundidade da árvore para esse nó, recorrendo a uma tabela que indica para cada nível de profundidade a dimensão do rectângulo utilizado nas subdivisões quaternárias utilizadas nesse nível.

Este algoritmo exige preocupações especiais a nível do *"stack"*, pelo facto da função ser recursiva, e exigir tanto mais *stack* quanto mais vezes esta for chamada durante a construção da árvore. Para precaver contra o *'estoiro'* do *"stack"*, além de se lhe ter reservado mais espaço, impôs-se uma condição limite, que impede a árvore de crescer para além de um determinado nível, impondo-se também um limite para o número de variáveis locais, que foram reduzidas ao minimo estritamente indispensável.

7.4 Selecção e Extracção de Características

7.4.1 Introdução

Num processo de reconhecimento de padrões, a selecção e extracção de características[33-A], está relacionado com o problema da redução da dimensão do vector de características que representa as diferentes classes de padrões. Essa necessidade impõe-se pelo facto, da complexidade do classificador ou do *"hardware"* que o implementa, crescer rapidamente com a dimensão do espaço de características. É necessário por isso, basear a decisão apenas no número suficiente de características, que sejam essenciais para descriminar a informação.

O desenho de um sistema supervisado de reconhecimento, está baseado num número finito de padrões, geralmente pequeno, de amostras de treino. A redução da dimensão do vector de padrões, pode fazer-se à luz do seguinte princípio: efectua-se a classificação com um vector de padrões de dimensão reduzida: vão-se juntando em seguida, uma a uma, novas características ao vector que representa a classe, até ao ponto em que a junção de uma nova característica, em vez de melhorar o resultado, o piore. A redução da dimensão do vector de características pode inclusivamente diminuir a taxa de erros de classificação.

A estrutura do nó está representado esquemáticamente na fig3-17 (cap. 3)

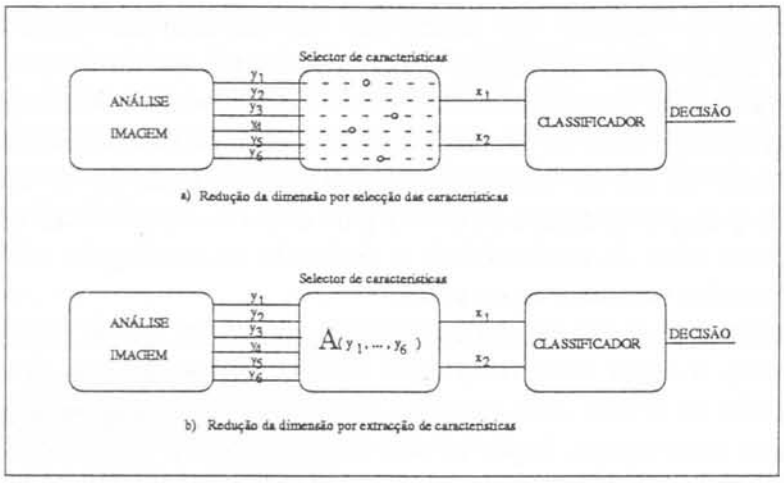


fig. 7.11 Processos de redução da dimensão dum vector de características representativo duma classe de padrões

Se para um conjunto de características, as diversas classes de padrões se sobreposerem, é necessário acrescentar novas características ao espaço de padrões, para as poder tornar separáveis as classes.

A redução da dimensão do espaço de padrões pode ser conseguida de dois modos: um dos processos identifica as características que não contribuem para a separabilidade das classes (selecção de características fig. 7. 11a)). Esse problema é posto da seguinte forma:

> Selecciona-se de um conjunto Y_k de características, com $k=1, 2, \ldots, k$ um conjunto x_j de características, com $j= 1, 2, \ldots, d$ em que d seja menor que k.

O outro processo (extracção de características), consiste em transformar um conjunto de características, através de um processo de projecção, noutro espaço de características de menor dimensão (fig. 7. 11b)).

Apenas será abordado o problema da selecção de características, e dentro desse, apenas algumas das técnicas que nos pareceram ajustar-se melhor ao tipo, dimensão e natureza do problema a tratar.

7.4.2 ORDENAÇÃO DE CARACTERÍSTICAS, ATRAVÉS DAS SUAS PROPRIEDADES INTRÍNSECAS DE AGLOMERAÇÃO

As características não têm todas o mesmo poder para descriminar classes de padrões. Através de processos descritos por Tou e Gonzalez [32-A], Sing-Tze Bow[9-A], etc., podem ser atribuídos coeficientes pesados às características com esse significado. O valor desses coeficientes têm a particularidade de apresentar um valor decrescente à medida que o seu significado diminui para a classe considerada. A ordenação baseia-se na comparação relativa da variância interna de uma característica (dentro de uma mesma classe, para um conjunto de amostras) dada, em função da variância interna das outras características para a mesma classe. A transformação matricial usada com esse significado é dada por:

$$w = w_{jk} \qquad w_{jk} = \begin{cases} 0 & quando\ j \neq k \\ w_{jj} & quando\ j = k \end{cases} \tag{7.3}$$

onde w_{jj} representam os coeficientes pesados das características, com j = 1, 2,. . . ,n e n representa o número de características

Procura-se determinar os coeficientes w_{jj}, de tal modo que sejam encontrados bons aglomerados para a classe. Isso pode ser obtido através da minimização da distância entre os padrões pertencentes a uma mesma classe.

A medida de distância entre os padrões pode ser determinada por:

$$\overline{D}^2 = 2\sum_{j=1}^{n} (w_{jj}\sigma_j)^2 \tag{7.4}$$

onde σ^2 é a variância da característica ao longo dum eixo (por ex: dos x_{js}).

Os multiplicadores de Lagrange podem ser utilizadosna determinação das variáveis características que minimizam a distância entre os

elementos dum mesmo conjunto sujeito a dois tipos de restrições. Essas restrições permitem considerar duas situações distintas:

$$\text{Caso 1:} \sum_{j=1}^{n} w_{jj} = 1 \text{(soma dos pesos igual a 1)} \tag{7.5}$$

a minimização de (7.4) é equivalente à minimização da expressão seguinte (método dos múltiplicadores de Lagrange):

$$S = 2\sum_{j=1}^{n} (w_{jj}\sigma_j)^2 - \rho_1\left(\sum_{j=1}^{n} w_{jj} - 1\right) \tag{7.6}$$

derivando S em ordem a w_{jj} e igualando a zero, obtém-se um minimo para

$$w_{jj} = \frac{\rho_1}{4\sigma_j^2} \tag{7.7}$$

Através de 7.5 e de 7.7, vem

$$\sum_{j=1}^{n} \frac{\rho_1}{4\sigma_j^2} = 1 \tag{7.8}$$

retirando o valor de ρ_1 de (7.8) e substituindo em (7.7), obtém-se o valor de w_{jj}

$$w_{jj} = \frac{1}{\sigma_j^2 \sum_{j=1}^{n} \sigma_j^{-2}} \qquad \text{em que } w_{jj} \text{ varia na razão inversa de } \sigma_j^2 \tag{7.9}$$

$$\text{Caso 2:} \prod_{j=1}^{n} w_{jj} = 1 \text{ (produto dos pesos igual a 1)} \tag{7.10}$$

Seguindo o mesmo raciocínio chega-se à conclusão que

$$w_{jj} = \frac{1}{\sigma_j}\left(\prod_{j=1}^{n}\sigma_j\right)^{1/n} \quad \text{em que } w_{jj} \text{ varia na razão inversa de } \sigma_j \qquad (7.11)$$

Conclui-se que os pesos maiores são atribuídos às características com desvios quadráticos menores, que são aquelas que apresentam maior consistência. Daí resulta que as características com maior peso w_{jj}, são as que apresentam maiores potencialidades relativas para poder separar a classe avaliada das restantes, numa situação de classificação.

7.4.3 CÁLCULO DO FACTOR DE ORDENAÇÃO, EM FUNÇÃO DO VALOR DA DISPERSÃO DAS CARACTERÍSTICAS

A escolha das características envolve dois aspectos muito importantes e complementares. Em primeiro lugar importa que num processo de classificação, a escolha das características, obedeça a critérios que contemplem essa escolha com a menor dispersão possível, em relação aos seus valores médios, de modo a evitar comportamentos erráticos do classificador. Em segundo lugar importa que essas características apresentem a maior separação possível, de modo que o classificador possa ser facilmente desenhado, garantindo a separação das classes com segurança.

O quadro 7.3, foi calculada com base na fórmula (7.9) deduzida no ponto7.4.2 deste capítulo e mostra o factor de ordenação calculado para as características(*'0'* a *'42'*, descritas no Anexo **2**) em função das suas pontencialidades ou poder de separação, entre cada uma das oito classes em que foi dividido o problema de classificação dos defeitos em pranchas de madeira, e as restantes. Estes factores de ordenação, procuram resolver o primeiro problema colocado no ponto anterior, já que são uma medida do grau de aglomeração de cada característica, para cada uma das classes.

Constata-se pela observação da tabela, que existem classes em que algumas das características aparecem bastante concentradas, casos por exemplo da característica *'0'* para *b_k*, da *'9'* para as *brn* ou da *'26'* para as *bch*, só para citar algumas. Outras classes mostram contudo, que não se apresenta qualquer característica em condições aceitáveis de aglomeração, para poder fazer parte de um vector de características com base na qual se possa desenhar um hiperplano de decisão. Isso acontece, poe exemplo, no caso do defeito *'Cardido'*, o que coloca por esse motivo, sérias dificuldades para o separar das restantes classes.

clct/clas	0	1	2	3	4	5	6	7
0	0.46	0.11	0.06	0.09	0.14	0.05	0.07	0.02
1	0.02	0.01	0.07	0.03	0.07	0.04	0.04	0.71
2	0.16	0.2	0.07	0.08	0.23	0.09	0.09	0.08
3	0.24	0.14	0.06	0.15	0.05	0.15	0.09	0.12
4	0.21	0.19	0.05	0.15	0.05	0.18	0.05	0.11
5	0.07	0.1	0.2	0.1	0.13	0.19	0.12	0.1
6	0.06	0.07	0.09	0.08	0.33	0.14	0.16	0.06
7	0.05	0.14	0.08	0.18	0.26	0.09	0.08	0.12
8	0	0.02	0.04	0.05	0.75	0.02	0.11	0
9	0	0.01	0.02	0.03	0.91	0.01	0.02	0
10	0.06	0.11	0.2	0.17	0.1	0.05	0.12	0.19
11	0.01	0.18	0.03	0.3	0.05	0.03	0.16	0.25
12	0	0.05	0.02	0.07	0.34	0.4	0.02	0.09
13	0.03	0.09	0.04	0.16	0.2	0.32	0.05	0.11
14	0.01	0.03	0.06	0.04	0.34	0.11	0.15	0.26
15	0.31	0.09	0.16	0.1	0.09	0.04	0.09	0.12
16	0.23	0.05	0.32	0.11	0.07	0.01	0.06	0.14
17	0.04	0.15	0.03	0.08	0.17	0.45	0.04	0.04
18	0.1	0.11	0.07	0.11	0.15	0.32	0.06	0.08
19	0.08	0.07	0.03	0.06	0.19	0.4	0.12	0.04
20	0.02	0.06	0.05	0.11	0.36	0.18	0.1	0.14
21	0	0	0	0	0.46	0.27	0.13	0.12
22	0.08	0.07	0.06	0.13	0.27	0.15	0.09	0.15
23	0	0.01	0.02	0.06	0.24	0.62	0.03	0.01
24	0.02	0.03	0.04	0.06	0.39	0.19	0.12	0.16
25	0.03	0.04	0.03	0.06	0.11	0.63	0.08	0.02
26	0	0.02	0.01	0.03	0.12	0.74	0.06	0.02
27	0.09	0.06	0.05	0.06	0.1	0.51	0.08	0.04
28	0	0.01	0.02	0.05	0.16	0.73	0.02	0.01
29	0.09	0.06	0.04	0.07	0.12	0.48	0.09	0.04
30	0.02	0.2	0.02	0.01	0.33	0.38	0.01	0.03
31	0.12	0.12	0.06	0.2	0.28	0.06	0.09	0.07
32	0.04	0.11	0.03	0.3	0.35	0.05	0.07	0.04
33	0.02	0.2	0.06	0.04	0.29	0.02	0.08	0.3
34	0.22	0.08	0.05	0.11	0.33	0.07	0.07	0.07
35	0	0.02	0	0.86	0.02	0.01	0.06	0.02
36	0	0.11	0.08	0.03	0.22	0.41	0.02	0.13
37	0.4	0.11	0.03	0.19	0.12	0.02	0.07	0.06
38	0.18	0.17	0.02	0.37	0.07	0.01	0.12	0.05
39	0.09	0.49	0.05	0.05	0.13	0.05	0.02	0.13
40	0	0	0	0.93	0.01	0	0.05	0.01
41	0	0.54	0.06	0.01	0.19	0.05	0	0.15

Quadro 7-3 Factor de ordenação des características em função da aglomeração, por classe

classe 0 - backup(B_K1)
classe 1 - Casca(Casca)
classe 2 - Sem defeito(SDF)
classe 3 - Nós(Nos)
classe 4 - Bolsa de Resina(BRN)
classe 5 - Brechas de Resinagem(BCH)
classe 6 - Cardido(PCD/CRD)
Azulado 7 - Azulado da Madeira(AZULAD)

As características referenciadas na coluna um do Quadro 7.3, estão descritas na pag1 do Anexo2.

7.4.4 SEPARABILIDADE ENTRE CLASSES POR MEDIDAS DE DISTÂNCIA

Alguns critérios de selecção de características baseiam-se em processos heurísticos. É isso que acontece na determinação da separabilidade entre classes por utilização de diversas medidas de distância[33-A].

Assim se determinar a distância média entre padrões pertencentes a diferentes classes, duas a duas, fico com uma medida da separabilidade entre essas duas classes de padrões. Representando por $\delta(\xi_{ik}, \xi_{jl})$ a distância entre o padrão k da classe i e o padrão l da classe j, a distância média para as classes pode ser calculada por

$$j_\delta = \frac{1}{2}\sum_{i=1}^{m} P(w_i)\sum_{j=1}^{m} P(wj)\frac{1}{N_i N_j}\sum_{k=1}^{N_i}\sum_{l=1}^{N_j}\delta(\xi_{ik},\xi_{j,l}) \qquad (7.11)$$

onde N_i é o número de padrões pertencentes à classe w_i e N_j é o número de padrões pertencentes à classe w_j.

As métricas utilizadas são de vários tipos, sendo as mais usuais as que estão representadas em (7.13), através das expressões (1) a (6).

$$\delta_C(\xi_k,\xi_l) = \sum_{j=1}^{d}\left|\xi_{kj}-\xi_{lj}\right| \qquad \text{"city block"} \qquad (1)$$

$$\delta_E(\xi_k,\xi_l) = \left[\sum_{j=1}^{d}(\xi_{kj}-\xi_{lj})^2\right]^{1/2} \qquad \textit{Euclideana} \qquad (2)$$

$$\delta_T(\xi_k,\xi_l) = \max_j\left|\xi_{kj}-\xi_{lj}\right| \qquad \textit{Chebyshev} \qquad (3)$$

$$\delta_Q(\xi_k,\xi_l) = (\xi_k-\xi_l)^T Q(\xi_{kj}-\xi_{lj}) \qquad \textit{Quadrática} \qquad (4)$$

(7.13)

em 7.13.6 Q representa uma matriz definida positiva e simétrica.

A escolha do critério de medida, depende de vários factores, tais como factores de complexidade computacional, tractibilidade analítica, segurança, etc.

As medidas calculadas através dos métodos de *"City block"* e *"Chebyshev"*, são das mais fáceis de calcular. Estes critérios têm a vantagem de não necessitar da determinação das funções de densidade de probabilidades das características por classe.

Outro processo utiliza as equações (7.14.5) a (7.14.8) como medidas de distância, que são medidas paramétricas retiradas do critério J_δ , através do uso da métrica Euclideana.

$$\Sigma = \sum_{i=1}^{m} P(w_i)\Sigma_i$$

$$\mathrm{M} = \sum_{i=1}^{m} P(w_i)(\mu_i - \mu)(\mu_i - \mu)^T$$

$$\mu = \sum_{i=1}^{m} P(w_i)\mu_i \qquad \textit{em que } \mu \textit{ é a mistura das médias}$$

$$J_1 = Tr(\Sigma + \mathrm{M}) \qquad (5)$$

$$J_2 = Tr(\mathrm{M}/\Sigma) \qquad (6)$$

$$J_3 = Tr(\Sigma^{-1}\mathrm{M}) \qquad (7)$$

$$J_4 = Tr|\Sigma + \mathrm{M}|/|\Sigma| \qquad (8)$$

(7.14)

7.4-5 Separação Numérica de Classes Método Jeffries-Matusita (J-M)

7.4.5-1 Descrição

Só é possível obter uma boa classificação, das classes envolvidas num processo de reconhecimento, se as classes estiverem numericamente separadas, por um intervalo perfeitamente identificável no espaço de características. Se existir para essas classes uma sobreposição das funções densidade de probabilidades (de uma ou mais variáveis), isso implicará um aumento de incerteza na classificação, em função dessa sobreposição.

Para se fazer uma boa classificação é necessário encontrar as melhores características. Para tal é necessário uma grandeza que meça a separação entre as classes, em função dessas características. Uma tal medida pode ser encontrada pelo método de Jeffries-Matusita(JM)[34-A][8], que estabelece uma medida de distância entre as classes, consideradas duas a duas, em função de um conjunto variáveis características representativas, num conjunto de padrões pertencentes às classes.

A distância entre a classe i e a classe j é definida por:

$$\Im_{ij} = \left(\int_v \left(\sqrt{q(v|i)} - \sqrt{q(v|j)} \right)^2 dv \right)^{1/2} \tag{7.15}$$

Para o caso de distribuições de normais de multivariável, a função distância de JM, pode ser calculada por:

$$\Im_{ij} = \sqrt{\left[2(1 - e^{-a}) \right]} \tag{7.16}$$

onde

$$a = \frac{1}{8}(m_i - m_j)^T \left(\frac{\Sigma_i + \Sigma_j}{2} \right)^{-1} (m_i - m_j) + \frac{1}{2} \log_e \left(\frac{|(\Sigma_i + \Sigma_j)/2|}{\sqrt{(|\Sigma_i||\Sigma_j|)}} \right) \tag{7.17}$$

Esta fórmula é avaliada para cada par de classes i e j, representando m_i, m_j o vector de médias e Σ_i, Σ_j as matrizes de covariância. Através do valor de τ_{ij}, é fornecida uma indicação da forma como as classes estão separadas. τ_{ij} só nos dá a medida de separação entre duas classes, mas em função de vectores de qualquer número de características.

Se pretendermos apenas uma medida, que indique o grau de separação de todas as classes, podemos utilizar a expressão (7.18) que é uma medida da média dos pares de classes.

$$\Im_{ave} = \sum_i \sum_j q(i) q(j) \Im_{ij} \tag{7.18}$$

[8]Processo referido por Wayne Niblack, em "Digital image processing"[34-A], cap VII, pp177s

Na expressão (7.18) a probabilidade à priori *q(i)* favorece as classes com maior probabilidade de ocorrência.

A distância JM pode ser usada com dois objectivos diferentes: primeiro, fornecendo uma indicação da distância entre as classes, para as características testadas (características com as quais foi determinado τ_{ij}), podendo ser esperada a melhor classificação com aquelas, a partir das quais se obtiverem os valores mais elevados para o factor; segundo dando uma indicação sobre o número de características a introduzir no vector de classificação, indicação obtida através da combinação de característica que forneçam a maior separação.

O método JM é apenas um dos métodos que é possível usar para medir a separação entre duas classes, como já foi referido, e que se enquadra num campo de estudo vasto, chamado *selecção de características*.

7.4.5-2 Separação inter-classes. Cálculo do factor de distancia JM, em defeitos de madeira

Considerando um conjunto de oito classes para a madeira de pinho: *backup(b_k)*, *casca(csc)*, *madeira escura sem defeito(sdf)*, *nós(no)*, *bolsas de resina(brn)*, *brechas de resinagem(bch)*, *cardidos(crd)* e *azulados(azld)*, foram retiradas diversas medidas de distância inter-classes (duas a duas) pelo método JM. Essas medidas estão compiladas no Anexo **6** e referem-se a diversas combinações de características, referenciadas através do número e seguindo que seguem a nomenclatura de características para defeitos individuais, utilizadas no Anexo **2**. Os quadros 5, 6 e 7 são exemplos retirados ao acaso, para efeitos de ilustração, desse Anexo **6**.

Dado o valor combinatório de todas as variáveis características ser elevado (combinações de quarenta e duas variáveis duas a duas e três a três), optou-se pelo teste das combinações que ofereciam à partida, alguma garantia de poderem separar algumas classes. Foi com base nos valores encontrados, para os factores de aglomeração das variáveis características por classe (apresentados no Quadro 7.4), que se procuram os melhores valores esperados para a separação inter-classes, dadas pelo factor de distância τ_i de JM, em função de combinações de duas e três variáveis. O Anexo **6**, apresenta precisamente os coeficientes de distância inter-classe (τ_i)[9] duas a duas, obtidos pelo método JM, em função de duas e três variáveis.

[9] Os coeficientes de distância inter-classe(τ_i) JM, foram obtidos através da função FcUfMt(...) em TECT.C e os dados lançados em FJM.TXT.

Apresenta-se aqui a título ilustrativo, dois exemplos de separação entre classes, para dois conjuntos de três variáveis, oferecida pelos coeficientes τ_i de JM como mostram as tabelas 7.4 a 7.6. Estes quadros evidenciam até que ponto, os três conjuntos de variáveis têm hipótese de separar bem a classe BRN das restantes. Pode observar-se seguindo as respectivas linha e coluna, que as variáveis da tabela 7.6, podem separar melhor das restantes classes, as *brn* do que as variáveis da tabela 7.5, já que os coeficientes respectivos apresentam valores maiores.

3 características	9	21	30				
b_k	1						
casca	1.38	1					
sdf	1.37	0.83	1				
nos	1.38	1.02	0.54	1			
brn	1.41	1.27	1.33	1.34	1		
bch	1.4	1.21	1.25	1.31	1.24	1	
card	1.4	1.2	1.02	0.97	1.25	1.18	1
azld	1.37	1.24	1.15	1.21	1.36	1.18	1.06	1
	b_k	csc	sdf	nos	brn	bch	crd	azld

Quadro7.4 Coeficientes de distância inter-classe(τ_i), para as variáveis 9, 21 e 30

3 características	14	20	32				
b_k	1						
casca	1.4	1					
sdf	1.39	0.81	1				
nos	1.41	0.83	0.96	1			
brn	1.41	1.03	1.13	1.17	1		
bch	1.41	0.72	0.73	1.03	0.85	1	
crd	1.41	0.76	0.61	0.92	0.82	0.49	1
azld	1.4	0.94	0.65	0.94	1.21	0.92	0.74	1
	b_k	csc	sdf	nos	brn	bch	crd	azld

Quadro 7.5 Coeficientes de distância inter-classe(τ_i), para as variáveis 14, 20 e 32

Outro exemplo é apresentado na tabela 7.6, em que as variáveis testadas têm como objectivo, verificar se a separação por distância entre as classes, *SDF* e as classes *CASCA*, *AZLD* e *CRD* é possível, através desse conjunto de variáveis características. Saliente-se que num processo de *Decisão Binária Hierarquizada*, analisado em 7.7, são as classes a jusante do nó da árvore de decisão que interessa separar através do classificador. Interessa ainda saber até que ponto um conjunto determinado de características, tem possibilidade de o fazer (separar essas classes).

3 características	5	10	16				
b_k	1						
casca	1.17	1					
sdf	1.24	1.03	1				
nos	1.04	0.7	1.05	1			
brn	1.38	1.05	1.33	1.27	1		
bch	1.35	0.99	1.37	1.17	1.07	1	
card	1.28	0.71	0.91	0.93	1.13	1.2	1
azld	1.02	0.91	0.81	0.59	1.32	1.28	0.95	1
	b_k	csc	sdf	nos	brn	bch	crd	azld

Quadro 7.6 Coeficientes de distância inter-classe(τ_i), para as variáveis 5, 10 e 16

Muitos outros exemplos poderiam ser dados, e através deles constatar-se por exemplo, que o conjunto de variáveis *35,38,40* oferece mais garantias, para poder separar a classe *nós* das restantes, do que aquelas que ofereceriam as características *11,32,35*, mesmo sabendo que à partida todas elas tinham potencialidades para o poder fazer.

O Anexo **6** apresenta os valores encontrados para τ_i. A partir deles, podem ser tiradas conclusões sobre as variáveis mais convenientes a utilizar na separação das classes/defeitos na madeira, e mais concretamente em cada nó do processo de decisão hierárquica descrita em 7.7 deste capítulo.

7.5 Processo de reconhecimento de defeitos em madeira, por decisão singular de etapa única

Este processo envolve uma tomada de decisão, que procura resolver o seguinte problema. A que classe deve ser atribuída uma dada *região/defeito*, das várias em que foi segmentada a peça de madeira, para que se possa fazer um mapa das diversas regiões defeituosas presentes? As oito classes focadas no problema são novamente as já apresentadas no ponto 7.5.2. A escolha faz-se directamente, entre uma das cinco opções possíveis: *casca(csc)* mais *nós(no)*, *backup(b_k)*, *cardidos(crd)* mais *azulados(azld)*, *madeira escura sem defeito(sdf)* e *bolsas de resina(brn)* mais *brechas de resinagem(bch)*.

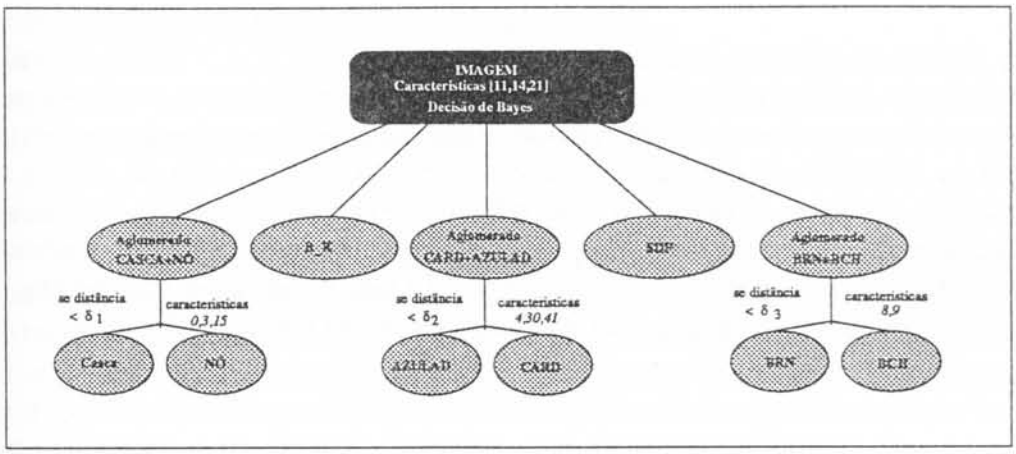


fig. 7.12 Esquema de classificação por decisão singular, com base nas características e funções de decisão indicadas

Com base no poder separador entre as classes que, que as características manifestam, calculado através dos coeficientes de distância de Jeffries-Matusita(JM), foi escolhida a melhor combinação possível para fazer a referida separação. O processo que foi implementado, está esquematizado na fig.7.12, e referência não só o esquema de decisão, mas também as variáveis escolhidas (referenciadas numericamente de acordo com o Anexo **2**) e as classes separadas.

O processo é frágil, pelo facto de se estar a determinar a menor probabilidade de erro entre cinco grupos/classes possíveis, numa só etapa. Qualquer erro na decisão, implica uma escolha mal feita.

Dos cinco grupos/classes, três são complexos e por isso são posteriormente separados em nós secundários. As funções de decisão utilizadas nos nós respectivos, baseiam-se nas características referenciadas na figura e em processos de decisão de Bayes para o nó primário e em funções de mínima distância para os nós secundários.

Resumindo, no nó primário foram utilizadas as características *11, 14* e *21*, com significados definidas no Anexo **2** e com a finalidade de separar os primeiros cinco grupos/classes (duas classes e três aglomerados), com significados seguintes:

1-Aglomerado de *casca+Nó*
2-classe *B_K*
3-Aglomerado de *Cardido+Azulado*
4-classe *SDF*
5-Aglomerado de *Brn+Bch*

característica [*11*] - *contraste*($d=1;\theta = 0^o$)
característica [*14*] - *media inversa*
característica [*21*] - *LRE(cadeias longas)*

Depois da primeira decisão e para os aglomerados *'1' '3' e '5'* foram utilizadas nas funções de decisão, as características que apresentavam melhores condições para separar as classes presentes no aglomerado. Essas características não são as mesmas do que as utilizadas na função de decisão inicial. A estruturação da árvore de classificação obedeceu a valores de separabilidade encontrados através do cálculo dos coeficientes de J. Matusita (JM). A escolha doutras características implicaria necessariamente a separação das classes segundo condições diferentes, e consequentemente uma estrutura e resultados de separação também diferentes.

Defeitos	*B_K*	*CASCA*	*SDF*	*NÓS*	*BRN*	*BCH*	*CRD*	*AZLD*
% sucesso	84%	22%	32%	72%	56%	58%	32%	33%

Quadro 7.7 Taxas de sucesso de classificação correcta pelo processo decisão singular

Apesar dos factores de indicação de distância JM, apresentarem em função das características seleccionadas um grau de separabilidade entre as classes razoável, o facto é que as taxas de classificação correcta encontrados com estas características, por este processo, não se revelaram muito encorejadores, tal como se pode observar pela tabela 7.8.

Algumas das razões que podem explicar o sucedido, prendem-se com o facto da distribuição de probabilidades das características por classe, não apresentar um comportamento normal (Gaussiano), pressuposto em que assenta o cálculo da probabilidade que determina se um padrão pertence ou não a uma classe, e no facto das variáveis características das amostras, apresentarem uma dispersão muito grande, em relação ao que foi considerado representativo para cada um dos grupos de classes, que foram caracterizadas no problema.

È provável que a escolha doutro conjunto de características pudesse melhorar o rendimento do classificador, em termos de taxa de êxito de classificação. Podemos contudo afirmar sem correr grandes riscos, que a melhoria nunca seria muito significativa, já que a dificuldade essencial permanece: separar cinco grupos de classes na mesma etapa.

Face a essa conclusão, foi preferível enveredar por um processo de classificação hierárquica, já que desse modo a dimensão do problema é artificialmente diminuida, através da selecção de uma classe apenas, em cada nó de decisão da árvore hierárquica. O adiamento das decisões difíceis (ou mais difíceis) para a etapa seguinte de decisão, ou em última análise para o fim do problema, permite diminuir a probabilidade de erro de classificação, através da diminuição das opções possíveis em cada nó de decisão.

7.6 PROCESSO DE DECISÃO BINÁRIA GLOBAL

7.6.1 DESCRIÇÃO DO PROCESSO

Um ensaio destinado à identificação de duas classes, correspondentes à divisão do problema de classificação da madeira em pranchas de pinho; *sem defeito(SDF)* e *com defeito(CDF)*, é descrito neste subtítulo. O objectivo prende-se com a procura de formas destinadas à divisão do problema em subproblemas de menor dimensão, com possibilidades de resolução maior. Saliente-se que se for possível identificar em primeiro lugar, se uma determinada prancha possui ou não defeitos, é possível evitar processos mais complexos de o fazer através da identificação individual de cada uma das suas subregiões segmentadas.

O processo utilizado é simples e baseia-se numa escolha binária entre as duas opções, *SDF* e *CDF*, tal como se esquematiza na fig. 7.23. Baseia a escolha na determinação da distância, de cada vector de características a cada um dos valores centrais das classes respectivas, com opção pela menos distante.

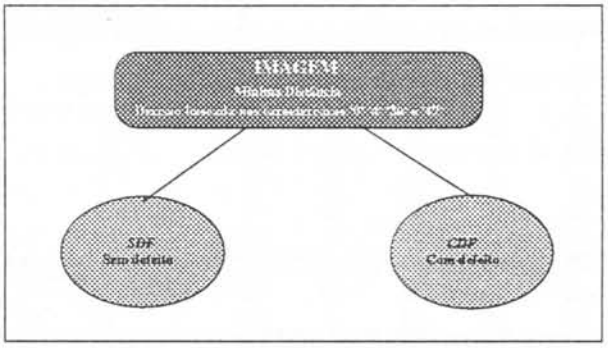


fig. 7.13 Diagrama de blocos do processo de decisão binária *SDF/CDF*

O processo de decisão[10] recorre, em primeiro lugar, a um conjunto de operações sobre imagem que permitem; não só efectuar uma segmentação nessa

[10] A função fddnd(...) em TECT1.C executa o pr ocesso de decisão que conduz ao reconhecimento, de saberem se a imagem tem defeito(*CDF*) ou não (*SDF*). A função recorre a um conjunto de funções invisiveis para o operador e que envolvem as funções: Fretbk1(...) em TSEGMT2.C na detecção da presença de background, bnthresd(...) em TECT.C na binarização da imagem, conectividade(...), Fecho1(...), limpz1(...), esqltzcf2(...) em TSEGMT3.C, na definição dos contornos e tpbin1(...) em TBINCT.C na definição das caracteristicas. O ficheiro TIPDND.TXT é utilizado pela função fddnd(...) na recolha de informação necessária à normalização das variáveis e na determinação dos factores utilizados nas funções de decisão, é obtido através da função cvmsic(...) em TECT.C.

imagem, precedidas de um processo de identificação da presença ou não de *"background"*, mas também da determinação dos contornos e atribuição de propriedades (variáveis características) à imagem global.

No processo de selecção ou escolha das variáveis que melhores condições oferecia na separação das duas classes *SDF* e *CDF* (selecção de características 7.6-2) foram calculados os factores de separação F[11], apresentados no Quadro 7.11, a partir dum conjunto de treino de amostras padrão, ou de protótipos identificativos das duas classes. As variáveis escolhidas para efectuar a classificação, foram desse modo previamente seleccionadas entre aquelas que apresentavam maiores valores indicativos (maiores valores para o factor F - Quadro 7.5) para poder separar as duas classes. Para este conjunto de variáveis características determina-se em seguida o grau de correlação[12] amostral das variáveis seleccionadas, tomadas duas a duas, como mostra o Quadro 7.8. As variáveis que apresentavam graus de correlação amostral elevado face às variáveis já escolhidas foram rejeitadas. A partir da tabela 7.8 foi assim possível seleccionar um conjunto de treino, constituído pelas seis variáveis que apresentavam maior potencialidade separadora e que tinham os menores índices de correlação amostral entre si.

0	1											
2	-70	1										
4	-61	88	1									
5	50	1	0	1								
17	-28	67	52	33	1							
20	-34	69	53	32	97	1						
32	-23	51	35	32	87	92	1					
34	-23	53	37	34	90	94	99	1				
39	69	-56	-57	51	-9	-11	-6	-6	1			
43	57	-26	-25	66	0	0	4	3	62	1		
44	43	-28	-26	30	-21	-25	-24	-23	45	48	1	
47	61	-32	-30	64	-8	-10	-5	-5	64	95	70	1
caracterist.	*0*	*2*	*4*	*5*	*17*	*20*	*32*	*34*	*39*	*43*	*44*	*47*

Quadro 7.8 Correlação entre variáveis características, definidas para um problema de classificação de duas classes

[11] Citado por Nazaré, Luís M. C. G. Meneses em trabalho de sintese de Provas de Capacidade Científica, 1990 e atribuida a Lopes, P. em Estabilidade Transitória de Sistemas de Produção e Transporte de Energia, Dissertação de Doutoramento, Faculdade de Engenharia da Universidade do Porto.

[12] A correlação amostral entre as variáveis préviamente selecionadas, tirada duas a duas, é implementada pela função crrl1(...) em TCORRLC.C e os resultados armazenados em CORR1.TXT. Os factores de correlação foram calculados entre as doze variáveis selecionadas (as que possuiam maiores índices de separação), *'0', '2', '4', '5', '17', '20', '32', '34', '39', '43', '44' e '47'* - Anexo 1. No processo de escolha, algumas variáveis foram preteridas, por exemplo '39' e '44' por se encontrarem muito correlacionadas com '47', ou por exemplo *'17', '20' '32' e '34'* muito correlacionadas entre si, levou à escolha de apenas uma delas *'20'*. A escolha final recaiu sobre as seis com melhores condições globais *'0', '4', '5', '20', '39' e '47'*.

A este conjunto de variáveis foi finalmente aplicado, um algoritmo de determinação de distância[13] inter-classes(Anexo **4**), e seleccionadas as variáveis que combinadas, correspondiam a uma distância média inter-classes maior. Do resultado destes cálculos tabelados no Anexo**1**, foi possível estabelecer como opção final, um conjunto de variáveis características constituído por *'0', '4', '20' e '47'*, como é mostrado no esquema da figura 7.13, onde figuram as variáveis seleccionadas.

Através deste processo de identificação, conseguem-se resultados razoáveis em termos de reconhecimento das imagens que não são portadoras de defeito, permitindo deste modo um processo de identificação mais rápido, já que neste caso o processo de análise termina. Já no caso inverso (identificação de imagens com defeitos, relacionados com defeitos do tipo *cardido* e *azulado*), persistem ainda alguns erros, em imagens com defeitos, reconhecidas como se de facto não os tivessem. O quadro da fig. 7.10 regista os resultados obtidos por ensaio de teste, sobre imagens de madeira de pinho, através do processo global de reconhecimento definido.

CLASSIFICAÇÃO IMAGENS SDF/CDF	IMAGENS SDF	IMAGENS CDF
% classificação correcta	92	86
% classificação incorrecta	8	14

Quadro 7.9 Resultados da classificação SDF/CDF

7.6.2 SELECÇÃO DE CARACTERÍSTICAS NO PROCESSO GLOBAL

O processo de selecção de características é decisivo, no que respeita à *"performance"* que apresentará o classificador no problema concreto. O problema da selecção de características pode ser resumido do seguinte modo: de todas as variáveis testadas, devem ser eliminadas aquelas que contiverem uma pequena quantidade de informação e aquelas que estiverem muito correlacionadas entre si, ou sejam, as que forem redundantes. É de salientar que o problema da selecção envolve também o da eficácia com que o classificador pode funcionar, já que o trabalho computacional e a complexidade deste, dependem largamente do número de variáveis a extrair em cada operação.

[13] A função difc(...) em TECT.C calcula a distância Euclidiana entre as categorias (ou classes) *SDF* e *CDF* e lança os dados em D_FC.TXT.

Este processo de selecção de características, foi adaptado a partir de processo citado por Nazaré[14] e envolve a ordenação prévia das variáveis em função da quantidade de informação que contêm, seguida da eliminação das variáveis redundantes.

O processo envolve o cálculo de um factor F de separação inter-classes, como mostra a expressão 7.19, através da assunção de que estamos na presença de um problema a duas classes [SDF *versus* CDF]. O factor F, que nos dá uma medida da capacidade de separação de cada variável, é calculado sobre um conjunto de treino, exactamente o conjunto de treino das imagens *com* e *sem defeito*[15]

$$F_i = \frac{\left| \overline{sdf}_i - \overline{cdf}_i \right|}{\sigma_{sdf_i} + \sigma_{cdf_i}} \qquad (7.18)$$

sendo:

F_i - medida F da característica i ;

$\overline{sdf_i}$ - média da característica i, para as imagens sem defeito

$\overline{cdf_i}$ - média da característica i, para as imagens com defeito

$\sigma^2_{sdf_i}$ - variância para os valores da característica i, para as imagens sem defeito.

$\sigma^2_{cdf_i}$ - variância para os valores da característica i, para as imagens com defeito.

A fórmula explícita um valor crescente para F, quanto maior for a sua capacidade de separação inter-classes, na medida em que quanto mais afastadas forem as médias e menores os desvios quadráticos das características nas amostras de treino, maior será este valor F.

Com os valores da medida do factor F determinado para cada variável, mostrados no Quadro 7.10, fica conhecida a sua capacidade separadora, através dos quais se pode iniciar o caminho de escolha das variáveis (características).

---

[14] Nazaré, Luís M. C. G. Meneses em trabalho de sintese de Provas de Capacidade Científica, 1990 e atribuida a Lopes, P. em Estabilidade Transitória de Sistemas de Produção e Transporte de Energia, Dissertação de Doutoramento, Faculdade de Engenharia da Universidade do Porto, 1988.

[15] Os valores das caracteristicas, para as imagens globais, *com* e *sem defeito* estão armazenados nos ficheiro CDF.TXT e SDF.TXT respectivamente. O factor de separação F é determinado através da função spitc(...) em TECT.C. O ficheiro FCTF.TXT armazena os valores relativos ao factor de separação inter-classes das variáveis, F.

características	0	1	2	3	4	5	6	7	8	9
0	1.11	0.56	0.7	0.08	0.65	0.33	0.43	0.04	0.39	0.51
10	0.31	0.57	0.05	0.59	0.54	0.51	0.31	0.56	0.59	0.23
20	0.59	0.02	0.66	0.59	0.38	0.01	0.83	0.61	0.42	0.44
30	0.45	0.51	0.51	0.21	0.46	0.48	0.3	0.52	0.21	0.56
40	0.54	0.55	0.65	0.53	0.47	0.45	0.36	0.51		

Quadro 7.10 Tabela de representação dos valores do factor F para as quarenta e oito características globais definidas

A necessidade de eliminar as variáveis redundantes, torna necessário o cálculo dos coeficientes de correlação amostral das variáveis, achado duas a duas, através da expressão (7.20).

$$\varphi_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}} \qquad (7.20)$$

O processo de selecção pode ser resumido do seguinte modo:

1 - Ordenar as variáveis por ordem decrescente do valor da medida do seu factor F.

2 - Escolher as dez primeiras variáveis que apresentem os maiores factores de separação.

3 - Determinar os coeficientes de correlação amostral entre as dez variáveis escolhidas, tomadas duas a duas.

4 - Rejeitar qualquer variável que apresente coeficiente de correlação amostral superior a 90%, com outra variável já escolhida.

5 - Repetir o processo a partir de 2 se não se obtiver o número de variáveis pretendido, depois de eliminar as indesejáveis.

7.7 PROCESSO DE DECISÃO BINÁRIA HIERARQUIZADA

7.7.1 DESCRIÇÃO DO PROCESSO

O processo de classificação hierarquizada, baseia a sua eficácia na existência de classes (ou grupos de classes) que podem ser separadas das restantes com maior facilidade. Nessa separação é determinante, a função de decisão aplicada em cada nó e as variáveis em que se baseia essa decisão. A precisão e a consistência com que é separada uma classe, determina a qualidade do sistema de classificação, já que um erro numa etapa anterior se repercute nas etapas sequentes. Por esse motivo as variáveis que separam cada classe, em cada nó da árvore de decisão, devem obedecer a critérios de exigência acrescida.

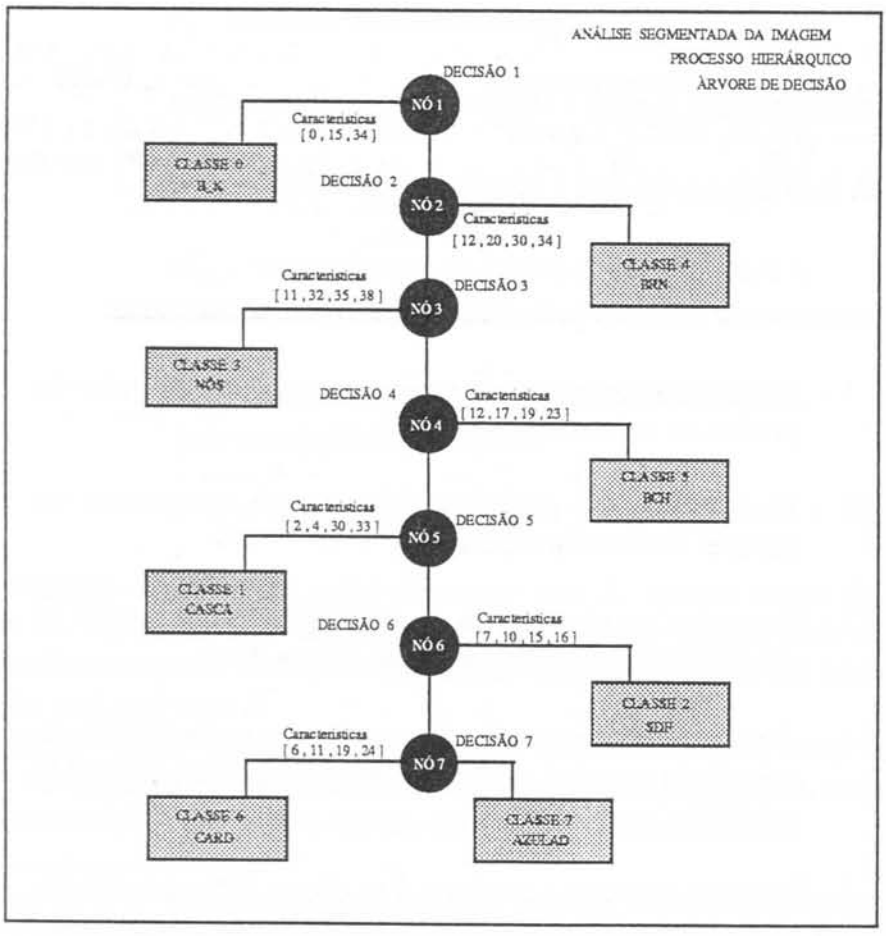


fig. 7.14 Árvore de decisão, em processo de classificação hierarquizada, para subregiões da imagem segmentada

Desse modo, utilizou-se na escolha das variáveis usadas em cada nó (que suportam a implementação das funções de decisão), uma medida que reflecte o seu poder de separação entre classes. A escolha recaiu no cálculo dos coeficientes de distância entre classes, pelos factores de distância de Jeffries-Matusita(JM)16. Através da medida e teste de ensaio, com diversos conjuntos de variáveis características (neste caso em número de duas e três - Anexo 7), escolheu-se entre elas, o conjunto de variáveis características que acusava a maior separabilidade entre classes. A arquitectura da árvore de decisão foi desse modo implementada de acordo com o estabelecido no esquema da figura 7.14, que resume o desempenho do *sistema de localização e identificação de defeitos em pranchas de madeira de pinho*, por processo de classificação hierarquizada.

7.7.2 ÁRVORES DE DECISÃO

As árvores de decisão, baseiam-se na definição de uma estrutura hierarquizada, de lógica binária, que funciona na base dum princípio, que consiste em comparar dentro de uma ordem pré-estabelecida (hierarquizada), os atributos x_k, de um vector de características x, com um limiar *("threshold")* previamente estabelecido, afim de tomar uma decisão necessariamente exclusiva.

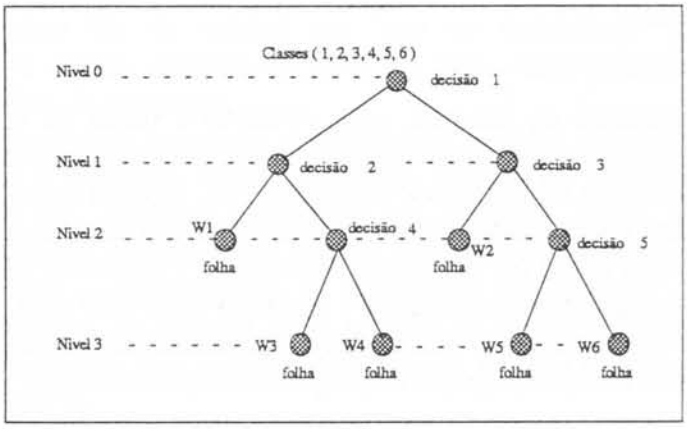


fig. 7.15 Arquitectura de uma árvore de decisão binária

16 Estes coeficientes são calculados pela utilização das funções fctJfMt(...), que determina o coeficiente τ de JM. Utiliza as funções Detn(...), McvJfMt(...) para determinação de médias, covariância e determinante dentro de fctJfMt(...). Os coeficientes determinados por este processo, para diversos conjuntos de variáveis pré selecionadas através dos factores de aglomeração apresentados em 7.4-2, entre diversas classes, encontram-se apresentados no Anexo 6, e constituiram a fonte de selecção das características utilizadas neste processo de classificação hierárquica.

Depois de efectuar a decisão em cada um dos nós da árvore, o número de classes potenciais, às quais pode ser atribuído x_k, fica diminuído. A cada nó da árvore corresponde um ou vários atributos x_k, com os quais se compara o limiar de decisão, ou aos quais é aplicado uma função de decisão determinada. Enquanto a cada nó corresponde a uma tomada de decisão, a uma folha corresponde uma classe única.

O reconhecimento de um vector de padrões x, equivale numa estrutura deste tipo, a percorrer a árvore desde a raiz até uma folha qualquer.

Este processo tem vantagens em termos do número de testes de decisão que necessita de efectuar, mas apresenta o inconveniente, das decisões tomadas terem uma natureza crítica, não suportando por isso o mínimo erro, sem que o processo incorra numa classificação errada. Por esse motivo, só pode ser utilizado se em cada nó a regra de decisão tiver natureza robusta.

7.7.3 Arquitectura do Algoritmo

O problema foi dividido em vários níveis de execução:

1 - Obtenção de imagem digitalizada, num formato 320x200 (colunas x linhas) em oito níveis de cinzento [imagens pré-gravadas, já que não se trabalhou com um sistema de aquisição de imagem em tempo real].

2 - Processamento de imagem, com obtenção a partir da imagem global de regiões segmentadas uniformes, e através da extracção dos contornos de cada uma dessas regiões, as orlas respectivas. Uma lista ligada passa a estar disponível no fim desta fase, com informação sobre os pixels que constituem a fronteira, além de outras informações topológicas e analíticas para cada uma destas regiões [num processo que envolve a segmentação, extracção de *pixels* não conexos, fecho e extracção de orlas/contornos].

3 - Extracção das variáveis características por análise das regiões, cálculo do valor das funções de decisão em cada nó da estrutura hierárquica - segundo esquema da fig.7.14, e tomada da correspondente decisão/classificação. A árvore de decisão binária apresentada na figura, associa a cada nó, uma função de decisão (que foi implementada na fase de teste e ensaio, de três modos diferentes: determinação da distância mínima a cada classe, classificador de Bayes, e distância de Mahalanobis. Um conjunto

de variáveis de decisão (referidas na figura em cada um dos nós) foi também escolhido, em função dos valores obtidos através de factores de distância inter-classes, na fase da selecção de características. As variáveis são escolhidas para cada um dos nós, mediante processo de selecção, descrito em 7.4-5, de modo a garantirem, a maior separação possível entre as classes.

4 - Display dos contornos de cada defeito e da sua identificação, segundo a nomenclatura seguinte: *casca(csc), nós(no), backup(b_k), cardidos(crd) azulados(azld), madeira escura sem defeito(sdf), bolsas de resina(brn)* e *brechas de resinagem(bch).*

O processo da construção/desenho das funções de decisão em cada nó (classificadores), foi feita à custa de uma série de conjuntos de treino (um por classe) num processo de aprendizagem supervisado, depois da aplicação dos processos descritos nos capítulos sobre classificação, 6.2-2, 6.3 e 6-5.[17].

As características utilizadas em cada um dos nós da árvore de decisão, estão em consonância com as características escolhidas no processo de aprendizagem e selecção, e com base nas quais foi implementado o classificador. Estas características são extraídas através da função selctd3(...)[operação 5 do diagrama de fluxos] em TECT.C e envolvem o cálculo de todas as características referenciadas no Anexo **5**. Apesar de em cada um dos nós da árvore, apenas se utilizarem algumas das características, o cálculo de todas elas apenas se deve à necessidade de possuir uma estrutura flexível durante a fase de teste, que permita o ensaio com diversas características, sem ter de refazer por isso os respectivos programas. Esta flexibilidade, teve custos contudo ao nível do peso computacional, acrescido em cada um dos passos do programa.

As características envolvem o cálculo de medidas estatísticas do primeiro grau, tipo média e variância, cinzento mínimo, número de máximos e mínimos no histograma, por ex., medidas estatísticas do 2º grau, como grandezas *GLDM* horizontais e verticais, ou grandezas *GLRLM* horizontais e verticais e ainda medidas geométricas, como a determinação de momentos das diversas regiões e ainda informação topológica da imagem como número de objectos (projecções) relações de forma, inclinação do eixo maior circunferência inscrita e circunscrita (todas as descritas no Anexo **3**).

O diagrama de fluxos da fig. 7.16, esquematiza através de blocos, o sistema implementado para a *detecção e identificação de defeitos em madeira de pinho*. O processo de classificação hierárquica envolve um conjunto de operações[18],

[17] A função DMdia(...) determina o vector de decisão de minima distância, que irá constituir o classificador por distância, em cada um dos nós da àrvore de decisão binária. Esta função está implementada em TECT.C.

[18] A função fd_d2(...) em TECT1.C executa o processo hierárquico de decisão binária que conduz à delimitação e ao reconhecimento de um conjunto de defeitos presentes numa imagem, se a imagem

referenciadas no diagrama por op. *x*, a que correspondem uma ou várias funções de decisão que as implementa.

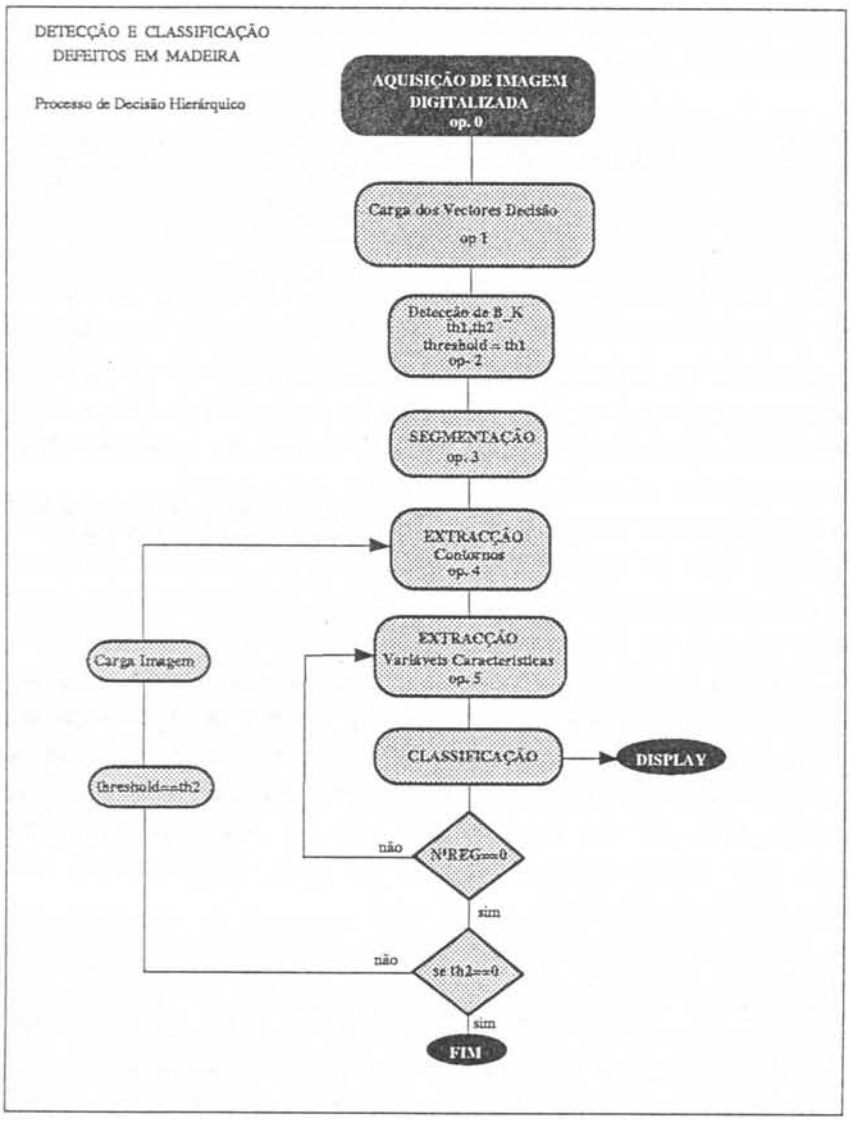


fig. 7.16 Diagrama de fluxos do sistema de detecção e identificação defeitos. Processo hierárquico de decisão binária

os contiver. Esses defeitos podem ser *'b-k'(background)*, *'casca'(csc)*, *'SDF'.(região delimitada escura não constituindodefeito)*, *'NÓS' (nós)*, *'BRN' (bolsas de resina)*, *'BCH' (brechas de resinagem)*, *'CARD' (cardido)*, e *'AZULD' (azulado)*. A função recorre a um conjunto operações implementadas por funções invisiveis para o operador, e que envolvem respectivamente: DMdia(...) - na op.1, em TECT.C; Fretbk1(...) - na op.2, em TSEGMT2.C na detecção da presença de background, bnthresd(...) e bntresd2(...) - na op.3, em TECT.C na binarização da imagem; conectividade(...), Fecho1(...), limpz1(...), esqltzcf2(...) - na op.4, em TSEGMT3.C na definição dos contornos; selctd3(...) - op.5, em TECT.C; e tpbin2(...) em TBINCT.C e a função fd_d2(...) na operação de decisão.

As imagens das figuras 7.17 e 7.18, bem como as imagens das figuras 7.26 a 7.33 do fim do capítulo, ilustram alguns dos resultados obtidos por processo de decisão binária hierarquizada.

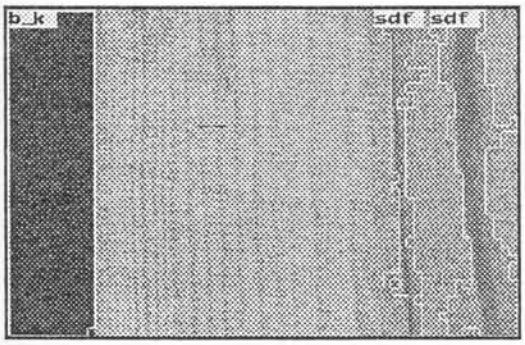


fig. 7.17 Imagem classificada, com duas regiões: *SDF* e presença de *backup*

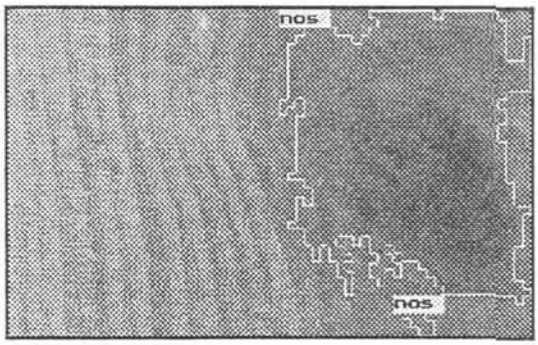


fig.7.18 Imagem classificada, com a presença de um nó

Dos ensaios realizados pelo processo de decisão binária hierarquizada, foi possível obter os resultados representados no Quadro 7.9, onde se pode observar que das oito classes testadas, quatro delas, *B_K, NÓS, BRN e BCH (brechas* de resinagem), oferecem algum grau de segurança na identificação, enquanto para as restantes quatro classes/defeitos, os resultados já não podem ser considerados tão bons.

classificação %	*B_K*	*CSC*	*SDF*	*NÓS*	*BRN*	*BCH*	*CRD*	*AZLD*
B_K	98	1	1					
CSC		38	14	8	18	4	12	6
SDF		6	44	10	14	8	6	12
NÓS	1		6	88		4	1	
BRN			8	6	76	6	4	
BCH		6	2	6	6	78	2	
CRD		12	16	4	8	12	36	12
AZLD		2	18	4	12	6	6	52

Quadro 7.11 Resultados de classificação, obtidos através de processo de decisão binária hierarquizada, do *sistema de localização e identificação de defeitos em madeira de pinho.*

O Quadro 7.11 mostra não só as taxas de sucesso de classificações correctas, obtidas pelo *sistema de localização e identificação de defeitos em madeira de pinho,* por classe de defeito, mas também a tendência de engano para cada uma dessas classes. Claramente se vê que as taxas de sucesso são muito maiores para as quatro classes já apontadas: *B_K, NÓS, BRN e BCH.*

7.8 PROCESSO CONJUNTO DE CLASSIFICAÇÃO GLOBAL E HIERÁRQUICA

Este processo de localização e reconhecimento de defeitos em madeira, não é mais do que a fusão dos processos de classificação global, descritos no ponto 7.6, com o processo de decisão hierárquica, descrito em 7.7. A essa cadeia de decisão foram acrescentados critérios *ad hoc*, definidos empiricamente. Pelo facto de já terem sido descritos anteriormente, só se apresenta o esquema de blocos da fig. 7.19, com a indicação, para os diversos níveis, das funções de decisão e das *n* variáveis utilizadas em cada nó.

classificação %	*B_K*	*CSC*	*SDF*	*NÓS*	*BRN*	*BCH*	*CRD*	*AZLD*
B_K	98	1	1	-	-	-	-	-
CSC	-	38	14	8	18	4	12	6
SDF	-	2	63	6	8	4	4	3
NÓS	-	-	4	91	3	2	-	-
BRN	-	-	8	6	80	4	2	-
BCH	-	5	3	6	6	79	1	-
CRD	-	6	12	4	8	8	52	10
AZLD	-	-	15	4	7	6	7	59

Quadro 7.12 Resultados de classificação, obtidos através de processo de decisão binária global e hierarquizada, do *sistema de localização e identificação de defeitos em madeira de pinho.*

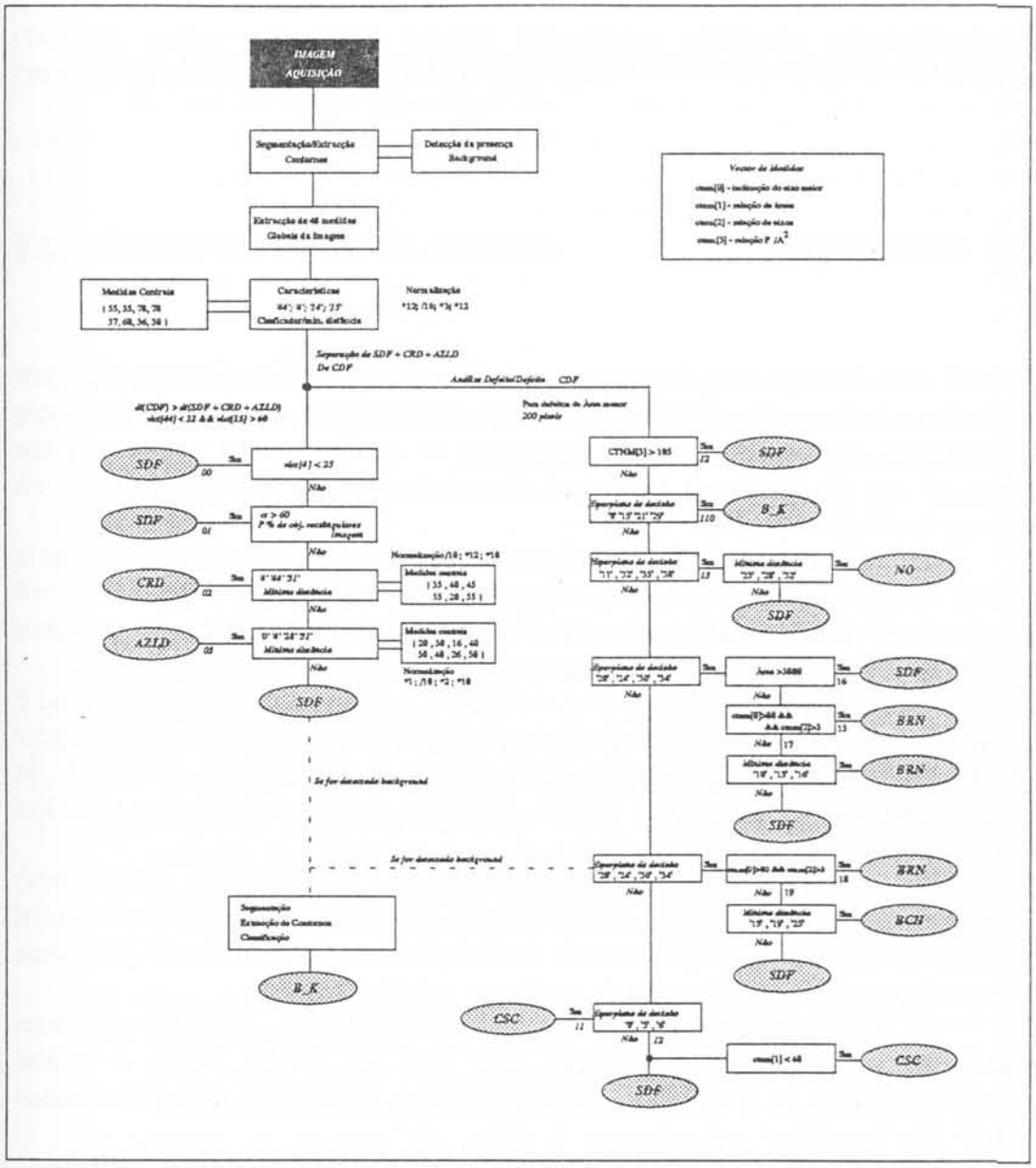


fig. 7.19 Esquema de decisão do sistema final de localização e reconhecimento de defeitos em madeira de pinho

A tabela do Quadro 7.12 mostra os resultados obtidos com o processo de classificação. Comparativamente com os resultados do Quadro 7.11, notam-se melhorias na identificação das classes/defeito *SDF*, principalmente porque algumas imagens sem defeito já não passam pelo processo de classificação defeito a defeito. Algumas melhorias menos significativas são também notadas na identificação dos defeitos *AZLD* e *CRD*. Para as classes *NÓS*, *BRN* e *BCH* as diferenças nos resultados são pouco significativas.

7.9 IDENTIFICAÇÃO DE DEFEITOS PREVIAMENTE MARCADOS

7.9.1 INTRODUÇÃO

É usual encontrar, integrado em diversos processos produtivos, quer envolvam madeira, pedras, peles ou vestuário, a integração da marcação de defeitos e de referências de trabalho, com giz geralmente de cor branca (ou doutra cor, em função da cor do artefacto), que sirva de referência e de indicação diversa ao processo de fabrico a jusante.

Com o objectivo de fazer um estudo, que forneça indicações, sobre uma possível automatização da identificação desse tipo de marcações com giz branco, em pranchas de madeira aplainadas, é apresentado neste ponto do capítulo, um processo utilizado na identificação desse tipo de marcações.

Saliente-se que o giz branco foi escolhido, por ser o que mais usualmente é encontrado e utilizado em processos de marcação industrial, e por se adaptar razoavelmente bem à sua identificação sobre madeira, a qual depende obviamente da cor da madeira que lhe serve de suporte. Supõe-se que se trabalha com madeiras exoticas, portanto de várias cores/*NDC*.

Se a detecção fosse feita, ou tentada sobre faia (madeira muito clara) provavelmente e por esse motivo, a detecção não resultaria, o que imporia a escolha de outra cor para o giz, que através das adaptações necessárias no programa, poderiam ser tentadas na obtenção de melhores resultados.

Na preparação desta operação, as peças devem ser inspeccionadas visualmente e a marcação deverá ser feita com giz branco, sobre o defeito visualizado, através de um polígono ou circunferência fechada, ou no mínimo pouco aberta.

O princípio de detecção baseia-se no facto de o giz, por possuir cor branca, poder ser identificado através do seu destaque sobre a superfície em que foi inscrito, que tem forçosamente de ser mais escura, sem o que o processo não seria viável. Por outras palavras, a superfície da madeira sendo relativamente mais escura que a marcação, poderá ser destacada como se esta fosse um *"background"*, retendo sobre esta o contorno branco e a sua localização.

O problema parecendo simples, levanta contudo algumas dificuldades, nomeadamente pelo facto de o *"background"*, entendido deste modo, não se apresentar duma forma uniforme, nem em termos de cor, nem em termos de textura, e o facto da marcação poder ter zonas indefinidas ou abertas, acrescido ainda do facto de se trabalhar sobre tipos de madeira diferentes, com muitos veios ou com zonas muito claras.

A qualidade da luz incidente sobre a prancha de madeira, e o estado e natureza do polimento da superfície (por ex: o grau(%) de gordura da própria

madeira), podem acrescentar também dificuldades adicionais, principalmente se provocarem reflexos sobre as zonas de marcação.

7.9.2 ARQUITECTURA DO ALGORITMO

O algoritmo de reconhecimento de marcações a giz em pranchas de madeira, assenta na utilização da função thd7(...)[19], e está estruturado do seguinte modo:

1 - Fazer a aquisição da imagem com marcações de giz e efectuar a sua segmentação. Esta operação deverá retornar também o valor da variância e do *"threshold"* (patamar de segmentação).

2 - Se a variância for muito elevada ($\sigma^2 > \tau$), definir novo *"threshold"* através de nova segmentação para a região mais clara da madeira, utilizando nessa divisão o *"threshold"* anterior. Se a variância for muito baixa corrigir o valor de *th* em função de σ^2, senão manter o valor de *th*.

3 - Com o novo *"threshold"*, definir uma transformação linear que aumente o contraste da imagem (tal como foi definida no cap. 2.3.4), em que $s=(s_{max}-s_{min})/(r_{max}-r_{min}).r$ é a função de transferência. Esta operação pretende aumentar a dispersão dos pixels na imagem de modo a simplificar a operação de segmentação da imagem na sua faixa de maior brilho (ou de maiores níveis de cinzento) onde se situa a marcação de giz. É de salientar que esta operação é indispensável em situações em que a diferença de cor entre a madeira e as marcações de giz sejam reduzidas, o que inviabilizaria a segmentação nessa área de imagem, se não se procedesse ao processo de separação dos níveis de cinzento/*NDC*.

---

[19]A função thd7( ) em TSEGMT1.C concentra os principais procedimentos de resolução do algoritmo. As funções float vlsegm2(unsigned int [ ], int *) determina o *threshold* e retorna o valor da variância da imagem - em TFUNCOES.C; esqltzcf1( ) retira o contorno das regiões segmentadas da imagem, esqltzC(char) determina o esqueleto dos segmentos, em TSEGMT1.C e tputimg1(...) em TFUNCOES.C, faz o display da imagem.

4 - Um novo valor do *"threshold"*, para uma nova segmentação é calculada para a banda clara da imagem, tendo em conta o espalhamento (dispersão) operado sobre os níveis de cinzento.

5 - Com o novo valor do *"threshold"* é feita a binarização da imagem, a que se segue uma operação de fecho, extracção dos contornos, esqueletização e display da imagem.

As imagens mostradas nas figuras 7.20 a 7.23, representam um processo de reconhecimento em madeira com marcações a giz, de pseudo defeitos existentes nas pranchas de madeira exótica.

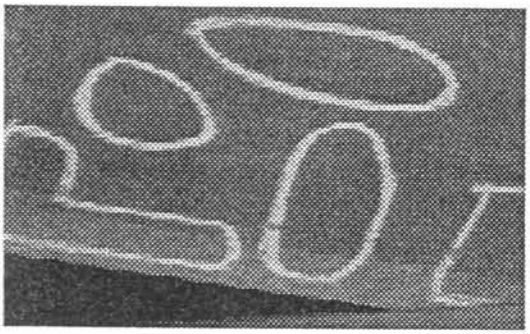


fig. 7.20 Imagem original marcada a giz

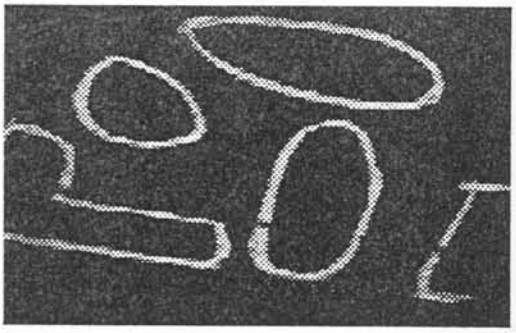


fig. 7.21 Imagem depois de operação de realce

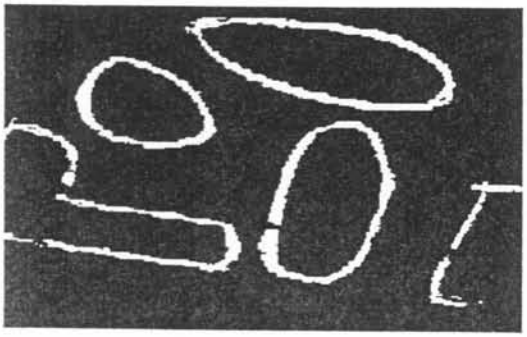


fig. 7.22 Imagem depois de binarizada

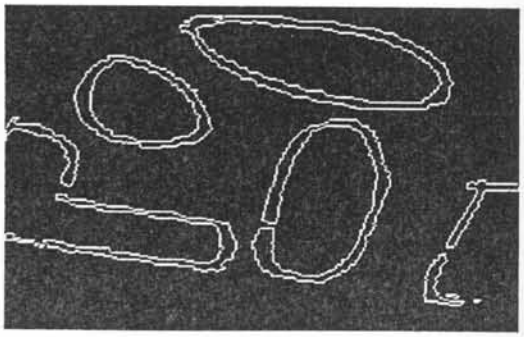


fig 7.23 Contorno extraído da imagem

A operação de realce efectuada sobre a primeira imagem (fig. 7.20) provoca uma alteração no histograma, do modo que está representado no gráfico da fig. 7.24.

O histograma da imagem após realce (apenas das regiões com pixels de valor superior a *th*) mostra o aparecimento de pequenos picos (máximos locais) no histograma, duma forma muito mais nítida. Um desses picos pertencerá inevitavelmente às regiões de contorno marcadas a giz branco. Através de processos de segmentação sucessiva, controlada pela medição da variância em cada região, é possível extrair os contornos previamente marcados na imagem.

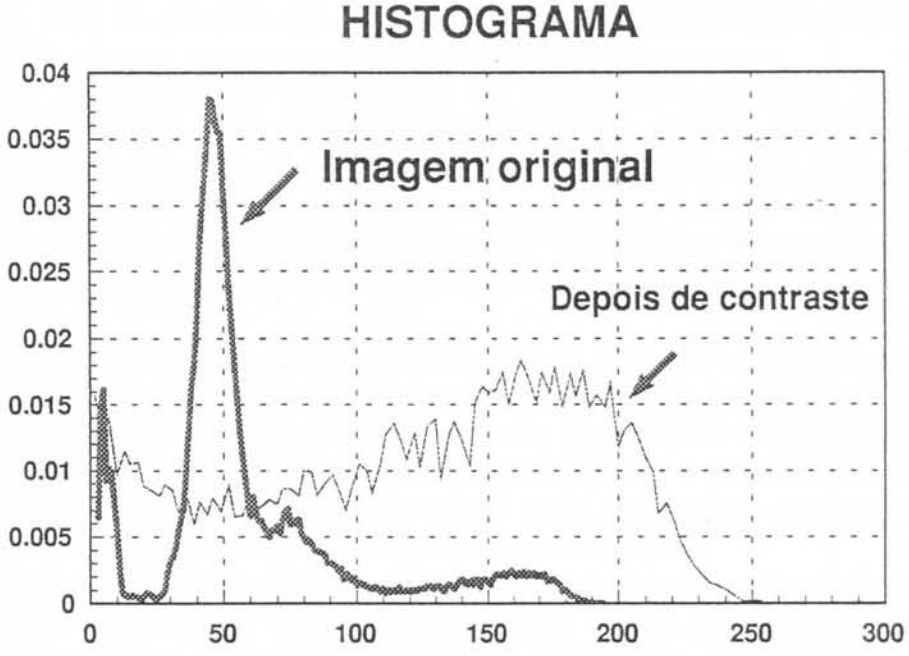


fig. 7.24 Histogramas das imagens das figuras 7.20 e 7.21

O diagrama de fluxos, representado no esq. da fig. 7.24, representa as etapas de execução do sistema utilizado na identificação de defeitos previamente marcados a giz, em pranchas de madeira.

A imagem é segmentada uma ou duas vezes em função da variância que apresentar a parte mais clara da imagem. Através de uma operação linear de remapeamento dos pixels, obtém-se uma imagem realçada $R(x,y)$. Esta imagem é novamente segmentada por *"threshold"* na sua região mais clara, e desta forma extraída a marcação de giz branco.

É um processo rápido e eficaz, já que não necessita de fazer correcções na imagem nem o seu display, trabalhando apenas sobre vectores que a representam.

As operações finais de *fecho*, *esqueletização*, destinam-se apenas a obter formas ajustadas de representação do contorno.

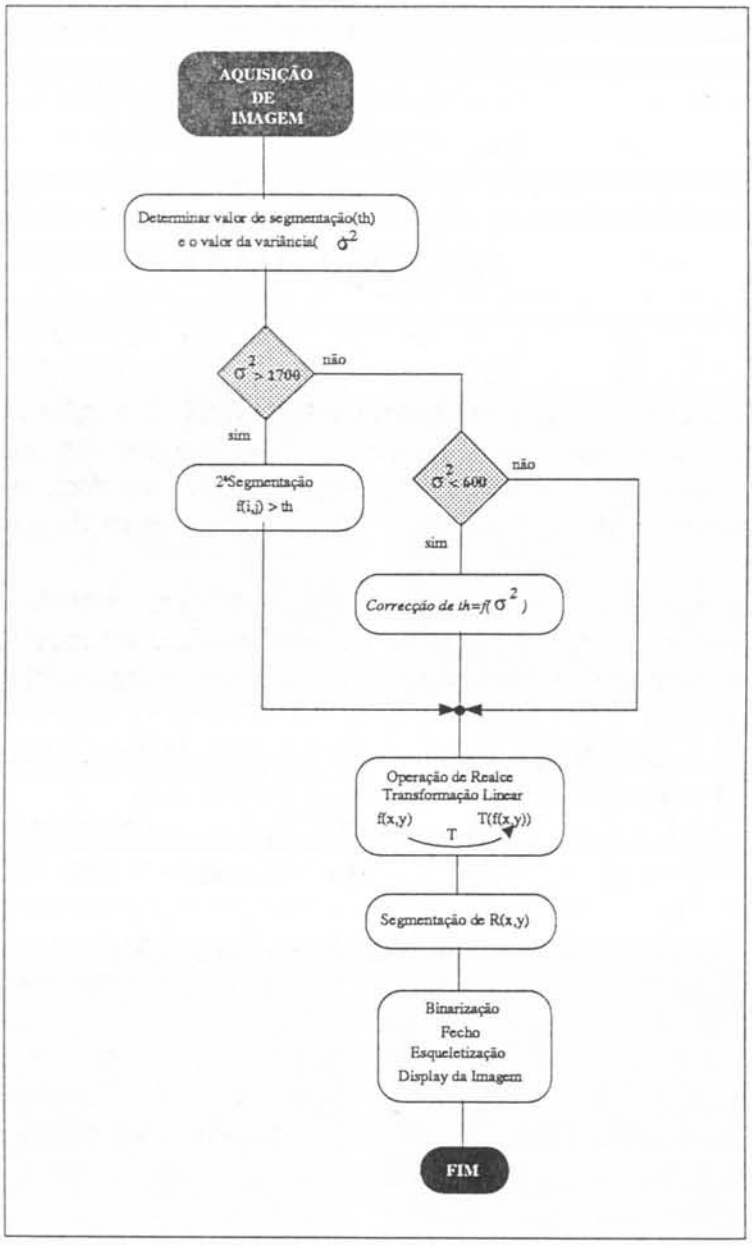


fig. 7.25 Diagrama de fluxos do algoritmo de detecção de marcações a giz em madeira

7.10 CONCLUSÕES

Este capítulo faz um estudo sobre alguns processos, analisados e testados durante a fase de implementação, sobre a *delimitação e reconhecimento de defeitos em madeira de pinho*. Por outras palavras, pode dizer-se que neste capítulo se faz a síntese entre as técnicas e conceitos apresentados nos capítulos anteriores (Cap. I a Cap. VI), através de uma aplicação de natureza prática. São apresentados os algoritmos desenvolvidos (ou dos vários algoritmos) que têm por objectivo a *localização e o reconhecimento de defeitos em madeira.*

Neste estudo sobre a *localização e reconhecimento de defeitos em madeira de pinho*, pode dizer-se que o papel principal é assumido pela identificação dos defeitos, mais do que pela sua localização, obtida por segmentação, determinação das regiões conexas na imagem e posterior extracção de contornos.

O reconhecimento dos defeitos foi procurado basicamente por quatro processos diferentes, tendo envolvido essencialmente os mesmos métodos no que se refere à localização e extracção de contornos (localização/detecção de defeitos). A esses processos chamámos respectivamente, pela ordem em que são tratados no capítulo: processo *ad-hoc* de reconhecimento, processo de *decisão singular em etapa única*, processo de *decisão binária global* e processo de *decisão binária hierarquizada.*

O reconhecimento envolve inicialmente a procura, através de inspecção global, da presença, ou não, de defeitos na madeira. Esta operação inicial tem o objectivo de acelerar o processo de identificação, já que se for detectada a presença de uma prancha sem defeito, a análise região a região não é processada. Pressupõe-se neste caso, que se a madeira não estiver limpa de defeitos, podem na mesma prancha apresentar-se uma ou mais regiões como defeituosas.

O problema da classificação é posto do seguinte modo: sabendo que existem oito classes diferentes de defeitos, como atribuir a cada região segmentada uma dessas classes com o máximo de precisão possível? O problema pressupõe a existência de uma função de decisão eficaz (obtida por processo de aprendizagem, a partir de uma amostra de treino) e a recolha e selecção das características adequadas aos objectivos da classificação.

Os dados foram definidos num espaço de multivariável e de multiclasse. A cada região delimitada foi atribuída uma classe em função das suas propriedades geométricas, topológicas, ou de textura (do primeiro ou do segundo grau), da sua superfície.

Os resultados obtidos, no que toca à identificação global da presença ou não, de defeito na madeira são satisfatórios. No que diz respeito à localização dos defeitos individualizados nas pranchas, quando estas os possuem, os resultados voltam também a satisfazer. No que toca à identificação desses defeitos, há um

grupo formado pelas classes de *'B_K', 'Nós' 'BCH' e 'BRN'* em que a percentagem de êxito é elevado, mas nas restantes classes de defeitos; *'SDF', 'CRD', 'AZLD' e 'CSC'* os resultados já não são tão bons quanto o desejável. Isso deve-se a diversos factores, nomeadamente à dificuldade em caracterizar um conjunto de variáveis estáveis para estes últimos defeitos, principalmente se atendermos ao facto de que a aquisição das imagens de trabalho não ter sido feita em ambiente totalmente controlado.

No que se refere aos vários processos utilizados, o processo de *decisão binária hierarquizada* é aquele que apresenta maior eficiência, de todos os processos testados.

Concluiu-se que, na *localização e reconhecimento de defeitos em madeira de pinho*, o que é critico na implementação de um processo de classificação não é o método utilizado na definição do classificador *(Bayes, Minima Distância, LMSE, Perceptron, Distância de Malahnobis)*, mas o conjunto de características seleccionadas para suportar o classificador. Se essas características apresentarem uma boa capacidade separadora para as classes, os resultados a partir dos diversos classificadores não diferirão muito.

O capítulo termina com a descrição de um sistema de localização de defeitos em madeira, previamente marcados a giz através de inspecção visual. O processo baseia-se na determinação do *"threshold"* adequado, capaz de separar os contornos marcados a giz. O algoritmo introduz um processo de determinação de níveis de segmentação múltiplos, controladas pela avaliação da variância na imagem. A imagem antes da segmentação final é sujeita a uma operação de realce, com a finalidade de dispersar os níveis de cinzento e desse modo facilitar a segmentação, nos limites do seu histograma de imagem.

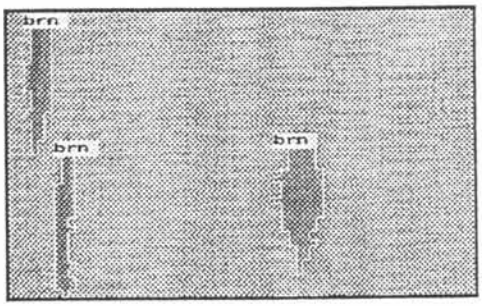


fig. 7.26 *Bolsas de resina* classificadas por processo hierárquico

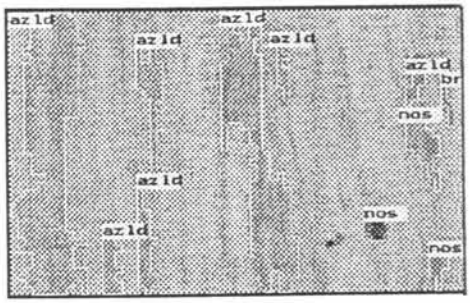


fig. 7.27 *Azulado* da madeira classificadas por processo hierárquico

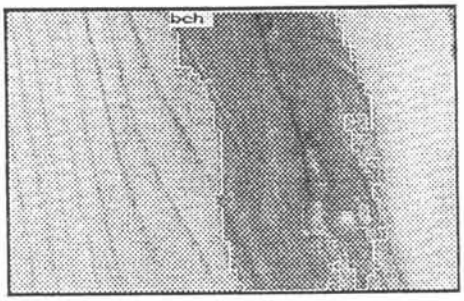


fig. 7.28 *Brecha de resinagem* classificadas por processo hierárquico

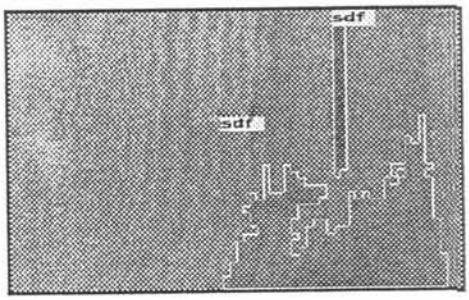


fig7.29 *Medula* e madeira *sem defeito* classificadas por processo hierárquico

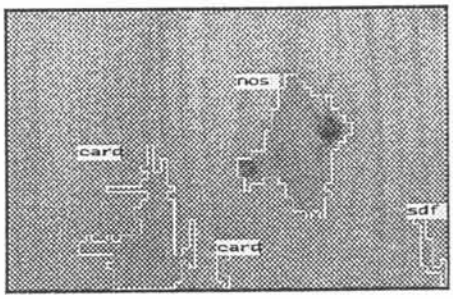


fig. 7.30 *Nós* classificadas. Erro em *SDF*, processo de classificação hierárquica

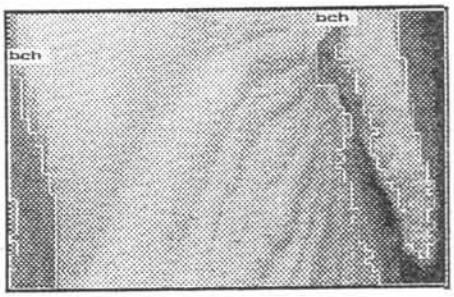


fig. 7.31 *Brechas de resinagem* da madeira. Processo classificação hierárquica

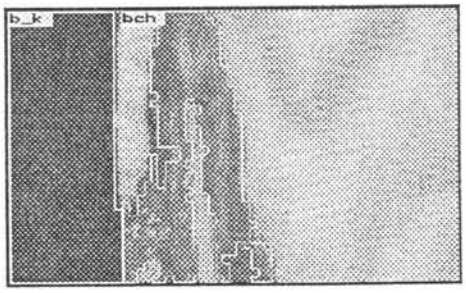


fig7.32 *Brechas de resinagem*, classificadas por processo hierárquico

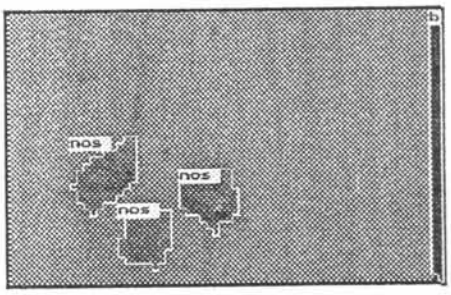


fig. 7.33 *Nós* e *B_K* na madeira, classificada por processo hierárquico

CAPÍTULO VIII

CONCLUSÕES

8 Conclusões

Este trabalho foi iniciado com o objectivo último de preparar um conjunto de *algoritmos/funções*, que permitissem medir ou testar as possibilidade que um sistema, baseado em visão computacional, oferecia (através da utilização de imagens obtidas por intermédio de sensores ópticos), para poder *localizar e identificar/reconhecer defeitos em pranchas de madeira de pinho* (marcados ou não previamente). O sistema foi implementado, e dele fazem parte um conjunto de funções, cujo nome, tarefa desempenhada, tipo de argumentos utilizados e arquivos onde se encontram, está referido no Anexo 7 desta dissertação de mestrado.

A primeira conclusão que se pode retirar do estudo empreendido, é de que os defeitos passíveis de serem encontrados, numa prancha de madeira de pinho, são muito mais numerosos do que à partida se poderia supor. Desse modo, o primeiro capítulo faz um estudo com algum detalhe sobre o assunto. Nele se referem defeitos atribuíveis a mal formações do lenho, como o fio torcido ou determinado tipo de brechas, mas também, aqueles defeitos provenientes das condições de crescimento e exploração das árvores, nós, bolsas de resina, brechas de resinagem, etc. São focados também defeitos ligados às condições sanitárias e biológicas da madeira, como certas podridões ou alguns ataques de parasitas e ainda aos defeitos provocados pelas condições de laboração da madeira, como fissuras de secagem, nós falhados, descaios, etc. Desse conjunto muito vasto de defeitos, foram seleccionados oito: *casca(csc), nós(no), backup(b_k), cardidos(crd), azulados(azld), madeira escura sem defeito(sdf), bolsas de resina(brn)* e *brechas de resinagem(bch)*. Essa selecção recaiu sobre os defeitos mais prováveis (maior frequência de ocorrência) nas pranchas de madeira de pinho, e naqueles que maiores reflexos repercutem na qualidade e valor comercial da madeira.

Outra conclusão resultante do trabalho, mas também da sua preparação prévia, é de que um sensor óptico poderá não ser o melhor sensor para a detecção de defeitos em madeira. Assim os raios *x*, sonares e outras formas de exploração do espectro electromagnético de frequências, poderão fornecer melhores e maiores indicações sobre as características físicas da madeira (estado sensorial obtido), que não tenham só a ver com o aspecto visual externo. Saliente-se que as técnicas de análise e reconhecimento de padrões que se utilizaram na identificação dos defeitos, a partir de variáveis características retiradas de imagens de madeira de pinho, podem ser obtidas a partir de padrões conseguidos a partir de qualquer outra fonte de sinal, que não a luz.

As operações de pré-tratamento de imagem, esquematizadas no Capítulo II e envolvendo descrições e exemplos sobre manipulação de histogramas, realce de imagens, suavização e filtragens, são no que respeita à *localização e reconhecimento de defeitos em madeira de pinho*, secundarizados, já que o processo pode na maior parte das vezes prescindir delas (excepção feita ao processo utilizado na *localização de defeitos em madeira de pinho previamente marcados a giz*). Além disso, estas técnicas são grandes consumidoras de tempo de processamento, o que faz protelar o seu emprego, ou reservar o seu emprego apenas a situações excepcionais.

O estudo destas técnicas de pré-tratamento de imagem fazem parte essencial do percurso de aprendizagem que se percorreu e foram essenciais na compreensão do significado da informação recolhida a partir dessas funções de processamento.

A segmentação (Capítulo III) é um dos capítulos essenciais deste trabalho. Através da segmentação é possível extrair regiões uniformes na imagem. Essa uniformidade pode ser medida em termos de brilho na *imagem/região* ou em termos de outra propriedade de textura qualquer. Ao longo da dissertação foram descritos e utilizados vários processos de segmentação de imagens, mas aquele que acabou por se impor como o mais eficaz (e também um dos mais rápidos) foi o processo de segmentação por *"threshold"*, extraído pelo método de Otsu. Como um defeito na madeira é sempre uma região com alguma particularidade comum, a segmentação foi largamente utilizada na *localização e reconhecimento de defeitos em madeira de pinho*.

No sistema implementado parte-se do princípio de que todas as regiões segmentadas numa prancha de madeira, são defeitos potenciais. Para que se possa acelerar o processo de classificação da região são extraídos os contornos que a delimitam. Essa informação, organizada sob a forma de lista ligada (de que consta a localização de todos os pixels constituintes da fronteira da região) é essencial para a sua manipulação como entidade autónoma e essencial para a sua representação rápida sobre a imagem da prancha no écran. O Capítulo IV aborda essa questão e envolve o

estudo de diversos assuntos ligados à manipulação topológica e morfológica de imagens binárias.

Outro aspecto fundamental é a caracterização de regiões conexas, entendidas como aquelas em que todos os *pixels* vizinhos apresentam *conectividade quatro* ou *oito*. Esse conceito é utilizado na delimitação de *regiões/defeito* homogéneas.

Um outro aspecto ainda ligado à caracterização das regiões da imagem com superfície homogénea, prende-se com a morfologia. Ou seja, com a caracterização da superfície em relação às suas propriedades geométricas.

Estes três conceitos: região conexa, extracção de orlas, caracterização geométrica da região, são completamente determinantes, para que se possa efectuar a localização do defeito na prancha de madeira. Em segundo lugar, essa operação de localização é essencial para que se possa fazer em seguida, a sua identificação.

Para caracterizar as *regiões/defeito* segmentadas na madeira e com contornos conhecidos, são extraídas medidas da superfície dessas regiões. Essa operação de análise baseia-se na extracção de medidas, através de métodos estatísticos do primeiro e do segundo grau. Grandezas com diversos significados (variáveis características) foram extraídas desse modo das *regiões/defeito*, tal como estão referidas e descritas no Anexo **2** e que utilizam processos de cálculo referidos no Capítulo V, casos dos métodos *SGLDM ("Spacial Level Dependence Matrices")*, *GLDM ("Gray Level Difference Method")* e do método *GLRLM ("Gray Level Length Method")*. Este tipo de variáveis, referido anteriormente, caracterizam a superfície do defeito, enquanto as variáveis obtidas por considerações morfológicas, caracterizam a sua geometria. É com este conjunto formado pelos dois tipos de variáveis (ligada à textura da superfície/defeito e à geometria da superfície/defeito) que se irá fazer a classificação do padrão *(região/defeito)* da classe desconhecida.

A selecção das variáveis características é um processo determinante para que se possa fazer uma classificação das *regiões/defeito* com eficácia. Essa selecção utilizou diversos métodos, tais como: o cálculo de factores de aglomeração das características por classe *(cap. 7.4.2)*, separabilidade das classes por medidas de distância *(cap. 7.4.3)*, separação de classes pelo método de Jeffries-Matusita(JM) *(cap. 7.4)* e o cálculo do factor de separação *F (cap. 7.6.2)*.

As variáveis que melhores condições oferecem para separar globalmente imagens com defeito *(cdf)* de imagens sem defeito *(sdf)*, englobam características, tais como: variância, número global de regiões na imagem, energia (às diferenças) *GLDM* $(d=1, \theta = 0^{\circ})$ e percentagem da cadeia *GLRLM* $(\theta = 90^{\circ})$. A referência a algumas características utilizadas na identificação de *regiões/defeito* de forma individualizada, depende largamente das classes ou conjunto de classes que se pretenda separar. O capítulo VII referência (caso a caso para os vários processos apresentados), essas características. A título de exemplo podemos referir a *variância*, a *média às diferenças*, a *área*, a *relação de forma* P^2/A, etc., só para citar algumas. Não se pode

contudo falar de características, que duma forma universal, permitam separar todas as classes, nem de características excepcionalmente estáveis que garantam com tranquilidade uma identificação sempre correcta.

Constata-se a existência de classes, tais como: *casca(csc), cardidos(crd), azulados(azld), madeira escura sem defeito(sdf)*, que pelo facto de não poderem ser identificadas por nenhum tipo de medida geométrica, torna a sua classificação, exclusivamente dependente das características de textura, medidas sobre as amostras. Pelo facto de não se encontrarem características com grau de estabilidade (ou de aglomeração) elevada, a sua classificação torna-se extremamente difícil e falível.

Os classificadores foram escolhidos com base em metodologias diferenciadas. O Capítulo VI faz um resumo e apresenta exemplos de classificação, aplicados à madeira, através de algumas técnicas supervisadas e não supervisadas, do cálculo de funções de decisão. Os classificadores utilizados foram essencialmente: métodos de classificação por *mínima distância*, *classificadores lineares* obtidos por métodos de aprendizagem não paramétrica *"perceptron criterion function"*, método *lmse "least-mean-square-error algorithm"* e finalmente métodos estatísticos, baseados em processos de decisão de Bayes e cálculo da distância de Mahalonobis. Os classificadores foram obtidos por intermédio de quatrocentos defeitos identificados em cento e sessenta amostras, que designámos por amostras protótipo, e a partir das quais treinamos os classificadores, ou a partir das quais obtivemos medidas centrais como: médias, variâncias, matrizes de covariância, etc.

No Capítulo VII foram ensaiados diversos processos de teste em defeitos, sobre pranchas de madeira de pinho. Os objectivos eram localizar os defeitos e identificar-lhes as famílias.

O primeiro processo, *ad hoc*, foi assim chamado por assentar a escolha das variáveis características utilizadas no reconhecimento de padrões, apenas em bases empíricas. Limita a classificação apenas a seis defeitos: *nós(no), cardidos(crd), azulados(azld), madeira escura sem defeito(sdf), bolsas de resina(brn)* e *brechas de resinagem(bch)* e utiliza um processo de localização do defeito semelhante aos outros processos apresentados.

Os outros processos são referidos no texto, respectivamente por: *processo de reconhecimento de defeitos em madeira, por decisão singular de etapa única*; *processo de decisão binária global* e o *processo de decisão binária hierarquizada*. Têm de comum o facto de utilizarem uma metodologia de selecção prévia das características, como forma de melhorar essa escolha, utilizada na fase da classificação. Os processos são diversos, sendo um deles global (o primeiro). Importa

reter que é o *processo de decisão binária hierarquizada*, o que apresenta melhores resultados no plano da identificação dos defeitos. Este último processo foi ainda avaliado, em conjunto com o *processo de decisão binária global*, com obtenção de resultados ligeiramente melhores, onde se salienta principalmente uma melhoria significativa na identificação da classe *sdf*.

Dos processos de teste retêm-se algumas conclusões, das quais referimos as mais significativas:

1 - A identificação da presença de defeitos ou não na prancha, corresponde à primeira decisão que é necessário tomar no processo de reconhecimento em madeira. Informações deste tipo, efectuadas na fase de análise global do estado da prancha (ensaios onde se conseguem taxas de sucesso superiores a 90%), são muito importantes porque permitem uma economia de esforço computacional. Nestes casos, a análise defeito a defeito não é processada, sendo o resultado do ensaio correcto se a prancha não contiver efectivamente defeitos.

2 - Se nos limitar-mos a defeitos de grande dimensão, e apenas a cinco classes, *B_K, NÓS, SDF, BRN e BCH,* é possível conseguir taxas de identificação com sucesso superiores a 80%.

3 - Através de um processo de *decisão singular de etapa única* não é possível separar com eficácia, um conjunto elevado de classes, com as peculiaridades apresentadas pelo problema de *identificação de defeitos em madeira de pinho*. Mesmo que o conjunto de variáveis características, utilizado na definição do classificador, fosse aumentado, a taxa de erros obtida por esse processo continuava a ser elevado.

4 - Num *processo de decisão binária hierarquizada*, é possível identificar bem alguns tipos de *defeitos/classes*, tais como: *B_K* e *NÓS*, e outras com algum sucesso (taxas > 75%) como: *BCH* e *BRN*.

5 - O *processo de decisão binária hierarquizada* apresenta as seguintes vantagens em relação ao processo unitário:

a) Ajusta as variáveis características, à classe que se pretende separar, em cada nó.

b) Permite ir deixando para o fim, os problemas de maior difículdade de resolução, diminuindo deste modo a probabilidade de erro, já que a escolha passa a incidir sobre um leque de opções menor.

6 - O classificador de *Bayes* e a distância de *Mahalonobis* apresentam duas dificuldades:

a) Exigem o conhecimento da distribuição de probabilidades à priori em multivariável. A suposição de que a distribuição é normal, conduz geralmente a erros elevados ou no minimo a desvios consideráveis.

b) A alternativa consistiria na recolha e memorização numa base de dados de todas as probabilidades de ocorrência para cada possibilidade de ocorrência das variáveis, o que para mais de duas variáveis em bases amostrais elevadas, se tornaria proibitivo.

7 - A alternativa aos classificadores de *Bayes* e de *Mahalonobis* são os classificadores de mínima distância. São mais fáceis de construir e mais rápidos de calcular, mas podem ser menos eficientes se as classes não forem linearmente separáveis e se a distância inter-classes for reduzida.

8 - Em muitas situações, o reconhecimento dos defeitos é mal efectuado, não porque as características tenham sido mal escolhidas, tenham pouca capacidade discriminante ou porque o defeito tenha características atípicas, mas porque a segmentação não ficou ajustada correctamente e englobou regiões não pertencentes ao defeito, ou que apenas lhe

pertencem marginalmente, ou ainda porque foram englobados defeitos de espécies diferentes.

9 - A forma é um mau indicador característico para a caracterização da generalidade dos defeitos: ou porque estes são irregulares, casos dos *SDF, CRD, CSC, AZLD, BCH*, ou porque, apesar de serem mais ou menos regulares *B_K, NO, BRN*, a segmentação pode não lhes manter a forma, ou podem aparecer incompletos em parcelas de imagem. Por outro lado, os restantes defeitos podem aparecer disfarçados com formas regulares, vestindo a capa dos verdadeiros *defeitos* regulares. Ou seja, a forma só por si não chega para fazer a identificação de *defeitos* em madeira.

10 - Genericamente pode dizer-se que o sistema implementado para a *localização e reconhecimento de defeitos em madeira de pinho*, funciona bem no aspecto da localização dos defeitos; razoavelmente bem na identificação de algumas *classes/defeito*: *B_K, NO, BRN, BCH*; menos bem em defeitos difíceis de caracterizar: *SDF, CRD, CSC, AZLD*; de uma forma sofrível no balanço global, dado que as caracterizações incorrectas degradam o aproveitamento do sistema.

11 - Nota final: importa salientar que a base de incidência dos testes de ensaio, assentaram num número relativamente pequeno de amostras. Importa por isso, aprofundar no futuro os testes agora realizados, para que as conclusões, que se pensa serem correctas, possam ser cofirmadas ou melhoradas.

Em resumo, pode dizer-se que a localização e reconhecimento de defeitos em madeira por via automática não é um problema fácil de resolver, principalmente porque as medidas características necessárias para identificar os defeitos, têm de ser caracterizadas por índices de aglomeração elevados. Das características testadas, verifica-se que para um conjunto de classes, *SDF, CRD, CSC, AZLD, BCH*, essa estabilidade do valor das características não se verifica, quando se passa de uma amostra para outra. A localização de defeitos, é por seu turno, uma tarefa mais fácil. Contudo, necessita da capacidade de reconhecimento do sistema, para eliminar as regiões segmentadas *sdf* (sem defeito). Para um conjunto de defeitos mais limitado, pode

apesar de tudo afirmar-se, que o comportamento do sistema de reconhecimento evidencia já um comportamento aceitável.

Refira-se finalmente, que este trabalho não fornece uma solução acabada para um problema que se apresenta com várias vertentes e que apresenta diversas hipóteses de desenvolvimento. A experimentação de novas variáveis, e o desenvolvimento de outro tipo de classificadores, bem como a ampliação do universo de treino e a utilização de um controlo maior em termos de resolução e iluminação das imagens, em fase de aquisição, poderiam trazer progressos ao desenvolvimento do estudo do *reconhecimento de defeitos em madeira de pinho*. Por outro lado, a exploração de novos sensores, e portanto de outros modos de registo de padrões na madeira, poderia fornecer alternativas, à identificação de defeitos por via óptica.

PARTE 4
ANEXOS

ANEXOS

Anexo 1	Características Globais
Anexo 2	Características Locais das *Regiões/Defeito*
Anexo 3	Selecção de Características. Medição do factor de separação *F* para as características globais definidas no Anexo 1. Apresentação do valor médio das características globais.
Anexo 4	Selecção de Características. Medição de distância Euclidiana, entre duas classes globais: *SDF* e *CDF*
Anexo 5	Selecção de Características. Medição de distância Euclidiana, entre duas classes/defeito a partir de características locais
Anexo 6	Selecção de Características. Medição de distância de Jeffries Matusita, entre duas classes/defeito a partir de duas e três características locais
Anexo 7	Listagem de funções e de menus, do Sistema Mínimo de Imagem
Anexo 8	Exemplo de Implementação, de processo de *localização e reconhecimento de defeitos em madeira*, através do Programa fd_d6(...) desenvolvido em C.

ANEXO 1

Anexo 1 Características Globais

Foram definidas ao longo da dissertação, variáveis características extraídas das imagens, com as quais se pretendia caracterizar globalmente a imagem. Pretendia-se com isso, encontrar também o melhor conjunto de variáveis características, que pudesse com o máximo de eficácia, separar as imagens *com defeito* das imagens *sem defeito*. O **Anexo 1** faz a listagem dessas variáveis características e fornece os nomes utilizados nos programas com que são identificadas. O significado e modo de obter estas características, foram descritos ao longo dos capítulos quatro e cinco (cap. IV e cap. V).

Listagem do Nome das Características e Variáveis que as Representam Utilizadas no Teste de Separação Global de Imagens [duas classes - *sem defeito/com defeito*] em Madeira de Pinho

/* A função selct1() retira da imagem 48 características, com as quais se pretende investigar, a possibilidade de as poder integrar num processo de classificação, com duas classe: classe das imagens *sem defeitos(SDF)* ou classe das imagens *com defeitos(CDF)*

1ªsérie/medidas do 1º grau

/*	[0]v0 - valor mínimo (int)	int
/*	[1]v1 - desvio do pixel mais frequente em relação à média	int
/*	[2]n - número de máximos relativos	int
/*	[3]md - média (int)	int
/*	[4]vc - variância (int)	int
/*	[16]sk - skewness (float)	int
/*	[17]kt - kurtosis (float)	int

2ª Série /valores GLDM(d=1;0=0º)

/*	[18]mdia - media(às diferenças)	float
/*	[19]varc - variância(às diferenças)	float
/*	[20]sm - segundo momento angular(às diferenças)	float
/*	[21]ent - entropia(às diferenças)	float
/*	[22]miv - média inversa	

3ª Série /valores GLDM(d=2;0=0º)

/*	[23]mdia - media(às diferenças)	float
/*	[24]varc - variância(às diferenças)	float
/*	[25]sm - segundo momento angular(às diferenças)	float
/*	[26]ent - entropia(às diferenças)	float
/*	[27]miv - média inversa	float

4ª Série /valores GLDM(d=1;0=90º)

/*	[28]mdia - media(às diferenças)	float
/*	[29]varc - variância(às diferenças)	float
/*	[30]sm - segundo momento angular(às diferenças)	float
/*	[31]ent - entropia(às diferenças)	float
/*	[32]miv - média inversa	float

5ª Série /valores GLDM(d=2;0=90º)

/*	[33]mdia-media(às diferenças)	float
/*	[34]varc - variância(às diferenças)	float
/*	[35]sm - segundo momento angular(às diferenças)	float
/*	[36]ent - entropia(às diferenças)	float
/*	[37]miv - média inversa	float

6ª Série /valores GLRLM(0=0º)

/*	[38]mdia - SRE-enfase nas cadeias curtas	float
/*	[39]varc - LRE-enfase nas cadeias longas	float
/*	[40]sm - RLN-não uniformidade do comprimento	float
/*	[41]ent - NNG-não uniformidade dos níveis de cinzento	float
/*	[42]miv - RP -percentagem da cadeia	float

7ª Série /valores GLRLM(0=90º)

/*	[43]mdia - SRE-enfase nas cadeias curtas	float
/*	[44]varc - LRE-enfase nas cadeias longas	float
/*	[45]sm - RLN-não uniformidade do comprimento	float
/*	[46]ent - NNG-não uniformidade dos níveis de cinzento	float
/*	[47]miv - RP -percentagem da cadeia	float

2ª Série /valores Gres GLDM(d=1;0=0º)

/*	[18]mdia - media(às diferenças)	float
/*	[19]varc - variância(às diferenças)	float
/*	[20]sm - segundo momento angular(às diferenças)	float
/*	[21]ent - entropia(às diferenças)	float
/*	[22]miv - média inversa	

3ª Série /valores Gres GLDM(d=2;0=0º)

/*	[23]mdia - media(às diferenças)	float
/*	[24]varc - variância(às diferenças)	float
/*	[25]sm - segundo momento angular(às diferenças)	float
/*	[26]ent - entropia(às diferenças)	float
/*	[27]miv - média inversa	float

4ª Série /valores Gres GLDM(d=1;0=90º)

/*	[28]mdia - media(às diferenças)	float
/*	[29]varc - variância(às diferenças)	float
/*	[30]sm - segundo momento angular(às diferenças)	float
/*	[31]ent - entropia(às diferenças)	float
/*	[32]miv - média inversa	float

5ª Série /valores Gres GLDM(d=2;0=90º)

/*	[33]mdia-media(às diferenças)	float
/*	[34]varc - variância(às diferenças)	float
/*	[35]sm - segundo momento angular(às diferenças)	float
/*	[36]ent - entropia(às diferenças)	float
/*	[37]miv - média inversa	float

6ª Série /valores Gres GLRLM(0=0º)

/*	[38]mdia - SRE-enfase nas cadeias curtas	float
/*	[39]varc - LRE-enfase nas cadeias longas	float
/*	[40]sm - RLN-não uniformidade do comprimento	float
/*	[41]ent - NNG-não uniformidade dos níveis de cinzento	float
/*	[42]miv - RP -percentagem da cadeia	float

7ª Série /valores Gres GLRLM(0=90º)

/*	[43]mdia - SRE-enfase nas cadeias curtas	float
/*	[44]varc - LRE-enfase nas cadeias longas	float
/*	[45]sm - RLN-não uniformidade do comprimento	float
/*	[46]ent - NNG-não uniformidade dos níveis de cinzento	float
/*	[47]miv - RP -percentagem da cadeia	float

8ª Série /bitquad(q0,q1,q2,q3,q4,q5)

```
/*  [5]q0 - zero brancos                                  int
/*  [6]q1 - um branco                                     int
/*  [7]q2 - dois brancos                                  int
/*  [8]q3 - três brancos                                  int
/*  [9]q4 - quatro brancos                                int
/*  [10]q5 - dois brancos em diagonal                     int
```

9ª Série / quantidade de regiões

```
/*  [11]vl[0] - quantidade de objectos muito pequenos(<=25)                  int
/*  [12]vl[1] - quantidade de objectos pequenos(>25 && <=420)                int
/*  [13]vl[2] - quantidade de objectos médios(>420 && <=5000)                int
/*  [14]vl[3] - quantidade de objectos muito grandes(>5000)                  int
/*  [15]k  - número total de regiões                                         int
```

ANEXO 2

Anexo 2 Características Locais das *Regiões/Defeito*

Foram definidas ao longo da dissertação, variáveis características extraídas das imagens, que as pretendiam caracterizar localmente defeito a defeito. Pretendia-se com isso encontrar o melhor conjunto de variáveis características, que pudesse com o máximo de eficácia, separar as *regiões/defeito* nas imagens. Cada uma das oito classes poderia ser identifica por: classe 0 - *backup(B_K)*, classe 1 - *Casca(CSC)*, classe 2 - *Sem defeito(SDF)*, classe 3 - *Nós(NOS)*, classe 4 - *Bolsa de Resina(BRN)*, classe 5 - *Brechas de Resinagem(BCH)*, classe 6 - *Cardido(CRD)*, classe 7 - *Azulado da Madeira(AZLD)*. O **Anexo 2** faz a listagem dessas características e fornece o nome das variáveis utilizadas nos programas para as identificar. O significado e modo de obter estas características, foram descritos ao longo dos capítulos quatro e cinco (cap. IV e cap. V).

Este anexo apresenta ainda uma lista tabelada dos valores médios das características locais para cada uma das classes, obtidas sobre cento e sessenta amostras de madeira, a partir das quais se descriminaram trezentos defeitos diferentes.

LISTAGEM DO NOME DAS CARACTERÍSTICAS UTILIZADAS NA CLASSIFICAÇÃO LOCAL DE DEFEITOS EM MADEIRA ATRAVÉS DE PROCESSO HIERÁRQUICO

1ª Série / medidas do 1º grau

```
/*  [0]v0 - cinzento minimo                                  int
/*  [1]v1 - cinzento máximo                                  int
/*  [2]v3 - desvio relativo do cinzento mais frequente       int
/*  [3]max - número de máximos na imagem                     int
/*  [4]min - número de minimos na imagem                     int
/*  [5]n  - número de extremos no histograma                 int
/*  [6]md - media                                            int
/*  [7]vc - variância                                        int
/*  [8]sk - skewness                                         float
/*  [9]kt - kurtosis                                         float
```

2ª Série /valores Ges GLDM(d=1;θ=0º)

/*	[10]mdia - media(às diferenças)	float
/*	[11]varc - variância(às diferenças)	float
/*	[12]sm - segundo momento ângular(às diferenças)	float
/*	[13]ent - entropia(às diferenças)	float
/*	[14]miv - média inversa	float

3ª Série /valores Ges GLDM(d=1;θ=90º)

/*	[15]mdia - media(às diferenças)	float
/*	[16]varc - variância(às diferenças)	float
/*	[17]sm - segundo momento ângular(às diferenças)	float
/*	[18]ent - entropia(às diferenças)	float
/*	[19]miv - média inversa	float

4ª Série /valores Ges GLRLM(θ=90º)

/*	[20]mdia - SRE-enfase nas cadeias curtas	float
/*	[21]varc - LRE-enfase nas cadeias longas	float
/*	[22]sm - RLN-não uniformidade do comprimento	float
/*	[23]ent - NNG-não uniformidade dos niveis de cinzento	float
/*	[24]miv - RP -percentagem da cadeia	float

5ª Série /valores Ges GLRLM(θ=90º)

/*	[25]mdia - SRE-enfase nas cadeias curtas	float
/*	[26]varc - LRE-enfase nas cadeias longas	float
/*	[27]sm - RLN-não uniformidade do comprimento	float
/*	[28]ent - NNG-não uniformidade dos niveis de cinzento	float
/*	[29]miv - RP -percentagem da cadeia	float

6ª Série/Medidas das Topológicas

/*	[30]md(sk) - declive do eixo maior do defeito.	float
/*	[31]v1 - àrea em termos percentuais ocupada pela	float
/*	região, em relação ao rectângulo envolvente	float
/*	[32]vc(kt) - relação de forma P*P/A(12,56)	float
/*	[33]k - diâmetro da circunferência inscrita;	float
/*	[34]vrc - diâmetro da circunferência circunscrita	float

7º Série/momentoentos

/*	[35]mdia - centroide horizontal	float
/*	[36]sk - centroide vertical	float
/*	[37]kt - diagonalidade	float
/*	[38]varc - divergência horizontal	float
/*	[39]sm - divergência vertical	float
/*	[40]ent - balanceamento horizontal	float
/*	[41]miv - balanceamento vertical	float

VALOR MÉDIO NORMALIZADO DAS CARACTERÍSTICAS
OBTIDAS POR ENSAIO DE APRENDIZAGEM SOBRE CENTO E SESSENTA AMOSTRAS DE MADEIRA

caract\class	0	1	2	3	4	5	6	7
0	7.8	25.65	64.31	28.04	37.57	26.15	49.98	75.68
1	31.65	57.88	54.27	47.49	74.59	80.05	48.12	62.63
2	13.25	24.84	59.07	43.46	37.96	41.03	54.44	57.26
3	33.53	57.36	63.15	50.65	50.15	67.53	76.89	55.76
4	29.59	54.32	58.84	48.48	51.65	61.79	68.93	51.84
5	27.66	50	21.35	43.27	72.44	71.92	28.46	22.44
6	16.02	43.94	66.44	50.87	52.27	53.18	61.46	67.68
7	15.87	28.87	41.13	31.01	36.51	27.86	34.44	35.62
8	50.06	15.78	12.38	14.79	13.99	16.63	13.39	10.65
9	36.76	1.65	1.26	1.07	0.32	3.28	0.99	2.8
10	9.5	43.4	45.76	35.04	71.82	58.72	51.14	34.34
11	11.27	21.62	23.49	14.14	50.08	36.49	25.17	13.85
12	69.58	11.11	16.25	15.97	3.09	7.22	13.33	17.13
13	16.98	70.82	62.02	60.94	91.88	80.69	67.36	58.74
14	64.5	12.22	7.73	12.67	4.29	6.59	6.14	11.74
15	7.65	43.69	32.97	38.29	60.58	76.54	61.04	27.8
16	6.66	32.24	18.61	25.2	45.79	78.64	45.78	15.95
17	77.78	19.79	36.46	25.83	12.04	6.67	16.67	37.5
18	19.56	64.98	47.43	60.05	75.87	82.38	71.01	45.59
19	87.78	27.17	36.2	29.78	10.39	7.83	12.61	38.04
20	22.62	83.24	83.52	79.43	94.44	90.45	89.55	78.12
21	49.71	2.1	1.22	1.89	0.32	0.54	0.52	1.23
22	12.57	72.75	73.95	67.2	90.33	83.4	83	65.58
23	63.54	14.29	32.57	12.86	4.44	5.14	24.86	35.83
24	18.74	85.04	87.95	83.57	95.9	92.62	92.62	83.67
25	26.71	70.11	60.92	71.74	84.54	89.35	85.65	52.26
26	28.16	0.61	0.76	0.55	0.21	0.13	0.24	0.83
27	15.02	57.69	47.69	59.09	75.85	82.5	78.21	42.47
28	68.3	12.5	31.93	12.21	4.6	4.29	24.95	35.12
29	20.93	69.01	64.89	70.11	84.59	89.01	84.98	58.74
30	90.37	93.89	84.28	61.72	95	91	60.19	83.15
31	17.91	54.23	48.84	47.15	48.98	55.63	53	48.59
32	33.33	35.58	40.19	18.46	29.91	36.15	30.77	31.09
33	40.15	12.99	10.74	33.63	12.2	32.55	11.05	6.45
34	81.58	51.29	48.3	33.02	35.19	57.82	32.63	28.77
35	26.76	18.93	18.41	2.35	19.45	12.27	5.16	12
36	28.19	8.89	15.09	29.61	6.56	16.04	31.44	13.43
37	45.05	56.34	50.06	44.55	46.34	38.23	46.67	49.05
38	65.11	61.79	62.54	60.75	67.71	64.12	63.68	67.87
39	41.31	47.22	44.14	43.49	41.77	42.29	47.74	40.84
40	50.11	52.48	49.54	51.04	50.19	51.36	49.95	51.06
41	44.71	28.65	28.52	26.47	28.85	29.54	30.55	26.82

Tab. 1-Valor médio das caracteristicas, por classe

classe 0 - Backup(B_K1)
classe 1 - Casca(Casca)
classe 2 - Sem defeito(SDF)
classe 3 - Nós(Nos)
classe 4 - Bolsa de Resina(BRN)
classe 5 - Brechas de Resinagem(BCH)
classe 6 - Cardido(PCD)
classe 7 - Azulado da Madeira(AZULAD)

As variáveis caracteristicas referenciadas na 1ª coluna da tabela, estão descritas na pg. 1 deste anexo.

ANEXO 3

ANEXO 3 Selecção de Características. Medição do factor de separação *F* para as características globais definidas no Anexo 1. Apresentação do valor médio das características globais.

Este anexo apresenta duas tabelas. Uma mostra o conjunto de valores encontrados para o factor de ordenação *F*, definido em 7.6.2, e a outra mostra o valor médio encontrado para as quarenta e oito características globais definidas no Anexo 1. Estas tabelas são interessantes na resolução do problema da classificação global sdf/cdf da madeira e na selecção das características globais do mesmo problema.

factor F Dez/Unidades	0	1	2	3	4	5	6	7	8	9
0	1.11	0.56	0.70	0.08	0.65	0.31	0.56	0.59	0.23	0.59
1	0.02	0.66	0.59	0.38	0.01	0.83	0.61	0.42	0.44	0.45
2	0.51	0.51	0.21	0.46	0.48	0.30	0.52	0.21	0.56	0.54
3	0.55	0.65	0.53	0.47	0.45	0.36	0.51	0.33	0.43	0.04
4	0.39	0.51	0.31	0.57	0.05	0.59	0.54	0.51		

Tab. 1 Factor de ordenação *F* de características

Valores médios das 48 características globais SDF/CDF, medidas por intermédio dum conjunto de protótipos de imagens de treino

característica	SDF	CDF	característica	SDF	CDF	característica	SDF	CDF
0	76.91	30.67	16	0.01	-0.60	32	0.13	0.09
1	3.45	29.73	17	-0.22	1.03	33	9.23	12.10
2	9.00	19.85	18	11.31	14.87	34	150.19	282.82
3	144.73	142.27	19	183.92	338.79	35	0.05	0.04
4	349.73	1428.53	20	0.05	0.04	36	4.56	4.95
5	28.82	22.85	21	4.63	5.01	37	0.12	0.10
6	12.36	7.40	22	0.07	0.06	38	0.84	0.88
7	11.73	6.70	23	14.20	18.67	39	1.69	1.73
8	11.36	9.00	24	297.84	551.46	40	0.68	0.73
9	28.45	46.70	25	0.03	0.03	41	0.13	0.09
10	4.55	4.35	26	5.03	5.41	42	0.82	0.85
11	18.36	9.18	27	0.06	0.05	43	0.74	0.83
12	4.64	1.73	28	9.05	14.42	44	4.71	4.30
13	2.36	1.33	29	149.17	379.41	45	0.54	0.66
14	0.64	0.62	30	0.06	0.04	46	0.13	0.09
15	26.00	12.85	31	4.48	5.10	47	0.61	0.73

Tab. 2 Valores médios das quarenta e oito características globais *SDF/CDF*

ANEXO 4

Anexo 4 Selecção de Características. Medição da distância Euclidiana entre duas Classes Globais: *SDF* e *CDF*

Medição da Separação entre Classes

Distância Euclidiana entre as classes *SDF* e *CDF* (classes definidas globalmente para as imagens, *sem defeito* e *com defeito*), em função de uma, duas, três e quatro características. A descrição de cada uma das caracteristicas testadas está descrita no **Anexo 1**. As distâncias maiores correspondem a uma separação maior entre as classes respectivas.

Uma Variável

0	41.65	4	20.87	39	25.14
47	32.07	5	17.28	20	28.15

Duas Variáveis

0	4	46.59
0	39	48.65
0	47	52.57
0	5	45.10
0	20	50.27
4	39	32.67
4	47	38.27
4	5	27.10
4	20	35.04
39	47	40.75
39	5	30.51
39	20	37.74
47	5	36.43
47	20	42.67
5	20	33.03

três Variáveis

0 4 39	52.94
0 4 47	56.56
0 4 5	49.69
0 4 20	54.43
0 39 47	58.27
0 39 5	51.63
0 39 20	56.20
0 47 5	55.34
0 47 20	59.63
0 5 20	53.16
4 39 47	45.79
4 39 5	36.96
4 39 20	43.13
4 47 5	41.99
4 47 20	47.50
4 5 20	39.07
39 47 5	44.26
39 47 20	49.53
39 5 20	41.51
47 5 20	46.04

quatro Variáveis

0 4 39 47	61.90
0 4 39 5	55.69
0 4 39 20	59.96
0 4 47 5	59.14
0 4 47 20	63.18
0 4 5 20	57.11
0 39 47 5	60.78
0 39 47 20	64.71
0 39 5 20	58.80
0 47 5 20	62.09
4 39 47 5	48.94
4 39 47 20	53.74
4 39 5 20	46.46
4 47 5 20	50.55
7 47 5 20	52.45

Variáveis Caracteristicas Testadas

'0' - nivel de cinzento minimo
'4' - variância
'5' - skewness
'20' - entropia(d=1, θ =90º)
'39' - bit quad, q2
'47' - nº de objectos

ANEXO 5

Anexo 5 Selecção de Características. Medição da distância Euclidiana entre Classes/Defeito consideradas duas a duas

Separação entre Classes.

Distância Euclidiana entre *Classes/defeitos*, tomadas duas a duas em função de uma, duas, três ou quatro variáveis características. A listagem do valor médio das caracteristicas, em função de processo de aprendizagem tomado sobre cento e sessenta amostras, em trezentos defeitos delimitados, bem como a sua descrição, está indicada no Anexo 2. Quanto maior for o valor encontrado, maior a separação entre as classes.

classe 0 - backup(B_K1)
classe 1 - Casca(Casca)
classe 2 - Sem defeito(SDF)
classe 3 - Nós(Nos)
classe 4 - Bolsa de Resina(BRN)
classe 5 - Brechas de Resinagem(BCH)
classe 6 - Cardido(PCD)
Azulado 7. - Azulado da Madeira(AZULAD)

SEPARAÇÃO ENTRE A CLASSE '0' e as CLASSES '1' '2' '3' '4' '5' '6' e '7'.
As características testadas em função do seu factor potencial de separação são: '0' '3' 15' '16' '34' e '37'

Uma Variável

classe 0 cl2=1
* 0 17.85 * 3 23.83 * 15 36.03 * 16 25.58 * 34 30.29 * 37 11.29
classe 0 cl2=2
* 0 56.52 * 3 29.62 * 15 25.32 * 16 11.95 * 34 33.27 * 37 5.02
classe 0 cl2=3
* 0 20.24 * 3 17.12 * 15 30.64 * 16 18.54 * 34 48.56 * 37 0.50
classe 0 cl2=4
* 0 29.78 * 3 16.62 * 15 52.93 * 16 39.13 * 34 46.39 * 37 1.30
classe 0 cl2=5
* 0 18.35 * 3 34.01 * 15 68.89 * 16 71.98 * 34 23.76 * 37 6.82
classe 0 cl2=6
* 0 42.18 * 3 43.37 * 15 53.38 * 16 39.12 * 34 48.94 * 37 1.62
classe 0 cl2=7
* 0 67.88 * 3 22.24 * 15 20.15 * 16 9.29 * 34 52.81 * 37 4.00

Duas Variáveis

classe 0 cl2=1
0 -> * 3 29.78 * 15 40.21 * 16 31.19 * 34 35.16 * 37 21.12
3 -> * 15 43.20 * 16 34.96 * 34 38.54 * 37 26.37
15 -> * 16 44.19 * 34 47.07 * 37 37.76
16 -> * 34 39.64 * 37 27.96
34 -> * 37 32.32

classe 0 cse 0 cl2=2

0 ->	* 3	63.81	* 15	61.93	* 16	57.77	* 34	65.59	* 37	56.74
3 ->	* 15	38.97	* 16	31.94	* 34	44.55	* 37	30.04		
15 ->	* 16	28.00	* 34	41.81	* 37	25.81				
16 ->	* 34	35.35	* 37	12.96						
34 ->	* 37	33.65								

classe 0 cse 0 cl2=3

0 ->	* 3	26.51	* 15	36.72	* 16	27.45	* 34	52.61	* 37	20.25
3 ->	* 15	35.10	* 16	25.23	* 34	51.49	* 37	17.13		
15 ->	* 16	35.81	* 34	57.42	* 37	30.64				
16 ->	* 34	51.98	* 37	18.54						
34 ->	* 37	48.56								

classe 0 cse 0 cl2=4

0 ->	* 3	34.10	* 15	60.73	* 16	49.17	* 34	55.12	* 37	29.80
3 ->	* 15	55.48	* 16	42.51	* 34	49.28	* 37	16.67		
15 ->	* 16	65.83	* 34	70.38	* 37	52.95				
16 ->	* 34	60.69	* 37	39.15						
34 ->	* 37	46.41								

classe 0 cse 0 cl2=5

0 ->	* 3	38.64	* 15	71.29	* 16	74.28	* 34	30.02	* 37	19.58
3 ->	* 15	76.82	* 16	79.61	* 34	41.48	* 37	34.68		
15 ->	* 16	99.63	* 34	72.87	* 37	69.22				
16 ->	* 34	75.80	* 37	72.30						
34 ->	* 37	24.72								

classe 0 cse 0 cl2=6

0 ->	* 3	60.50	* 15	68.04	* 16	57.53	* 34	64.61	* 37	42.21
3 ->	* 15	68.78	* 16	58.40	* 34	65.39	* 37	43.40		
15 ->	* 16	66.18	* 34	72.43	* 37	53.41				
16 ->	* 34	62.66	* 37	39.15						
34 ->	* 37	48.97								

classe 0 cse 0 cl2=7

0 ->	* 3	71.43	* 15	70.81	* 16	68.52	* 34	86.00	* 37	68.00
3 ->	* 15	30.01	* 16	24.10	* 34	57.30	* 37	22.59		
15 ->	* 16	22.19	* 34	56.52	* 37	20.55				
16 ->	* 34	53.62	* 37	10.12						
34 ->	* 37	52.96								

três VariáVariáveis

classe 0 cse 0 cl2=1

0 3 15	15	46.75	0 3 16	39.26	0 3 34	42.47	0 3 37	31.85	0 15 16	47.66
0 15 34	334	50.34	0 15 37	41.77	0 16 34	43.48	0 16 37	33.17	0 34 37	36.92
3 15 16	316	50.21	3 15 34	52.76	3 15 37	44.65	3 16 34	46.26	3 16 37	36.74
3 34 37	337	40.16	15 16 34	53.57	15 16 37	45.61	15 34 37	48.41	16 34 37	41.22

classe 0 cse 0 cl2=2

0 3 15	15	68.65	0 3 16	64.92	0 3 34	71.96	0 3 37	64.01	0 15 16	63.08
0 15 34	734	70.30	0 15 37	62.14	0 16 34	66.67	0 16 37	57.99	0 34 37	65.78
3 15 16	316	40.76	3 15 34	51.24	3 15 37	39.29	3 16 34	46.12	3 16 37	32.33
3 34 37	437	44.83	15 16 34	43.49	15 16 37	28.45	15 34 37	42.11	16 34 37	35.71

classe 0 cse 0 cl2=3

0 3 15	45	40.52	0 3 16	32.35	0 3 34	55.33	0 3 37	26.52	0 15 16	41.14
0 15 34	834	60.88	0 15 37	36.73	0 16 34	55.78	0 16 37	27.45	0 34 37	52.61
3 15 16	316	39.69	3 15 34	59.92	3 15 37	35.10	3 16 34	54.73	3 16 37	25.24
3 34 37	537	51.49	15 16 34	60.34	15 16 37	35.81	15 34 37	57.42	16 34 37	51.98

classe 0 cse 0 cl2=4

0 3 15	65	62.97	0 3 16	51.91	0 3 34	57.58	0 3 37	34.13	0 15 16	72.25
0 15 34	734	76.42	0 15 37	60.75	0 16 34	67.60	0 16 37	49.19	0 34 37	55.14
3 15 16	516	67.89	3 15 34	72.32	3 15 37	55.50	3 16 34	62.93	3 16 37	42.53
3 34 37	437	49.30	15 16 34	80.53	15 16 37	65.84	15 34 37	70.40	16 34 37	60.70

classe 0 cse 0 cl2=5

0 3 15	75	78.98	0 3 16	81.69	0 3 34	45.36	0 3 37	39.24	0 15 16	101.31
0 15 34	734	75.14	0 15 37	71.61	0 16 34	77.99	0 16 37	74.59	0 34 37	30.79
3 15 16	516	105.27	3 15 34	80.41	3 15 37	77.12	3 16 34	83.08	3 16 37	79.90
3 34 37	437	42.04	15 16 34	102.42	15 16 37	99.86	15 34 37	73.19	16 34 37	76.10

classe 0 cse 0 cl2=6

0 3 15	65	80.68	0 3 16	72.04	0 3 34	77.82	0 3 37	60.52	0 15 16	78.48
0 15 34	834	83.81	0 15 37	68.06	0 16 34	75.53	0 16 37	57.55	0 34 37	64.63
3 15 16	716	79.13	3 15 34	84.42	3 15 37	68.80	3 16 34	76.20	3 16 37	58.43
3 34 37	637	65.41	15 16 34	82.32	15 16 37	66.20	15 34 37	72.44	16 34 37	62.68

classe 0 cl2=7

0 3 15	74.22	0 3 16	72.03	0 3 34	88.83	0 3 37	71.54	0 15 16	71.42		
0 15 34	88.33	0 15 37	70.92	0 16 34	86.50	0 16 37	68.63	0 34 37	86.10		
3 15 16	31.41	3 15 34	60.74	3 15 37	30.28	3 16 34	58.05	3 16 37	24.43		
3 34 37	57.44	15 16 34	57.28	15 16 37	22.55	15 34 37	56.66	16 34 37	53.77		

quatro Variáveis

classe 0 cl2=1

0 3 15 16	53.29	0 3 15 34	55.70	0 3 15 37	48.09	0 3 16 34	49.58	0 3 16 37	40.85	0 3 34 37	43.95
0 15 16 34	56.47	0 15 16 37	48.98	0 15 34 37	51.59	0 16 34 37	44.92	3 15 16 34	58.63	3 15 16 37	51.46
3 15 34 37	53.96	3 16 34 37	47.61	15 16 34 37		54.75					

classe 0 cl2=2

0 3 15 16	69.68	0 3 15 34	76.29	0 3 15 37	68.83	0 3 16 34	72.95	0 3 16 37	65.11	0 3 34 37	72.14
0 15 16 34	71.31	0 15 16 37	63.27	0 15 34 37	70.48	0 16 34 37	66.85	3 15 16 34	52.62	3 15 16 37	41.07
3 15 34 37	51.49	3 16 34 37	46.40	15 16 34 37		43.77					

classe 0 cl2=3

0 3 15 16	44.56	0 3 15 34	63.24	0 3 15 37	40.52	0 3 16 34	58.35	0 3 16 37	32.35	0 3 34 37	55.33
0 15 16 34	63.64	0 15 16 37	41.14	0 15 34 37	60.88	0 16 34 37	55.78	3 15 16 34	62.72	3 15 16 37	39.70
3 15 34 37	59.92	3 16 34 37	54.73	15 16 34 37		60.34					

classe 0 cl2=4

0 3 15 16	74.14	0 3 15 34	78.21	0 3 15 37	62.98	0 3 16 34	69.61	0 3 16 37	51.92	0 3 34 37	57.59
0 15 16 34	85.86	0 15 16 37	72.26	0 15 34 37	76.44	0 16 34 37	67.61	3 15 16 34	82.23	3 15 16 37	67.90
3 15 34 37	72.33	3 16 34 37	62.94	15 16 34 37		80.54					

classe 0 cl2=5

0 3 15 16	106.86	0 3 15 34	82.48	0 3 15 37	79.28	0 3 16 34	85.08	0 3 16 37	81.98	0 3 34 37	45.87
0 15 16 34	104.05	0 15 16 37	101.53	0 15 34 37	75.45	0 16 34 37	78.29	3 15 16 34	107.92	3 15 16 37	105.49
3 15 34 37	80.70	3 16 34 37	83.36	15 16 34 37		102.65					

classe 0 cl2=6

0 3 15 16	89.67	0 3 15 34	94.37	0 3 15 37	80.70	0 3 16 34	87.10	0 3 16 37	72.06	0 3 34 37	77.83
0 15 16 34	92.49	0 15 16 37	78.50	0 15 34 37	83.83	0 16 34 37	75.55	3 15 16 34	93.04	3 15 16 37	79.14
3 15 34 37	84.43	3 16 34 37	76.22	15 16 34 37		82.33					

classe 0 cl2=7

0 3 15 16	74.80	0 3 15 34	91.09	0 3 15 37	74.33	0 3 16 34	89.32	0 3 16 37	72.15	0 3 34 37	88.92
0 15 16 34	88.82	0 15 16 37	71.53	0 15 34 37	88.42	0 16 34 37	86.60	3 15 16 34	61.44	3 15 16 37	31.67
3 15 34 37	60.87	3 16 34 37	58.18	15 16 34 37		57.42					

SEPARAÇÃO ENTRE A CLASSE '1' e as CLASSES '0' '2' '3' '4' '5' '6' e '7'.
As características testadas em função do seu factor potencial de separação são: '2' '4' '30' '33' '39' e '41'

Uma Variável

classe 1 cl2=0

* 2	11.59	* 4	24.72	* 30	3.52	* 33	27.16	* 39	5.91	* 41	16.06

classe 1 cl2=2

* 2	34.23	* 4	4.52	* 30	9.61	* 33	2.25	* 39	3.09	* 41	0.13

classe 1 cl2=3

* 2	18.62	* 4	5.13	* 30	32.17	* 33	20.64	* 39	3.74	* 41	2.18

classe 1 cl2=4

* 2	13.12	* 4	2.66	* 30	1.11	* 33	0.79	* 39	5.46	* 41	0.20

classe 1 cl2=5

* 2	16.19	* 4	7.47	* 30	2.89	* 33	19.56	* 39	4.94	* 41	0.89

classe 1 cl2=6

* 2	29.60	* 4	14.61	* 30	33.70	* 33	1.94	* 39	0.52	* 41	1.90

classe 1 cl2=7

* 2	32.43	* 4	2.48	* 30	10.74	* 33	6.54	* 39	6.39	* 41	1.83

Duas Variáveis

classe 1 cl2=0

2 ->	* 4	27.31	* 30	12.11	* 33	29.53	* 39	13.01	* 41	10.80	
4 ->	* 30	24.97	* 33	36.73	* 39	25.42	* 41	29.48			
30 ->	* 33	27.39	* 39	6.88	* 41	16.44					
33 ->	* 39	27.80	* 41	31.55							
39 ->	* 41	17.11									

classe 1 cl2=2

2 ->	* 4	34.53	* 30	35.55	* 33	34.30	* 39	34.37	* 41	34.23	
4 ->	* 30	10.62	* 33	5.05	* 39	5.48	* 41	4.53			
30 ->	* 33	9.87	* 39	10.09	* 41	9.61					
33 ->	* 39	3.82	* 41	2.26							
39 ->	* 41	3.09									

classe 1 clse 1 cl2=3

2 ->	* 4	19.51	* 30	37.17	* 33	27.80	* 39	18.99	* 41	18.75
4 ->	* 30	32.69	* 33	21.45	* 39	6.93	* 41	6.23		
30 ->	* 33	38.22	* 39	32.38	* 41	32.24				
33 ->	* 39	20.97	* 41	20.75						
39 ->	* 41	4.33								

classe 1 clse 1 cl2=4

2 ->	* 4	13.39	* 30	13.17	* 33	13.15	* 39	14.21	* 41	13.12
4 ->	* 30	2.88	* 33	2.78	* 39	6.07	* 41	2.67		
30 ->	* 33	1.36	* 39	5.57	* 41	1.13				
33 ->	* 39	5.51	* 41	0.82						
39 ->	* 41	5.46								

classe 1 clse 1 cl2=5

2 ->	* 4	17.83	* 30	16.44	* 33	25.39	* 39	16.92	* 41	16.21
4 ->	* 30	8.01	* 33	20.94	* 39	8.95	* 41	7.52		
30 ->	* 33	19.77	* 39	5.72	* 41	3.02				
33 ->	* 39	20.17	* 41	19.58						
39 ->	* 41	5.02								

classe 1 clse 1 cl2=6

2 ->	* 4	33.01	* 30	44.86	* 33	29.67	* 39	29.61	* 41	29.67
4 ->	* 30	36.74	* 33	14.74	* 39	14.62	* 41	14.74		
30 ->	* 33	33.76	* 39	33.71	* 41	33.76				
33 ->	* 39	2.01	* 41	2.72						
39 ->	* 41	1.97								

classe 1 clse 1 cl2=7

2 ->	* 4	32.52	* 30	34.16	* 33	33.08	* 39	33.05	* 41	32.48
4 ->	* 30	11.02	* 33	6.99	* 39	6.85	* 41	3.08		
30 ->	* 33	12.57	* 39	12.50	* 41	10.90				
33 ->	* 39	9.14	* 41	6.79						
39 ->	* 41	6.64								

três Variáve/ariáveis

classe 1 cl2= 1 cl2=0

2 4 30	27.3	27.53	2 4 33	38.51	2 4 39	27.94	2 4 41	31.68	2 30 33	29.74
2 30 39	13.89	13.48	2 30 41	20.11	2 33 39	30.12	2 33 41	33.61	2 39 41	20.67
4 30 33	36.33	36.90	4 30 39	25.66	4 30 41	29.69	4 33 39	37.20	4 33 41	40.08
4 39 41	30.11	30.07	30 33 39	28.02	30 33 41	31.75	30 39 41	17.47	33 39 41	32.10

classe 1 cl2= 1 cl2=2

2 4 30	35.3	35.84	2 4 33	34.60	2 4 39	34.67	2 4 41	34.53	2 30 33	35.63
2 30 39	35.9	35.69	2 30 41	35.55	2 33 39	34.44	2 33 41	34.31	2 39 41	34.37
4 30 33	10.3	10.86	4 30 39	11.06	4 30 41	10.62	4 33 39	5.92	4 33 41	5.06
4 39 41	5.41	5.48	30 33 39	10.34	30 33 41	9.87	30 39 41	10.10	33 39 41	3.83

classe 1 cl2= 1 cl2=3

2 4 30	37.4	37.62	2 4 33	28.40	2 4 39	19.87	2 4 41	19.64	2 30 33	42.51
2 30 39	37.9	37.36	2 30 41	37.23	2 33 39	28.05	2 33 41	27.88	2 39 41	19.12
4 30 33	38.3	38.66	4 30 39	32.90	4 30 41	32.76	4 33 39	21.77	4 33 41	21.56
4 39 41	7.21	7.26	30 33 39	38.40	30 33 41	38.28	30 39 41	32.46	33 39 41	21.09

classe 1 cl2= 1 cl2=4

2 4 30	13.4	13.44	2 4 33	13.41	2 4 39	14.46	2 4 41	13.39	2 30 33	13.19
2 30 39	14.9	14.26	2 30 41	13.17	2 33 39	14.23	2 33 41	13.15	2 39 41	14.21
4 30 33	2.93	2.99	4 30 39	6.17	4 30 41	2.89	4 33 39	6.12	4 33 41	2.78
4 39 41	6.01	6.07	30 33 39	5.62	30 33 41	1.38	30 39 41	5.57	33 39 41	5.52

classe 1 cl2=1 cl2=5

2 4 30	18.1	18.06	2 4 33	26.46	2 4 39	18.50	2 4 41	17.85	2 30 33	25.55
2 30 39	17.19	17.17	2 30 41	16.47	2 33 39	25.86	2 33 41	25.40	2 39 41	16.95
4 30 33	21.03	21.14	4 30 39	9.41	4 30 41	8.06	4 33 39	21.51	4 33 41	20.96
4 39 41	9.001	9.00	30 33 39	20.38	30 33 41	19.79	30 39 41	5.79	33 39 41	20.19

classe 1 cl2=1 cl2=6

2 4 30	47.5	47.18	2 4 33	33.07	2 4 39	33.02	2 4 41	33.07	2 30 33	44.90
2 30 39	44.69	44.86	2 30 41	44.90	2 33 39	29.67	2 33 41	29.73	2 39 41	29.67
4 30 33	36.73	36.79	4 30 39	36.74	4 30 41	36.78	4 33 39	14.75	4 33 41	14.86
4 39 41	14.71	14.75	30 33 39	33.76	30 33 41	33.81	30 39 41	33.76	33 39 41	2.77

classe 1 cl2=1 cl2=7

2 4 30	34.2	34.25	2 4 33	33.17	2 4 39	33.14	2 4 41	32.57	2 30 33	34.78
2 30 39	34.7	34.75	2 30 41	34.21	2 33 39	33.69	2 33 41	33.13	2 39 41	33.10
4 30 33	12.8	12.82	4 30 39	12.74	4 30 41	11.17	4 33 39	9.47	4 33 41	7.23
4 39 41	7.09	7.09	30 33 39	14.10	30 33 41	12.71	30 39 41	12.63	33 39 41	9.32

quatro Variáveis

classe 1 cl2=0										
2 4 30 33	38.67	2 4 30 39	28.16	2 4 30 41	31.87	2 4 33 39	38.96	2 4 33 41	41.73	2 4 39 41	32.22
2 30 33 39	30.32	2 30 33 41	33.80	2 30 39 41	20.96	2 33 39 41	34.13	4 30 33 39	37.37	4 30 33 41	40.24
4 30 39 41	30.27	4 33 39 41	40.52	30 33 39 41		32.29					
classe 1 cl2=2											
2 4 30 33	35.91	2 4 30 39	35.97	2 4 30 41	35.84	2 4 33 39	34.74	2 4 33 41	34.60	2 4 39 41	34.67
2 30 33 39	35.76	2 30 33 41	35.63	2 30 39 41	35.69	2 33 39 41	34.44	4 30 33 39	11.29	4 30 33 41	10.86
4 30 39 41	11.06	4 33 39 41	5.92	30 33 39 41		10.34					
classe 1 cl2=3											
2 4 30 33	42.91	2 4 30 39	37.81	2 4 30 41	37.69	2 4 33 39	28.65	2 4 33 41	28.49	2 4 39 41	19.99
2 30 33 39	42.68	2 30 33 41	42.57	2 30 39 41	37.42	2 33 39 41	28.13	4 30 33 39	38.84	4 30 33 41	38.72
4 30 39 41	32.98	4 33 39 41	21.88	30 33 39 41		38.46					
classe 1 cl2=4											
2 4 30 33	13.46	2 4 30 39	14.50	2 4 30 41	13.44	2 4 33 39	14.48	2 4 33 41	13.42	2 4 39 41	14.46
2 30 33 39	14.28	2 30 33 41	13.20	2 30 39 41	14.26	2 33 39 41	14.24	4 30 33 39	6.22	4 30 33 41	3.00
4 30 39 41	6.18	4 33 39 41	6.13	30 33 39 41		5.63					
classe 1 cl2=5											
2 4 30 33	26.62	2 4 30 39	18.72	2 4 30 41	18.08	2 4 33 39	26.92	2 4 33 41	26.48	2 4 39 41	18.52
2 30 33 39	26.02	2 30 33 41	25.57	2 30 39 41	17.19	2 33 39 41	25.88	4 30 33 39	21.70	4 30 33 41	21.15
4 30 39 41	9.45	4 33 39 41	21.53	30 33 39 41		20.40					
classe 1 cl2=6											
2 4 30 33	47.22	2 4 30 39	47.18	2 4 30 41	47.22	2 4 33 39	33.08	2 4 33 41	33.13	2 4 39 41	33.07
2 30 33 39	44.90	2 30 33 41	44.94	2 30 39 41	44.90	2 33 39 41	29.73	4 30 33 39	36.79	4 30 33 41	36.84
4 30 39 41	36.79	4 33 39 41	14.87	30 33 39 41		33.82					
classe 1 cl2=7											
2 4 30 33	34.87	2 4 30 39	34.84	2 4 30 41	34.30	2 4 33 39	33.78	2 4 33 41	33.22	2 4 39 41	33.19
2 30 33 39	35.36	2 30 33 41	34.83	2 30 39 41	34.80	2 33 39 41	33.74	4 30 33 39	14.32	4 30 33 41	12.95
4 30 39 41	12.87	4 33 39 41	9.64	30 33 39 41		14.22					

SEPARAÇÃO ENTRE A CLASSE '2' e as CLASSES '0' '1' '3' '4' '5' '6' e '7'.
As caracteristicas testadas em função do seu factor potencial de separação são: '5' '7' '10' '15' '16' e '36'

Uma Variável

classe 2 cl2=0									
* 5 6.31	* 7	25.26	* 10	36.26	* 15	25.32	* 16	11.95	* 36	13.10
classe 2 cl2=1									
* 5 28.65	* 7	12.26	* 10	2.36	* 15	10.71	* 16	13.63	* 36	6.21
classe 2 cl2=3									
* 5 21.92	* 7	10.12	* 10	10.72	* 15	5.32	* 16	6.59	* 36	14.52
classe 2 cl2=4									
* 5 51.09	* 7	4.62	* 10	26.07	* 15	27.61	* 16	27.18	* 36	8.53
classe 2 cl2=5									
* 5 50.58	* 7	13.27	* 10	12.97	* 15	43.56	* 16	60.03	* 36	0.95
classe 2 cl2=6									
* 5 7.12	* 7	6.69	* 10	5.38	* 15	28.06	* 16	27.17	* 36	16.35
classe 2 cl2=7									
* 5 1.09	* 7	5.52	* 10	11.42	* 15	5.17	* 16	2.66	* 36	1.66

Duas Variáveis

classe 2 cl2=0									
5 ->	* 7	26.03	* 10	36.80	* 15	26.10	* 16	13.52	* 36	14.54
7 ->	* 10	44.19	* 15	35.77	* 16	27.94	* 36	28.45		
10 ->	* 15	44.22	* 16	38.18	* 36	38.55				
15 ->	* 16	28.00	* 36	28.51						
16 ->	* 36	17.73								
classe 2 cl2=1									
5 ->	* 7	31.17	* 10	28.75	* 15	30.59	* 16	31.73	* 36	29.21
7 ->	* 10	12.49	* 15	16.28	* 16	18.33	* 36	13.74		
10 ->	* 15	10.97	* 16	13.83	* 36	6.64				
15 ->	* 16	17.33	* 36	12.38						
16 ->	* 36	14.97								
classe 2 cl2=3									
5 ->	* 7	24.15	* 10	24.40	* 15	22.56	* 16	22.89	* 36	26.29
7 ->	* 10	14.74	* 15	11.43	* 16	12.07	* 36	17.69		
10 ->	* 15	11.96	* 16	12.58	* 36	18.04				
15 ->	* 16	8.46	* 36	15.46						
16 ->	* 36	15.94								

classe 2 cl2=4 cl2=4

5 ->	*7	*7	51.30	*10	57.36	*15	58.07	*16	57.87	*36	51.80
7 ->	*1	*10	26.47	*15	28.00	*16	27.57	*36	9.70		
10 ->	*1	*15	37.97	*16	37.66	*36	27.43				
15 ->	*1	*16	38.74	*36	28.90						
16 ->	*3	*36	28.49								

classe 2 cl2=4 cl2=5

5 ->	*7	*7	52.29	*10	52.21	*15	66.75	*16	78.49	*36	50.59
7 ->	*1	*10	18.56	*15	45.54	*16	61.48	*36	13.31		
10 ->	*1	*15	45.45	*16	61.41	*36	13.00				
15 ->	*1	*16	74.17	*36	43.57						
16 ->	*3	*36	60.03								

classe 2 cl2=4 cl2=6

5 ->	*7	*7	9.76	*10	8.92	*15	28.95	*16	28.09	*36	17.83
7 ->	*1	*10	8.58	*15	28.85	*16	27.98	*36	17.66		
10 ->	*1	*15	28.58	*16	27.70	*36	17.21				
15 ->	*1	*16	39.06	*36	32.48						
16 ->	*3	*36	31.71								

classe 2 cl2=4 cl2=7

5 ->	*7	*7	5.62	*10	11.47	*15	5.28	*16	2.88	*36	1.99
7 ->	*1	*10	12.68	*15	7.56	*16	6.12	*36	5.76		
10 ->	*1	*15	12.53	*16	11.72	*36	11.54				
15 ->	*1	*16	5.81	*36	5.43						
16 ->	*3	*36	3.14								

três Variáveis

classe 2 cl2=4 cl2=0

5 7 10	44	44.63	5 7 15	36.32	5 7 16	28.65	5 7 36	29.15	5 10 15	44.67
5 10 16	38	38.69	5 10 36	39.06	5 15 16	28.70	5 15 36	29.20	5 16 36	18.82
7 10 15	50	50.93	7 10 16	45.77	7 10 36	46.09	7 15 16	37.71	7 15 36	38.09
7 16 36	30	30.86	10 15 16	45.81	10 15 36	46.12	10 16 36	40.36	15 16 36	30.91

classe 2 cl2=4 cl2=1

5 7 10	31	31.26	5 7 15	32.96	5 7 16	34.02	5 7 36	31.78	5 10 15	30.68
5 10 16	31	31.82	5 10 36	29.41	5 15 16	33.49	5 15 36	31.21	5 16 36	32.33
7 10 15	16	16.45	7 10 16	18.48	7 10 36	13.94	7 15 16	21.23	7 15 36	17.43
7 16 36	19	19.35	10 15 16	17.49	10 15 36	12.60	10 16 36	15.16	15 16 36	18.41

classe 2 cl2=4 cl2=3

5 7 10	26	26.42	5 7 15	24.72	5 7 16	25.03	5 7 36	28.17	5 10 15	24.98
5 10 16	25	25.28	5 10 36	28.39	5 15 16	23.50	5 15 36	26.83	5 16 36	27.11
7 10 15	15	15.67	7 10 16	16.14	7 10 36	20.69	7 15 16	13.19	7 15 36	18.48
7 16 36	18	18.88	10 15 16	13.66	10 15 36	18.81	10 16 36	19.21	15 16 36	16.80

classe 2 cl2=4 cl2=4

5 7 10	57	57.54	5 7 15	58.26	5 7 16	58.05	5 7 36	52.00	5 10 15	63.66
5 10 16	63	63.47	5 10 36	57.99	5 15 16	64.12	5 15 36	58.70	5 16 36	58.49
7 10 15	38	38.25	7 10 16	37.94	7 10 36	27.81	7 15 16	39.02	7 15 36	29.27
7 16 36	28	28.86	10 15 16	46.70	10 15 36	38.92	10 16 36	38.61	15 16 36	39.67

classe 2 cl2=4 cl2=5

5 7 10	53	53.87	5 7 15	68.06	5 7 16	79.61	5 7 36	52.30	5 10 15	68.00
5 10 16	79	79.56	5 10 36	52.22	5 15 16	89.77	5 15 36	66.76	5 16 36	78.50
7 10 15	47	47.35	7 10 16	62.83	7 10 36	18.58	7 15 16	75.35	7 15 36	45.55
7 16 36	61	61.48	10 15 16	75.29	10 15 36	45.46	10 16 36	61.42	15 16 36	74.17

classe 2 cl2=4 cl2=6

5 7 10	11	11.15	5 7 15	29.71	5 7 16	28.87	5 7 36	19.04	5 10 15	29.45
5 10 16	28	28.60	5 10 36	18.63	5 15 16	39.70	5 15 36	33.25	5 16 36	32.50
7 10 15	29	29.35	7 10 16	28.49	7 10 36	18.47	7 15 16	39.63	7 15 36	33.16
7 16 36	32	32.41	10 15 16	39.43	10 15 36	32.92	10 16 36	32.16	15 16 36	42.34

classe 2 cl2=4 cl2=7

5 7 10	12	12.73	5 7 15	7.64	5 7 16	6.22	5 7 36	5.86	5 10 15	12.58
5 10 16	11	11.77	5 10 36	11.59	5 15 16	5.92	5 15 36	5.54	5 16 36	3.32
7 10 15	13	13.69	7 10 16	12.96	7 10 36	12.79	7 15 16	8.01	7 15 36	7.74
7 16 36	6.3	6.35	10 15 16	12.81	10 15 36	12.64	10 16 36	11.84	15 16 36	6.05

quatro Variáveis

classe 2 cl2=0

5 7 10 15	51.32	5 7 10 16	46.21	5 7 10 36	46.52	5 7 15 16	38.23	5 7 15 36	38.61	5 7 16 36	31.50
5 10 15 16	46.24	5 10 15 36	46.55	5 10 16 36	40.85	5 15 16 36	31.55	7 10 15 16	52.31	7 10 15 36	52.59
7 10 16 36	47.61	7 15 16 36	39.92	10 15 16 36	47.65						

classe 2 cl2=1

5 7 10 15	33.04	5 7 10 16	34.10	5 7 10 36	31.87	5 7 15 16	35.66	5 7 15 36	33.54	5 7 16 36	34.58
5 10 15 16	33.57	5 10 15 36	31.30	5 10 16 36	32.42	5 15 16 36	34.06	7 10 15 16	21.36	7 10 15 36	17.58
7 10 16 36	19.50	7 15 16 36	22.12	10 15 16 36	18.56						

classe 2 cl2=3

5 7 10 15	26.95	5 7 10 16	27.23	5 7 10 36	30.14	5 7 15 16	25.59	5 7 15 36	28.67	5 7 16 36	28.93
5 10 15 16	25.83	5 10 15 36	28.89	5 10 16 36	29.15	5 15 16 36	27.62	7 10 15 16	17.00	7 10 15 36	21.36
7 10 16 36	21.71	7 15 16 36	19.61	10 15 16 36	19.93						

classe 2 cl2=4

5 7 10 15	63.82	5 7 10 16	63.64	5 7 10 36	58.17	5 7 15 16	64.29	5 7 15 36	58.88	5 7 16 36	58.68
5 10 15 16	69.22	5 10 15 36	64.22	5 10 16 36	64.04	5 15 16 36	64.68	7 10 15 16	46.92	7 10 15 36	39.19
7 10 16 36	38.89	7 15 16 36	39.94	10 15 16 36	47.47						

classe 2 cl2=5

5 7 10 15	69.28	5 7 10 16	80.66	5 7 10 36	53.88	5 7 15 16	90.75	5 7 15 36	68.07	5 7 16 36	79.61
5 10 15 16	90.70	5 10 15 36	68.01	5 10 16 36	79.56	5 15 16 36	89.78	7 10 15 16	76.45	7 10 15 36	47.36
7 10 16 36	62.84	7 15 16 36	75.35	10 15 16 36	75.30						

classe 2 cl2=6

5 7 10 15	30.20	5 7 10 16	29.37	5 7 10 36	19.79	5 7 15 16	40.26	5 7 15 36	33.91	5 7 16 36	33.18
5 10 15 16	40.07	5 10 15 36	33.68	5 10 16 36	32.94	5 15 16 36	42.94	7 10 15 16	39.99	7 10 15 36	33.59
7 10 16 36	32.85	7 15 16 36	42.87	10 15 16 36	42.69						

classe 2 cl2=7

5 7 10 15	13.74	5 7 10 16	13.00	5 7 10 36	12.84	5 7 15 16	8.09	5 7 15 36	7.82	5 7 16 36	6.44
5 10 15 16	12.86	5 10 15 36	12.69	5 10 16 36	11.89	5 15 16 36	6.14	7 10 15 16	13.95	7 10 15 36	13.79
7 10 16 36	13.06	7 15 16 36	8.18	10 15 16 36	12.92						

SEPARAÇÃO ENTRE A CLASSE '3' e as CLASSES '0' '1' '2' '4' '5' '6' e '7'.
As características testadas em função do seu factor potencial de separação são: '11' '31' '32' '35' e '40'

Uma Variável

classe 3 cl2=0

* 11	2.88	* 31	29.24	* 32	14.87	* 35	24.41	* 38	4.36	* 40	0.93

classe 3 cl2=1

* 11	7.48	* 31	7.08	* 32	17.12	* 35	16.58	* 38	1.04	* 40	1.43

classe 3 cl2=2

* 11	9.34	* 31	1.69	* 32	21.73	* 35	16.05	* 38	1.79	* 40	1.50

classe 3 cl2=4

* 11	35.94	* 31	1.83	* 32	11.45	* 35	17.09	* 38	6.96	* 40	0.86

classe 3 cl2=5

* 11	22.34	* 31	8.49	* 32	17.69	* 35	9.91	* 38	3.37	* 40	0.32

classe 3 cl2=6

* 11	11.02	* 31	5.86	* 32	12.31	* 35	2.81	* 38	2.93	* 40	1.10

classe 3 cl2=7

* 11	0.30	* 31	1.44	* 32	12.63	* 35	9.65	* 38	7.12	* 40	0.02

Duas Variáveis

classe 3 cl2=0

11 ->	* 31	29.38	* 32	15.15	* 35	24.57	* 38	5.23	* 40	3.02	
31 ->	* 32	32.81	* 35	38.09	* 38	29.56	* 40	29.26			
32 ->	* 35	28.58	* 38	15.50	* 40	14.90					
35 ->	* 38	24.79	* 40	24.42							
38 ->	* 40	4.46									

classe 3 cl2=1

11 ->	* 31	10.30	* 32	18.68	* 35	18.19	* 38	7.55	* 40	7.62	
31 ->	* 32	18.52	* 35	18.02	* 38	7.15	* 40	7.22			
32 ->	* 35	23.83	* 38	17.15	* 40	17.18					
35 ->	* 38	16.61	* 40	16.64							
38 ->	* 40	1.77									

classe 3 cl2=2

11 ->	* 31	9.50	* 32	23.65	* 35	18.57	* 38	9.51	* 40	9.46	
31 ->	* 32	21.80	* 35	16.14	* 38	2.46	* 40	2.26			
32 ->	* 35	27.02	* 38	21.80	* 40	21.78					
35 ->	* 38	16.15	* 40	16.12							
38 ->	* 40	2.33									

classe 3 cl2=4

11 ->	* 31	35.99	* 32	37.72	* 35	39.80	* 38	36.61	* 40	35.95	
31 ->	* 32	11.60	* 35	17.19	* 38	7.20	* 40	2.02			
32 ->	* 35	20.58	* 38	13.40	* 40	11.48					
35 ->	* 38	18.46	* 40	17.11							
38 ->	* 40	7.01									

classe 3 cl2=5

11 ->	* 31	23.90	* 32	28.50	* 35	24.44	* 38	22.60	* 40	22.34	
31 ->	* 32	19.62	* 35	13.05	* 38	9.13	* 40	8.49			
32 ->	* 35	20.28	* 38	18.01	* 40	17.70					
35 ->	* 38	10.47	* 40	9.92							
38 ->	* 40	3.39									

classe 3 cl1 3 cl2=6

11 ->	* 31	12.48	* 32	16.52	* 35	11.37	* 38	11.41	* 40	11.08
31 ->	* 32	13.63	* 35	6.50	* 38	6.55	* 40	5.96		
32 ->	* 35	12.62	* 38	12.65	* 40	12.36				
35 ->	* 38	4.06	* 40	3.02						
38 ->	* 40	3.13								

classe 3 cl1 3 cl2=7

11 ->	* 31	1.47	* 32	12.63	* 35	9.65	* 38	7.13	* 40	0.30
31 ->	* 32	12.71	* 35	9.75	* 38	7.26	* 40	1.44		
32 ->	* 35	15.89	* 38	14.50	* 40	12.63				
35 ->	* 38	11.99	* 40	9.65						
38 ->	* 40	7.12								

três Variáveis

classe 3 cl1 3 cl2=0

11 31 32	32	32.93	11 31 35	38.20	11 31 38	29.70	11 31 40	29.40	11 32 35	28.72
11 32 38	38	15.76	11 32 40	15.18	11 35 38	24.96	11 35 40	24.59	11 38 40	5.31
31 32 35	35	40.89	31 32 38	33.09	31 32 40	32.82	31 35 38	38.34	31 35 40	38.10
31 38 40	40	29.58	32 35 38	28.91	32 35 40	28.60	32 38 40	15.53	35 38 40	24.81

classe 3 cl1 3 cl2=1

11 31 32	32	19.97	11 31 35	19.52	11 31 38	10.35	11 31 40	10.40	11 32 35	24.97
11 32 38	38	18.71	11 32 40	18.73	11 35 38	18.22	11 35 40	18.24	11 38 40	7.69
31 32 35	35	24.86	31 32 38	18.55	31 32 40	18.58	31 35 38	18.05	31 35 40	18.08
31 38 40	40	7.29	32 35 38	23.85	32 35 40	23.87	32 38 40	17.21	35 38 40	16.67

classe 3 cl1 3 cl2=2

11 31 32	32	23.71	11 31 35	18.65	11 31 38	9.66	11 31 40	9.61	11 32 35	28.59
11 32 38	38	23.72	11 32 40	23.70	11 35 38	18.66	11 35 40	18.63	11 38 40	9.63
31 32 35	35	27.07	31 32 38	21.87	31 32 40	21.85	31 35 38	16.24	31 35 40	16.21
31 38 40	40	2.88	32 35 38	27.08	32 35 40	27.06	32 38 40	21.86	35 38 40	16.22

classe 3 cl1 3 cl2=4

11 31 32	32	37.77	11 31 35	39.84	11 31 38	36.65	11 31 40	36.00	11 32 35	41.41
11 32 38	38	38.36	11 32 40	37.73	11 35 38	40.40	11 35 40	39.81	11 38 40	36.62
31 32 35	35	20.66	31 32 38	13.53	31 32 40	11.63	31 35 38	18.55	31 35 40	17.21
31 38 40	40	7.25	32 35 38	21.72	32 35 40	20.59	32 38 40	13.43	35 38 40	18.48

classe 3 cl1 3 cl2=5

11 31 32	32	29.74	11 31 35	25.87	11 31 38	24.14	11 31 40	23.90	11 32 35	30.17
11 32 38	38	28.70	11 32 40	28.50	11 35 38	24.67	11 35 40	24.44	11 38 40	22.60
31 32 35	35	21.98	31 32 38	19.91	31 32 40	19.62	31 35 38	13.48	31 35 40	13.05
31 38 40	40	9.14	32 35 38	20.56	32 35 40	20.28	32 38 40	18.01	35 38 40	10.47

classe 3 cl1 3 cl2=6

11 31 32	32	17.53	11 31 35	12.79	11 31 38	12.82	11 31 40	12.53	11 32 35	16.76
11 32 38	38	16.78	11 32 40	16.56	11 35 38	11.75	11 35 40	11.43	11 38 40	11.46
31 32 35	35	13.92	31 32 38	13.94	31 32 40	13.67	31 35 38	7.13	31 35 40	6.59
31 38 40	40	6.64	32 35 38	12.96	32 35 40	12.67	32 38 40	12.70	35 38 40	4.21

classe 3 cl1 3 cl2=7

11 31 32	32	12.71	11 31 35	9.76	11 31 38	7.27	11 31 40	1.47	11 32 35	15.89
11 32 38	38	14.50	11 32 40	12.63	11 35 38	11.99	11 35 40	9.65	11 38 40	7.13
31 32 35	35	15.96	31 32 38	14.57	31 32 40	12.71	31 35 38	12.08	31 35 40	9.75
31 38 40	40	7.26	32 35 38	17.41	32 35 40	15.89	32 38 40	14.50	35 38 40	11.99

quatro Variáveis

classe 3 cl2=0

11 31 32 35	40.99	11 31 32 38	33.22	11 31 32 40	32.94	11 31 35 38	38.44	11 31 35 40	38.21	11 31 38 40	29.72
11 32 35 38	29.05	11 32 35 40	28.74	11 32 38 40	15.79	11 35 38 40	24.98	31 32 35 38	41.12	31 32 35 40	40.90
31 32 38 40	33.11	31 35 38 40	38.35	32 35 38 40	28.93						

classe 3 cl2=1

11 31 32 35	25.96	11 31 32 38	20.00	11 31 32 40	20.03	11 31 35 38	19.54	11 31 35 40	19.57	11 31 38 40	10.45
11 32 35 38	25.00	11 32 35 40	25.01	11 32 38 40	18.76	11 35 38 40	18.27	31 32 35 38	24.88	31 32 35 40	24.90
31 32 38 40	18.61	31 35 38 40	18.11	32 35 38 40	23.89						

classe 3 cl2=2

11 31 32 35	28.64	11 31 32 38	23.78	11 31 32 40	23.76	11 31 35 38	18.74	11 31 35 40	18.71	11 31 38 40	9.78
11 32 35 38	28.64	11 32 35 40	28.63	11 32 38 40	23.77	11 35 38 40	18.72	31 32 35 38	27.13	31 32 35 40	27.11
31 32 38 40	21.92	31 35 38 40	16.31	32 35 38 40	27.12						

classe 3 cl2=4

11 31 32 35	41.45	11 31 32 38	38.40	11 31 32 40	37.78	11 31 35 38	40.44	11 31 35 40	39.85	11 31 38 40	36.66
11 32 35 38	41.99	11 32 35 40	41.42	11 32 38 40	38.37	11 35 38 40	40.41	31 32 35 38	21.80	31 32 35 40	20.67
31 32 38 40	13.55	31 35 38 40	18.57	32 35 38 40	21.74						

classe 3 cl2=5

11 31 32 35	31.34	11 31 32 38	29.93	11 31 32 40	29.74	11 31 35 38	26.09	11 31 35 40	25.87	11 31 38 40	24.14
11 32 35 38	30.36	11 32 35 40	30.17	11 32 38 40	28.70	11 35 38 40	24.68	31 32 35 38	22.24	31 32 35 40	21.99
31 32 38 40	19.91	31 35 38 40	13.48	32 35 38 40	20.56						

classe 3 cl2=6

11 31 32 35	17.75	11 31 32 38	17.77	11 31 32 40	17.56	11 31 35 38	13.13	11 31 35 40	12.84	11 31 38 40	12.87
11 32 35 38	17.01	11 32 35 40	16.79	11 32 38 40	16.82	11 35 38 40	11.80	31 32 35 38	14.22	31 32 35 40	13.96
31 32 38 40	13.99	31 35 38 40	7.21	32 35 38 40	13.01						

classe 3 cl2=7

11 31 32 35	15.96	11 31 32 38	14.57	11 31 32 40	12.71	11 31 35 38	12.08	11 31 35 40	9.76	11 31 38 40	7.27
11 32 35 38	17.42	11 32 35 40	15.89	11 32 38 40	14.50	11 35 38 40	11.99	31 32 35 38	17.47	31 32 35 40	15.96
31 32 38 40	14.57	31 35 38 40	12.08	32 35 38 40	17.41						

SEPARAÇÃO ENTRE A CLASSE '4' e as CLASSES '0' '1' '2' '3' '5' '6' e '7'.
As caracteristicas testadas em função do seu factor potencial de separação são: '12' '20' '24' '30' '32' e '34'.

Uma Variável

classe 4 cl2=0

* 12	66.49	* 20	71.83	* 24	77.17	* 30	4.63	* 32	3.42	* 34	46.39

classe 4 cl2=1

* 12	8.02	* 20	11.21	* 24	10.86	* 30	1.11	* 32	5.66	* 34	16.10

classe 4 cl2=2

* 12	13.16*	20	10.92	* 24	7.95	* 30	10.72	* 32	10.28	* 34	13.12

classe 4 cl2=3

* 12	12.89	* 20	15.01	* 24	12.34	* 30	33.28	* 32	11.45	* 34	2.17

classe 4 cl2=5

* 12	4.14	* 20	3.99	* 24	3.28	* 30	4.00	* 32	6.24	* 34	22.63

classe 4 cl2=6

* 12	10.25	* 20	4.90	* 24	3.28	* 30	34.81	* 32	0.85	* 34	2.55

classe 4 cl2=7

* 12	14.04	* 20	16.32	* 24	12.23	* 30	11.85	* 32	1.18	* 34	6.42

Duas Variáveis

classe 4 cl2=0

12 ->	* 20	97.88	* 24	101.86	* 30	66.65	* 32	66.58	* 34	81.07	
20 ->	* 24	105.42	* 30	71.97	* 32	71.91	* 34	85.50			
24 ->	* 30	77.31	* 32	77.24	* 34	90.04					
30 ->	* 32	5.76	* 34	46.62							
32 ->	* 34	46.52									

classe 4 cl2=1

12 ->	* 20	13.78	* 24	13.50	* 30	8.10	* 32	9.82	* 34	17.99	
20 ->	* 24	15.61	* 30	11.26	* 32	12.56	* 34	19.62			
24 ->	* 30	10.92	* 32	12.25	* 34	19.42					
30 ->	* 32	5.77	* 34	16.14							
32 ->	* 34	17.07									

classe 4 cl2=2

12 ->	* 20	17.10	* 24	15.38	* 30	16.98	* 32	16.70	* 34	18.58	
20 ->	* 24	13.51	* 30	15.31	* 32	15.00	* 34	17.07			
24 ->	* 30	13.35	* 32	12.99	* 34	15.34					
30 ->	* 32	14.85	* 34	16.94							
32 ->	* 34	16.67									

classe 4 cl2=3

12 ->	* 20	19.78	* 24	17.84	* 30	35.69	* 32	17.24	* 34	13.07	
20 ->	* 24	19.43	* 30	36.51	* 32	18.88	* 34	15.17			
24 ->	* 30	35.49	* 32	16.83	* 34	12.53					
30 ->	* 32	35.19	* 34	33.35							
32 ->	* 34	11.66									

classe 4 cl2=5

12 ->	* 20	5.75	* 24	5.28	* 30	5.75	* 32	7.49	* 34	23.01	
20 ->	* 24	5.16	* 30	5.65	* 32	7.41	* 34	22.98			
24 ->	* 30	5.17	* 32	7.05	* 34	22.87					
30 ->	* 32	7.41	* 34	22.98							
32 ->	* 34	23.48									

classe 4 cl2= cl2=6

12 ->	* 2	* 20	11.36	* 24	10.76	* 30	36.29	* 32	10.28	* 34	10.56
20 ->	* 2	* 24	5.89	* 30	35.16	* 32	4.97	* 34	5.52		
24 ->	* 3	* 30	34.97	* 32	3.39	* 34	4.16				
30 ->	* 3	* 32	34.83	* 34	34.91						
32 ->	* 3	* 34	2.69								

classe 4 cl2= cl2=7

12 ->	* 2	* 20	21.53	* 24	18.62	* 30	18.38	* 32	14.09	* 34	15.44
20 ->	* 2	* 24	20.39	* 30	20.17	* 32	16.36	* 34	17.54		
24 ->	* 3	* 30	17.03	* 32	12.28	* 34	13.81				
30 ->	* 3	* 32	11.91	* 34	13.48						
32 ->	* 3	* 34	6.52								

três Variáveriáveis

classe 4 cl2= cl2=0

12 20 24	4	124.64	12 20 30	97.99	12 20 32	97.94	12 20 34	108.31	12 24 30	101.97
12 24 32	2	101.92	12 24 34	111.93	12 30 32	66.74	12 30 34	81.21	12 32 34	81.15
20 24 30	0	105.52	20 24 32	105.48	20 24 34	115.18	20 30 32	72.06	20 30 34	85.63
20 32 34	4	85.57	24 30 32	77.38	24 30 34	90.16	24 32 34	90.10	30 32 34	46.75

classe 4 cl2= cl2=1

12 20 24	4	17.55	12 20 30	13.83	12 20 32	14.90	12 20 34	21.20	12 24 30	13.55
12 24 32	2	14.64	12 24 34	21.02	12 30 32	9.88	12 30 34	18.03	12 32 34	18.86
20 24 30	0	15.64	20 24 32	16.60	20 24 34	22.42	20 30 32	12.60	20 30 34	19.65
20 32 34	4	20.42	24 30 32	12.30	24 30 34	19.46	24 32 34	20.23	30 32 34	17.11

classe 4 cl2= cl2=2

12 20 24	4	18.86	12 20 30	20.19	12 20 32	19.95	12 20 34	21.56	12 24 30	18.75
12 24 32	2	18.50	12 24 34	20.21	12 30 32	19.85	12 30 34	21.46	12 32 34	21.24
20 24 30	0	17.25	20 24 32	16.97	20 24 34	18.83	20 30 32	18.44	20 30 34	20.16
20 32 34	4	19.93	24 30 32	16.85	24 30 34	18.72	24 32 34	18.46	30 32 34	19.82

classe 4 cl2= cl2=3

12 20 24	4	23.32	12 20 30	38.71	12 20 32	22.86	12 20 34	19.90	12 24 30	37.76
12 24 32	2	21.20	12 24 34	17.97	12 30 32	37.48	12 30 34	35.75	12 32 34	17.38
20 24 30	0	38.54	20 24 32	22.56	20 24 34	19.55	20 30 32	38.26	20 30 34	36.57
20 32 34	4	19.01	24 30 32	37.29	24 30 34	35.56	24 32 34	16.97	30 32 34	35.26

classe 4 cl2= cl2=5

12 20 24	4	6.62	12 20 30	7.00	12 20 32	8.48	12 20 34	23.35	12 24 30	6.62
12 24 32	2	8.17	12 24 34	23.24	12 30 32	8.49	12 30 34	23.35	12 32 34	23.84
20 24 30	0	6.53	20 24 32	8.10	20 24 34	23.21	20 30 32	8.42	20 30 34	23.33
20 32 34	4	23.81	24 30 32	8.10	24 30 34	23.22	24 32 34	23.70	30 32 34	23.81

classe 4 cl2= cl2=6

12 20 24	4	11.82	12 20 30	36.62	12 20 32	11.39	12 20 34	11.64	12 24 30	36.44
12 24 32	2	10.79	12 24 34	11.06	12 30 32	36.30	12 30 34	36.38	12 32 34	10.59
20 24 30	0	35.31	20 24 32	5.96	20 24 34	6.42	20 30 32	35.17	20 30 34	35.25
20 32 34	4	5.59	24 30 32	34.98	24 30 34	35.06	24 32 34	4.24	30 32 34	34.92

classe 4 cl2= cl2=7

12 20 24	4	24.76	12 20 30	24.58	12 20 32	21.56	12 20 34	22.47	12 24 30	22.07
12 24 32	2	18.66	12 24 34	19.69	12 30 32	18.41	12 30 34	19.46	12 32 34	15.48
20 24 30	0	23.59	20 24 32	20.43	20 24 34	21.38	20 30 32	20.20	20 30 34	21.16
20 32 34	4	17.57	24 30 32	17.07	24 30 34	18.20	24 32 34	13.86	30 32 34	13.53

quatro Variáveis

classe 4 cl2=0

12 20 24 30	124.72	12 20 24 32	124.68	12 20 24 34	132.99	12 20 30 32	98.05	12 20 30 34	108.41	12 20 32 34	108.37
12 24 30 32	102.02	12 24 30 34	112.02	12 24 32 34	111.98	12 30 32 34	81.28	20 24 30 32	105.58	20 24 30 34	115.27
20 24 32 34	115.23	20 30 32 34	85.70	24 30 32 34	90.22						

classe 4 cl2=1

12 20 24 30	17.58	12 20 24 32	18.44	12 20 24 34	23.82	12 20 30 32	14.94	12 20 30 34	21.23	12 20 32 34	21.94
12 24 30 32	14.68	12 24 30 34	21.05	12 24 32 34	21.77	12 30 32 34	18.90	20 24 30 32	16.64	20 24 30 34	22.45
20 24 32 34	23.13	20 30 32 34	20.45	24 30 32 34	20.26						

classe 4 cl2=2

12 20 24 30	21.70	12 20 24 32	21.48	12 20 24 34	22.98	12 20 30 32	22.65	12 20 30 34	24.08	12 20 32 34	23.88
12 24 30 32	21.38	12 24 30 34	22.88	12 24 32 34	22.68	12 30 32 34	23.79	20 24 30 32	20.08	20 24 30 34	21.67
20 24 32 34	21.45	20 30 32 34	22.63	24 30 32 34	21.35						

classe 4 cl2=3

12 20 24 30	40.63	12 20 24 32	25.98	12 20 24 34	23.42	12 20 30 32	40.37	12 20 30 34	38.78	12 20 32 34	22.96
12 24 30 32	39.46	12 24 30 34	37.82	12 24 32 34	21.31	12 30 32 34	37.54	20 24 30 32	40.20	20 24 30 34	38.60
20 24 32 34	22.66	20 30 32 34	38.32	24 30 32 34	37.36						

classe 4 cl2=5

12 20 24 30	7.73	12 20 24 32	9.09	12 20 24 34	23.58	12 20 30 32	9.38	12 20 30 34	23.69	12 20 32 34	24.17
12 24 30 32	9.10	12 24 30 34	23.58	12 24 32 34	24.06	12 30 32 34	24.17	20 24 30 32	9.03	20 24 30 34	23.56
20 24 32 34	24.04	20 30 32 34	24.15	24 30 32 34	24.04						

classe 4 cl2=6

12 20 24 30	36.77	12 20 24 32	11.85	12 20 24 34	12.09	12 20 30 32	36.63	12 20 30 34	36.71	12 20 32 34	11.67
12 24 30 32	36.45	12 24 30 34	36.53	12 24 32 34	11.09	12 30 32 34	36.39	20 24 30 32	35.32	20 24 30 34	35.40
20 24 32 34	6.48	20 30 32 34	35.26	24 30 32 34	35.07						

classe 4 cl2=7

12 20 24 30	27.45	12 20 24 32	24.79	12 20 24 34	25.58	12 20 30 32	24.60	12 20 30 34	25.40	12 20 32 34	22.50
12 24 30 32	22.10	12 24 30 34	22.99	12 24 32 34	19.73	12 30 32 34	19.50	20 24 30 32	23.61	20 24 30 34	24.44
20 24 32 34	21.41	20 30 32 34	21.20	24 30 32 34	18.23						

SEPARAÇÃO ENTRE A CLASSE '5' e as CLASSES '0' '1' '2' '3' '4' '6' e '7'.
As características testadas em função do seu factor potencial de separação são: '12' '17' 19' '23' '26' e '28'

Uma Variável

classe 5 cl2=0											
* 12	62.35	* 17	71.11	* 19	79.96	* 23	58.39	* 26	28.03	* 28	64.01
classe 5 cl2=1											
* 12	3.89	* 17	13.13	* 19	19.35	* 23	9.14	* 26	0.48	* 28	8.21
classe 5 cl2=2											
* 12	9.03	* 17	29.79	* 19	28.37	* 23	27.43	* 26	0.63	* 28	27.64
classe 5 cl2=3											
* 12	8.75	* 17	19.17	* 19	21.96	* 23	7.71	* 26	0.41	* 28	7.93
classe 5 cl2=4											
* 12	4.14	* 17	5.37	* 19	2.56	* 23	0.70	* 26	0.07	* 28	0.32
classe 5 cl2=6											
* 12	6.11	* 17	10.00	* 19	4.78	* 23	19.71	* 26	0.11	* 28	20.67
classe 5 cl2=7											
* 12	9.91	* 17	30.83	* 19	30.22	* 23	30.69	* 26	0.69	* 28	30.83

Duas Variáveis

classe 5 cl2=0									
12 -> * 17	94.58	* 19	101.40	* 23	85.43	* 26	68.36	* 28	89.36
17 -> * 19	107.01	* 23	92.01	* 26	76.44	* 28	95.68		
19 -> * 23	99.01	* 26	84.73	* 28	102.43				
23 -> * 26	64.77	* 28	86.65						
26 -> * 28	69.88								
classe 5 cl2=1									
12 -> * 17	13.69	* 19	19.73	* 23	9.94	* 26	3.92	* 28	9.09
17 -> * 19	23.38	* 23	16.00	* 26	13.13	* 28	15.48		
19 -> * 23	21.40	* 26	19.35	* 28	21.02				
23 -> * 26	9.16	* 28	12.29						
26 -> * 28	8.23								
classe 5 cl2=2									
12 -> * 17	31.13	* 19	29.77	* 23	28.88	* 26	9.05	* 28	29.08
17 -> * 19	41.14	* 23	40.50	* 26	29.80	* 28	40.64		
19 -> * 23	39.46	* 26	28.38	* 28	39.61				
23 -> * 26	27.44	* 28	38.94						
26 -> * 28	27.65								
classe 5 cl2=3									
12 -> * 17	21.07	* 19	23.64	* 23	11.67	* 26	8.76	* 28	11.81
17 -> * 19	29.15	* 23	20.66	* 26	19.17	* 28	20.74		
19 -> * 23	23.27	* 26	21.96	* 28	23.34				
23 -> * 26	7.73	* 28	11.06						
26 -> * 28	7.94								
classe 5 cl2=4									
12 -> * 17	6.78	* 19	4.86	* 23	4.19	* 26	4.14	* 28	4.15
17 -> * 19	5.95	* 23	5.42	* 26	5.37	* 28	5.38		
19 -> * 23	2.65	* 26	2.56	* 28	2.58				
23 -> * 26	0.70	* 28	0.77						
26 -> * 28	0.33								
classe 5 cl2=6									
12 -> * 17	11.72	* 19	7.76	* 23	20.64	* 26	6.11	* 28	21.55
17 -> * 19	11.08	* 23	22.11	* 26	10.00	* 28	22.96		
19 -> * 23	20.29	* 26	4.78	* 28	21.21				
23 -> * 26	19.71	* 28	28.56						
26 -> * 28	20.67								

classe 5 cl2=cl2=7

12 ->	* 17 * 17	32.39	* 19	31.80	* 23	32.25	* 26	9.93	* 28	32.39
17 ->	* 19 * 19	43.17	* 23	43.50	* 26	30.84	* 28	43.60		
19 ->	* 23 * 23	43.07	* 26	30.23	* 28	43.17				
23 ->	* 26 * 26	30.70	* 28	43.50						
26 ->	* 28 * 28	30.84								

três Variáveiáveis

classe 5 cl2=cl2=0

12 17 19	123.85	12 17 23	111.15	12 17 26	98.64	12 17 28	114.20	12 19 23	117.01
12 19 26	105.20	12 19 28	119.91	12 23 26	89.91	12 23 28	106.75	12 26 28	93.66
17 19 23	121.90	17 19 26	110.62	17 19 28	124.69	17 23 26	96.19	17 23 28	112.09
17 26 28	99.70	19 23 26	102.90	19 23 28	117.90	19 26 28	106.19	23 26 28	91.07

classe 5 cl2=cl2=1

12 17 19	23.70	12 17 23	16.46	12 17 26	13.70	12 17 28	15.96	12 19 23	21.75
12 19 26	19.74	12 19 28	21.38	12 23 26	9.95	12 23 28	12.89	12 26 28	9.10
17 19 23	25.10	17 19 26	23.38	17 19 28	24.78	17 23 26	16.00	17 23 28	17.98
17 26 28	15.49	19 23 26	21.40	19 23 28	22.92	19 26 28	21.02	23 26 28	12.30

classe 5 cl2=cl2=2

12 17 19	42.12	12 17 23	41.49	12 17 26	31.14	12 17 28	41.63	12 19 23	40.48
12 19 26	29.78	12 19 28	40.63	12 23 26	28.88	12 23 28	39.97	12 26 28	29.09
17 19 23	49.44	17 19 26	41.14	17 19 28	49.56	17 23 26	40.50	17 23 28	49.03
17 26 28	40.65	19 23 26	39.47	19 23 28	48.18	19 26 28	39.62	23 26 28	38.95

classe 5 cl2=cl2=3

12 17 19	30.43	12 17 23	22.44	12 17 26	21.07	12 17 28	22.51	12 19 23	24.86
12 19 26	23.64	12 19 28	24.93	12 23 26	11.67	12 23 28	14.10	12 26 28	11.82
17 19 23	30.15	17 19 26	29.15	17 19 28	30.20	17 23 26	20.66	17 23 28	22.13
17 26 28	20.75	19 23 26	23.28	19 23 28	24.59	19 26 28	23.35	23 26 28	11.07

classe 5 cl2=cl2=4

12 17 19	7.25	12 17 23	6.81	12 17 26	6.78	12 17 28	6.79	12 19 23	4.91
12 19 26	4.86	12 19 28	4.87	12 23 26	4.20	12 23 28	4.21	12 26 28	4.15
17 19 23	5.99	17 19 26	5.95	17 19 28	5.96	17 23 26	5.42	17 23 28	5.42
17 26 28	5.38	19 23 26	2.65	19 23 28	2.67	19 26 28	2.58	23 26 28	0.77

classe 5 cl2=cl2=6

12 17 19	12.66	12 17 23	22.93	12 17 26	11.72	12 17 28	23.76	12 19 23	21.19
12 19 26	7.76	12 19 28	22.08	12 23 26	20.64	12 23 28	29.21	12 26 28	21.55
17 19 23	22.62	17 19 26	11.09	17 19 28	23.45	17 23 26	22.11	17 23 28	30.26
17 26 28	22.96	19 23 26	20.29	19 23 28	28.96	19 26 28	21.21	23 26 28	28.56

classe 5 cl2=cl2=7

12 17 19	44.29	12 17 23	44.62	12 17 26	32.39	12 17 28	44.72	12 19 23	44.19
12 19 26	31.81	12 19 28	44.29	12 23 26	32.26	12 23 28	44.62	12 26 28	32.39
17 19 23	52.97	17 19 26	43.18	17 19 28	53.05	17 23 26	43.51	17 23 28	53.32
17 26 28	43.61	19 23 26	43.08	19 23 28	52.97	19 26 28	43.18	23 26 28	43.51

quatro Variáveis

classe 5 cl2=0

12 17 19 23	136.92	12	12 17 19 26	126.98	12 17 19 28	139.41	12 17 23 26	114.63	12 17 23 28	128.27	12 17 26 28	117.59
12 19 23 26	120.32	12	12 19 23 28	133.38	12 19 26 28	123.15	12 23 26 28	110.37	17 19 23 26	125.08	17 19 23 28	137.69
17 19 26 28	127.80	17	17 23 26 28	115.54	19 23 26 28	121.19						

classe 5 cl2=1

12 17 19 23	25.40	12	12 17 19 26	23.71	12 17 19 28	25.08	12 17 23 26	16.47	12 17 23 28	18.40	12 17 26 28	15.97
12 19 23 26	21.76	12	12 19 23 28	23.25	12 19 26 28	21.38	12 23 26 28	12.90	17 19 23 26	25.11	17 19 23 28	26.41
17 19 26 28	24.79	17	17 23 26 28	17.99	19 23 26 28	22.93						

classe 5 cl2=2

12 17 19 23	50.26	12	12 17 19 26	42.12	12 17 19 28	50.38	12 17 23 26	41.49	12 17 23 28	49.85	12 17 26 28	41.64
12 19 23 26	40.49	12	12 19 23 28	49.02	12 19 26 28	40.63	12 23 26 28	39.98	17 19 23 26	49.45	17 19 23 28	56.65
17 19 26 28	49.57	17	17 23 26 28	49.03	19 23 26 28	48.18						

classe 5 cl2=3

12 17 19 23	31.39	12	12 17 19 26	30.43	12 17 19 28	31.45	12 17 23 26	22.44	12 17 23 28	23.80	12 17 26 28	22.52
12 19 23 26	24.87	12	12 19 23 28	26.10	12 19 26 28	24.93	12 23 26 28	14.11	17 19 23 26	30.15	17 19 23 28	31.17
17 19 26 28	30.21	17	17 23 26 28	22.13	19 23 26 28	24.59						

classe 5 cl2=4

12 17 19 23	7.28	12	12 17 19 26	7.25	12 17 19 28	7.25	12 17 23 26	6.81	12 17 23 28	6.82	12 17 26 28	6.79
12 19 23 26	4.91	12	12 19 23 28	4.92	12 19 26 28	4.88	12 23 26 28	4.21	17 19 23 26	5.99	17 19 23 28	6.00
17 19 26 28	5.96	17	17 23 26 28	5.43	19 23 26 28	2.67						

classe 5 cl2=6

12 17 19 23	23.43	12	12 17 19 26	12.66	12 17 19 28	24.23	12 17 23 26	22.93	12 17 23 28	30.87	12 17 26 28	23.76
12 19 23 26	21.19	12	12 19 23 28	29.60	12 19 26 28	22.08	12 23 26 28	29.21	17 19 23 26	22.62	17 19 23 28	30.64
17 19 26 28	23.45	17	17 23 26 28	30.26	19 23 26 28	28.96						

classe 5 cl2=7

12 17 19 23	53.89	12 17 19 26	44.30	12 17 19 28	53.97	12 17 23 26	44.62	12 17 23 28	54.24	12 17 26 28	44.72
12 19 23 26	44.20	12 19 23 28	53.89	12 19 26 28	44.30	12 23 26 28	44.62	17 19 23 26	52.97	17 19 23 28	61.29
17 19 26 28	53.06	17 23 26 28	53.33	19 23 26 28	52.97						

SEPARAÇÃO ENTRE A CLASSE '6' e as CLASSES '0' '1' '2' '3' '4' '5' e '7'.
As caracteristicas testadas em função do seu factor potencial de separação são: '6' '11' 14' '19' '21' e '24'

Uma Variável

classe 6 cl2=0											
* 6	45.45	* 11	13.90	* 14	58.37	* 19	75.18	* 21	49.19	* 24	73.89
classe 6 cl2=1											
* 6	17.53	* 11	3.54	* 14	6.08	* 19	14.57	* 21	1.58	* 24	7.58
classe 6 cl2=2											
* 6	4.97	* 11	1.68	* 14	1.59	* 19	23.59	* 21	0.71	* 24	4.67
classe 6 cl2=3											
* 6	10.59	* 11	11.02	* 14	6.53	* 19	17.17	* 21	1.37	* 24	9.06
classe 6 cl2=4											
* 6	9.19	* 11	24.92	* 14	1.84	* 19	2.22	* 21	0.19	* 24	3.28
classe 6 cl2=5											
* 6	8.28	* 11	11.32	* 14	0.45	* 19	4.78	* 21	0.02	* 24	0.00
classe 6 cl2=7											
* 6	6.21	* 11	11.32	* 14	5.61	* 19	25.43	* 21	0.71	* 24	8.95

Duas Variáveis

classe 6 cl2=0											
6 ->	* 11	47.52	* 14	73.97	* 19	87.85	* 21	66.97	* 24	86.75	
11 ->	* 14	60.00	* 19	76.45	* 21	51.12	* 24	75.18			
14 ->	* 19	95.17	* 21	76.33	* 24	94.16					
19 ->	* 21	89.84	* 24	105.41							
21 ->	* 24	88.76									
classe 6 cl2=1											
6 ->	* 11	17.88	* 14	18.55	* 19	22.79	* 21	17.60	* 24	19.10	
11 ->	* 14	7.04	* 19	14.99	* 21	3.88	* 24	8.37			
14 ->	* 19	15.78	* 21	6.28	* 24	9.72					
19 ->	* 21	14.65	* 24	16.42							
21 ->	* 24	7.75									
classe 6 cl2=2											
6 ->	* 11	5.25	* 14	5.22	* 19	24.11	* 21	5.02	* 24	6.82	
11 ->	* 14	2.31	* 19	23.65	* 21	1.82	* 24	4.96			
14 ->	* 19	23.64	* 21	1.74	* 24	4.94					
19 ->	* 21	23.60	* 24	24.05							
21 ->	* 24	4.73									
classe 6 cl2=3											
6 ->	* 11	15.29	* 14	12.45	* 19	20.18	* 21	10.68	* 24	13.94	
11 ->	* 14	12.81	* 19	20.41	* 21	11.11	* 24	14.27			
14 ->	* 19	18.37	* 21	6.68	* 24	11.17					
19 ->	* 21	17.23	* 24	19.42							
21 ->	* 24	9.16									
classe 6 cl2=4											
6 ->	* 11	26.56	* 14	9.37	* 19	9.46	* 21	9.19	* 24	9.76	
11 ->	* 14	24.99	* 19	25.02	* 21	24.92	* 24	25.13			
14 ->	* 19	2.89	* 21	1.85	* 24	3.76					
19 ->	* 21	2.23	* 24	3.96							
21 ->	* 24	3.28									
classe 6 cl2=5											
6 ->	* 11	14.03	* 14	8.30	* 19	9.56	* 21	8.28	* 24	8.28	
11 ->	* 14	11.33	* 19	12.29	* 21	11.32	* 24	11.32			
14 ->	* 19	4.80	* 21	0.46	* 24	0.45					
19 ->	* 21	4.78	* 24	4.78							
21 ->	* 24	0.02									
classe 6 cl2=7											
6 ->	* 11	12.91	* 14	8.37	* 19	26.18	* 21	6.25	* 24	10.89	
11 ->	* 14	12.63	* 19	27.84	* 21	11.34	* 24	14.43			
14 ->	* 19	26.05	* 21	5.65	* 24	10.56					
19 ->	* 21	25.44	* 24	26.96							
21 ->	* 24	8.98									

três Variáveis

classe 6 cl2= cl2=0

6 11 14	73	75.27	6 11 19	88.94	6 11 21	68.40	6 11 24	87.85	6 14 19	105.47	
6 14 21	88	88.84	6 14 24	104.55	6 19 21	100.68	6 19 24	114.79	6 21 24	99.72	
11 14 19	9		96.18	11 14 21	77.58	11 14 24	95.18	11 19 21	90.91	11 19 24	106.32
11 21 24	4		89.85	14 19 21	107.13	14 19 24	120.49	14 21 24	106.23	19 21 24	116.32

classe 6 cl2= cl2=1

6 11 14	18	18.88	6 11 19	23.06	6 11 21	17.95	6 11 24	19.42	6 14 19	23.58	
6 14 21	18	18.62	6 14 24	20.04	6 19 21	22.84	6 19 24	24.02	6 21 24	19.16	
11 14 19	9		16.18	11 14 21	7.21	11 14 24	10.34	11 19 21	15.07	11 19 24	16.80
11 21 24	4		8.52	14 19 21	15.86	14 19 24	17.51	14 21 24	9.85	19 21 24	16.50

classe 6 cl2= cl2=2

6 11 14	5.	5.49	6 11 19	24.16	6 11 21	5.30	6 11 24	7.03	6 14 19	24.16	
6 14 21	5.	5.27	6 14 24	7.01	6 19 21	24.12	6 19 24	24.55	6 21 24	6.86	
11 14 19	9		23.70	11 14 21	2.42	11 14 24	5.21	11 19 21	23.66	11 19 24	24.10
11 21 24	4		5.01	14 19 21	23.65	14 19 24	24.10	14 21 24	4.99	19 21 24	24.06

classe 6 cl2= cl2=3

6 11 14	16	16.63	6 11 19	22.99	6 11 21	15.35	6 11 24	17.77	6 14 19	21.21	
6 14 21	12	12.52	6 14 24	15.39	6 19 21	20.22	6 19 24	22.12	6 21 24	14.00	
11 14 19	9		21.43	11 14 21	12.89	11 14 24	15.69	11 19 21	20.45	11 19 24	22.33
11 21 24	4		14.33	14 19 21	18.43	14 19 24	20.49	14 21 24	11.25	19 21 24	19.46

classe 6 cl2= cl2=4

6 11 14	28	26.62	6 11 19	26.65	6 11 21	26.56	6 11 24	26.76	6 14 19	9.63	
6 14 21	9.	9.38	6 14 24	9.93	6 19 21	9.46	6 19 24	10.01	6 21 24	9.76	
11 14 19	9		25.09	11 14 21	24.99	11 14 24	25.20	11 19 21	25.02	11 19 24	25.23
11 21 24	4		25.13	14 19 21	2.89	14 19 24	4.37	14 21 24	3.77	19 21 24	3.97

classe 6 cl2= cl2=5

6 11 14	14	14.03	6 11 19	14.82	6 11 21	14.03	6 11 24	14.03	6 14 19	9.58	
6 14 21	8.	8.30	6 14 24	8.30	6 19 21	9.56	6 19 24	9.56	6 21 24	8.28	
11 14 19	9		12.30	11 14 21	11.33	11 14 24	11.33	11 19 21	12.29	11 19 24	12.29
11 21 24	4		11.32	14 19 21	4.80	14 19 24	4.80	14 21 24	0.46	19 21 24	4.78

classe 6 cl2= cl2=7

6 11 14	14	14.08	6 11 19	28.52	6 11 21	12.93	6 11 24	15.71	6 14 19	26.78	
6 14 21	8.	8.40	6 14 24	12.25	6 19 21	26.19	6 19 24	27.67	6 21 24	10.92	
11 14 19	9		28.40	11 14 21	12.65	11 14 24	15.48	11 19 21	27.85	11 19 24	29.24
11 21 24	4		14.45	14 19 21	26.05	14 19 24	27.54	14 21 24	10.58	19 21 24	26.97

quatro Variáveis

classe 6 cl2=0

6 11 14 19	106.38	6 11 14 21	89.92	6 11 14 24	105.47	6 11 19 21	101.64	6 11 19 24	115.63	6 11 21 24	100.69
6 14 19 21	116.37	6 14 19 24	128.77	6 14 21 24	115.55	6 19 21 24	124.88	11 14 19 21	108.03	11 14 19 24	121.29
11 14 21 24	107.14	11 19 21 24	117.15	14 19 21 24	130.14						

classe 6 cl2=1

6 11 14 19	23.85	6 11 14 21	18.95	6 11 14 24	20.35	6 11 19 21	23.12	6 11 19 24	24.28	6 11 21 24	19.48
6 14 19 21	23.64	6 14 19 24	24.77	6 14 21 24	20.10	6 19 21 24	24.07	11 14 19 21	16.25	11 14 19 24	17.86
11 14 21 24	10.46	11 19 21 24	16.87	14 19 21 24	17.58						

classe 6 cl2=2

6 11 14 19	24.22	6 11 14 21	5.53	6 11 14 24	7.21	6 11 19 21	24.17	6 11 19 24	24.61	6 11 21 24	7.06
6 14 19 21	24.17	6 14 19 24	24.61	6 14 21 24	7.04	6 19 21 24	24.56	11 14 19 21	23.71	11 14 19 24	24.16
11 14 21 24	5.26	11 19 21 24	24.11	14 19 21 24	24.11						

classe 6 cl2=3

6 11 14 19	23.90	6 11 14 21	16.68	6 11 14 24	18.93	6 11 19 21	23.03	6 11 19 24	24.71	6 11 21 24	17.82
6 14 19 21	21.25	6 14 19 24	23.06	6 14 21 24	15.45	6 19 21 24	22.16	11 14 19 21	21.47	11 14 19 24	23.26
11 14 21 24	15.75	11 19 21 24	22.37	14 19 21 24	20.53						

classe 6 cl2=4

6 11 14 19	26.72	6 11 14 21	26.62	6 11 14 24	26.83	6 11 19 21	26.65	6 11 19 24	26.85	6 11 21 24	26.76
6 14 19 21	9.64	6 14 19 24	10.18	6 14 21 24	9.93	6 19 21 24	10.01	11 14 19 21	25.09	11 14 19 24	25.30
11 14 21 24	25.20	11 19 21 24	25.23	14 19 21 24	4.37						

classe 6 cl2=5

6 11 14 19	14.83	6 11 14 21	14.03	6 11 14 24	14.03	6 11 19 21	14.82	6 11 19 24	14.82	6 11 21 24	14.03
6 14 19 21	9.58	6 14 19 24	9.58	6 14 21 24	8.30	6 19 21 24	9.56	11 14 19 21	12.30	11 14 19 24	12.30
11 14 21 24	11.33	11 19 21 24	12.29	14 19 21 24	4.80						

classe 6 cl2=7

6 11 14 19	29.07	6 11 14 21	14.09	6 11 14 24	16.68	6 11 19 21	28.53	6 11 19 24	29.89	6 11 21 24	15.72
6 14 19 21	26.79	6 14 19 24	28.23	6 14 21 24	12.27	6 19 21 24	27.68	11 14 19 21	28.41	11 14 19 24	29.77
11 14 21 24	15.49	11 19 21 24	29.25	14 19 21 24	27.55						

SEPARAÇÃO ENTRE A CLASSE '7' e as CLASSES '0' '1' '2' '3' '4' '5' e '6'.
As características testadas em função do seu factor potencial de separação são: '1' '10' '11' '14' '16' '24' e '33'

Uma Variável

classe 7 cl2=0
* 1 30.98 * 11 2.58 * 10 24.84 * 14 52.76 * 24 64.94 * 33 33.70
classe 7 cl2=1
* 1 4.74 * 11 7.78 * 10 9.06 * 14 0.47 * 24 1.37 * 33 6.54
classe 7 cl2=2
* 1 8.36 * 11 9.64 * 10 11.42 * 14 4.02 * 24 4.28 * 33 4.28
classe 7 cl2=3
* 1 15.14 * 11 0.30 * 10 0.70 * 14 0.93 * 24 0.11 * 33 27.17
classe 7 cl2=4
* 1 11.96 * 11 36.24 * 10 37.49 * 14 7.45 * 24 12.23 * 33 5.75
classe 7 cl2=5
* 1 17.42 * 11 22.64 * 10 24.38 * 14 5.15 * 24 8.95 * 33 26.09
classe 7 cl2=6
* 1 14.51 * 11 11.32 * 10 16.80 * 14 5.61 * 24 8.95 * 33 4.59

Duas Variáveis

classe 7 cl2=0
1 -> * 11 31.09 * 10 39.71 * 14 61.18 * 24 71.95 * 33 45.77
11 -> * 10 24.97 * 14 52.82 * 24 64.99 * 33 33.79
10 -> * 14 58.31 * 24 69.53 * 33 41.86
14 -> * 24 83.67 * 33 62.60
24 -> * 33 73.16

classe 7 cl2=1
1 -> * 11 9.11 * 10 10.23 * 14 4.77 * 24 4.94 * 33 8.08
11 -> * 10 11.94 * 14 7.79 * 24 7.90 * 33 10.16
10 -> * 14 9.07 * 24 9.16 * 33 11.17
14 -> * 24 1.45 * 33 6.55
24 -> * 33 6.68

classe 7 cl2=2
1 -> * 11 12.76 * 10 14.15 * 14 9.27 * 24 9.39 * 33 9.39
11 -> * 10 14.94 * 14 10.44 * 24 10.55 * 33 10.55
10 -> * 14 12.10 * 24 12.19 * 33 12.19
14 -> * 24 5.87 * 33 5.87
24 -> * 33 6.05

classe 7 cl2=3
1 -> * 11 15.14 * 10 15.16 * 14 15.17 * 24 15.14 * 33 31.11
11 -> * 10 0.76 * 14 0.97 * 24 0.32 * 33 27.17
10 -> * 14 1.16 * 24 0.71 * 33 27.18
14 -> * 24 0.93 * 33 27.19
24 -> * 33 27.17

classe 7 cl2=4
1 -> * 11 38.16 * 10 39.35 * 14 14.09 * 24 17.10 * 33 13.27
11 -> * 10 52.14 * 14 36.99 * 24 38.24 * 33 36.69
10 -> * 14 38.22 * 24 39.43 * 33 37.92
14 -> * 24 14.32 * 33 9.41
24 -> * 33 13.51

classe 7 cl2=5
1 -> * 11 28.57 * 10 29.97 * 14 18.17 * 24 19.59 * 33 31.38
11 -> * 10 33.27 * 14 23.22 * 24 24.34 * 33 34.55
10 -> * 14 24.92 * 24 25.97 * 33 35.71
14 -> * 24 10.33 * 33 26.60
24 -> * 33 27.59

classe 7 cl2=6
1 -> * 11 18.40 * 10 22.20 * 14 15.55 * 24 17.05 * 33 15.22
11 -> * 10 20.26 * 14 12.63 * 24 14.43 * 33 12.21
10 -> * 14 17.71 * 24 19.04 * 33 17.42
14 -> * 24 10.56 * 33 7.25
24 -> * 33 10.06

três Variáveis

classe 7 cl2=0
1 11 10 39.79 1 11 14 61.24 1 11 24 72.00 1 11 33 45.85 1 10 14 66.03
1 10 24 76.12 1 10 33 52.08 1 14 24 89.22 1 14 33 69.85 1 24 33 79.45
11 10 14 58.37 11 10 24 69.58 11 10 33 41.94 11 14 24 83.71 11 14 33 62.65
11 24 33 73.21 10 14 24 87.28 10 14 33 67.35 10 24 33 77.26 14 24 33 90.20

classe 7 cl2= cl2=1

1 11 10	12	12.85	1 11 14	9.12	1 11 24	9.21	1 11 33	11.21	1 10 14	10.24	
1 10 24	10	10.32	1 10 33	12.14	1 14 24	4.96	1 14 33	8.09	1 24 33	8.19	
11 10 14			11.95	11 10 24	12.02	11 10 33	13.61	11 14 24	7.91	11 14 33	10.17
11 24 33			10.25	10 14 24	9.17	10 14 33	11.18	10 24 33	11.25	14 24 33	6.69

classe 7 cl2= cl2=2

1 11 10	17	17.12	1 11 14	13.38	1 11 24	13.46	1 11 33	13.46	1 10 14	14.71	
1 10 24	14	14.78	1 10 33	14.78	1 14 24	10.21	1 14 33	10.21	1 24 33	10.32	
11 10 14			15.47	11 10 24	15.54	11 10 33	15.54	11 14 24	11.28	11 14 33	11.29
11 24 33			11.38	10 14 24	12.84	10 14 33	12.84	10 24 33	12.92	14 24 33	7.26

classe 7 cl2= cl2=3

1 11 10	15	15.16	1 11 14	15.17	1 11 24	15.14	1 11 33	31.11	1 10 14	15.19	
1 10 24	15	15.16	1 10 33	31.11	1 14 24	15.17	1 14 33	31.12	1 24 33	31.11	
11 10 14			1.20	11 10 24	0.77	11 10 33	27.18	11 14 24	0.98	11 14 33	27.19
11 24 33			27.18	10 14 24	1.17	10 14 33	27.20	10 24 33	27.18	14 24 33	27.19

classe 7 cl2= cl2=4

1 11 10	53	53.49	1 11 14	38.88	1 11 24	40.07	1 11 33	38.59	1 10 14	40.05	
1 10 24	41	41.20	1 10 33	39.76	1 14 24	18.65	1 14 33	15.22	1 24 33	18.04	
11 10 14			52.67	11 10 24	53.55	11 10 33	52.45	11 14 24	38.96	11 14 33	37.44
11 24 33			38.67	10 14 24	40.13	10 14 33	38.65	10 24 33	39.85	14 24 33	15.43

classe 7 cl2= cl2=5

1 11 10	31	37.56	1 11 14	29.03	1 11 24	29.93	1 11 33	38.69	1 10 14	30.41	
1 10 24	31	31.28	1 10 33	39.74	1 14 24	20.25	1 14 33	31.80	1 24 33	32.63	
11 10 14			33.67	11 10 24	34.45	11 10 33	42.28	11 14 24	24.88	11 14 33	34.93
11 24 33			35.69	10 14 24	26.48	10 14 33	36.08	10 24 33	36.82	14 24 33	28.06

classe 7 cl2= cl2=6

1 11 10	24	24.92	1 11 14	19.24	1 11 24	20.46	1 11 33	18.96	1 10 14	22.90	
1 10 24	23	23.93	1 10 33	22.67	1 14 24	17.94	1 14 33	16.22	1 24 33	17.65	
11 10 14			21.02	11 10 24	22.15	11 10 33	20.77	11 14 24	15.48	11 14 33	13.44
11 24 33			15.14	10 14 24	19.84	10 14 33	18.30	10 24 33	19.58	14 24 33	11.51

quatro Variáveis

classe 7 cl2=0

1 11 10 14	66.08	1 11 10 24	76.16	1 11 10 33	52.14	1 11 14 24	89.26	1 11 14 33	69.90	1 11 24 33	79.49
1 10 14 24	92.61	1 10 14 33	74.13	1 10 24 33	83.24	1 14 24 33	95.37	11 10 14 24	87.32	11 10 14 33	67.40
11 10 24 33	77.31	11 14 24 33	90.24	10 14 24 33	93.56						

classe 7 cl2=1

1 11 10 14	12.85	1 11 10 24	12.92	1 11 10 33	14.41	1 11 14 24	9.22	1 11 14 33	11.22	1 11 24 33	11.29
1 10 14 24	10.33	1 10 14 33	12.14	1 10 24 33	12.21	1 14 24 33	8.20	11 10 14 24	12.03	11 10 14 33	13.62
11 10 24 33	13.68	11 14 24 33	10.26	10 14 24 33	11.26						

classe 7 cl2=2

1 11 10 14	17.59	1 11 10 24	17.65	1 11 10 33	17.65	1 11 14 24	14.04	1 11 14 33	14.04	1 11 24 33	14.12
1 10 14 24	15.32	1 10 14 33	15.32	1 10 24 33	15.39	1 14 24 33	11.07	11 10 14 24	16.05	11 10 14 33	16.05
11 10 24 33	16.12	11 14 24 33	12.07	10 14 24 33	13.53						

classe 7 cl2=3

1 11 10 14	15.19	1 11 10 24	15.16	1 11 10 33	31.12	1 11 14 24	15.17	1 11 14 33	31.12	1 11 24 33	31.11
1 10 14 24	15.19	1 10 14 33	31.13	1 10 24 33	31.11	1 14 24 33	31.12	11 10 14 24	1.20	11 10 14 33	27.20
11 10 24 33	27.18	11 14 24 33	27.19	10 14 24 33	27.20						

classe 7 cl2=4

1 11 10 14	54.01	1 11 10 24	54.87	1 11 10 33	53.80	1 11 14 24	40.76	1 11 14 33	39.30	1 11 24 33	40.48
1 10 14 24	41.87	1 10 14 33	40.46	1 10 24 33	41.60	1 14 24 33	19.52	11 10 14 24	54.07	11 10 14 33	52.98
11 10 24 33	53.86	11 14 24 33	39.38	10 14 24 33	40.54						

classe 7 cl2=5

1 11 10 14	37.91	1 11 10 24	38.61	1 11 10 33	45.73	1 11 14 24	30.37	1 11 14 33	39.03	1 11 24 33	39.71
1 10 14 24	31.70	1 10 14 33	40.07	1 10 24 33	40.73	1 14 24 33	33.03	11 10 14 24	34.84	11 10 14 33	42.60
11 10 24 33	43.22	11 14 24 33	36.06	10 14 24 33	37.18						

classe 7 cl2=6

1 11 10 14	25.54	1 11 10 24	26.47	1 11 10 33	25.34	1 11 14 24	21.22	1 11 14 33	19.78	1 11 24 33	20.97
1 10 14 24	24.58	1 10 14 33	23.35	1 10 24 33	24.37	1 14 24 33	18.52	11 10 14 24	22.84	11 10 14 33	21.51
11 10 24 33	22.62	11 14 24 33	16.15	10 14 24 33	20.37						

ANEXO 6

Anexo 6 Selecção de Características. Medição da distância entre *classes/defeitos* duas a duas pelo método de Jeffries Matusita(JM), para duas e três variáveis.

Separação entre classes, Medidas através da distância de Jeffries-Matusita(J-M). Os factores τ_{ij} entre classes, tomadas duas a duas, foram determinados para duas e três variáveis, referenciadas em cada um dos casos referidos neste anexo. As variáveis testadas incidiram naquelas que apresentavam potencialidades para poder separar uma ou várias classes. Foram escolhidas com base nos factores de aglomeração apresentados no capítulo 7 em 7.4.3. É observável que para diferentes combinações de variáveis, assim se encontram diferentes niveis de separabilidade entre as classes, bons, maus ou simplesmente razoáveis.

Classe 0 - backup(*B_K*)
Classe 1 - Casca(*CSC*)
Classe 2 - Sem defeito(*SDF*)
Classe 3 - Nós(Nos)
Classe 4 - Bolsa de Resina(*BRN*)
Classe 5 - Brechas de Resinagem(*BCH*)
Classe 6 - Cardido(*CRD*)
Classe 7 - Azulado da Madeira(*AZLD*)

I-SEPARABILIDADE TESTADA COM DUAS VARIÁVEIS

CARACTERÍSTICA 15 e 16(de 42) 0 a 41/objectivo testar a separabilidade de *B_K*

B_K	1							
CSC	1.26	1						
SDF	1.21	1.00	1					
NOS	1.23	0.69	0.78	1				
BRN	1.39	0.71	1.06	0.66	1			
BCH	1.41	1.11	1.33	1.26	1.14	1		
CRD	1.38	0.63	1.06	0.73	0.23	1.08	1	
AZLD	1.05	0.70	0.67	0.38	0.87	1.30	0.89	1
	B_K	*CSC*	*SDF*	*NÓS*	*BRN*	*BCH*	*CRD*	*AZLD*

Caracracterística 2 e 30 /objectivo testar a separabilidade de casca

B_K	1							
CSC	0.86	1						
SDF	1.07	1.07	1					
NOS	0.92	1.10	0.56	1				
BRN	1.10	0.65	0.98	1.06	1			
BCH	1.07	0.61	0.93	1.05	0.51	1		
CRD	1.10	1.19	0.50	0.29	1.13	1.09	1	
AZLD	1.07	1.10	0.30	0.53	0.99	0.93	0.49	1
	B_K	CSC	SDF	NÓS	BRN	BCH	CRD	AZLD

Caracracterística 20 e 22/objectivo testar a separabilidade de casca

B_K	1							
CSC	1.38	1						
SDF	1.35	0.60	1					
NOS	1.38	0.35	0.74	1				
BRN	1.41	1.07	1.07	1.19	1			
BCH	1.41	0.78	0.84	0.90	0.54	1		
CRD	1.41	0.68	0.76	0.83	0.73	0.51	1	
AZLD	1.39	0.41	0.77	0.18	1.19	0.93	0.79	1
	B_K	CSC	SDF	NÓS	BRN	BCH	CRD	AZLD

Variáviáveis 39 e 41/objectivo testar a separabilidade de Casca

B_K	1							
CSC	1.28	1						
SDF	1.23	0.97	1					
NOS	1.13	1.11	0.59	1				
BRN	1.26	0.80	0.63	0.88	1			
BCH	1.22	0.99	0.14	0.57	0.68	1		
CRD	1.16	1.24	0.82	0.65	1.11	0.78	1	
AZLD	1.25	0.77	0.54	0.83	0.56	0.56	1.05	1
	B_K	CSC	SDF	NÓS	BRN	BCH	CRD	AZLD

Caractacterística 15 e 16/objectivo testar a separabilidade de SDF

B_K	1							
CSC	1.26	1						
SDF	1.21	1.00	1					
NOS	1.23	0.69	0.78	1				
BRN	1.39	0.71	1.06	0.66	1			
BCH	1.41	1.11	1.33	1.26	1.14	1		
CRD	1.38	0.63	1.06	0.73	0.23	1.08	1	
AZLD	1.05	0.70	0.67	0.38	0.87	1.30	0.89	1
	B_K	CSC	SDF	NÓS	BRN	BCH	CRD	AZLD

Caractacterística 5 e 16/objectivo testar a separabilidade de SDF

B_K	1							
CSC	0.95	1						
SDF	0.82	0.95	1					
NOS	0.75	0.38	0.72	1				
BRN	1.21	0.66	1.31	0.86	1			
BCH	1.27	0.92	1.35	1.08	0.78	1		
CRD	1.11	0.67	0.85	0.74	1.05	1.18	1	
AZLD	0.63	0.78	0.40	0.52	1.18	1.27	0.76	1
	B_K	CSC	SDF	NÓS	BRN	BCH	CRD	AZLD

Característica 10 e 15 /objectivo testar a separabilidade de SDF

B_K	1							
CSC	1.14	1						
SDF	1.19	0.45	1					
NOS	1.07	0.56	0.78	1				
BRN	1.39	0.97	1.12	1.22	1			
BCH	1.27	0.81	1.04	0.82	0.97	1		
CRD	1.30	0.44	0.76	0.80	0.76	0.66	1	
AZLD	0.98	0.61	0.68	0.37	1.26	0.95	0.91	1
	B_K	*CSC*	*SDF*	*NÓS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica 10 e 16/objectivo testar a separabilidade de SDF

B_K	1							
CSC	1.08	1						
SDF	1.12	0.70	1					
NOS	0.97	0.63	0.82	1				
BRN	1.36	0.96	1.15	1.22	1			
BCH	1.27	0.83	1.05	1.02	1.05	1		
CRD	1.23	0.39	0.85	0.80	0.75	0.77	1	
AZLD	0.94	0.75	0.71	0.37	1.26	1.06	0.93	1
	B_K	*CSC*	*SDF*	*NÓS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica 7 e 11 /objectivo testar a separabilidade de Nos

B_K	1							
CSC	0.90	1						
SDF	0.89	0.64	1					
NOS	1.03	0.54	0.76	1				
BRN	0.93	1.01	0.90	1.19	1			
BCH	0.74	0.76	0.66	0.99	0.50	1		
CRD	0.96	0.29	0.55	0.72	1.00	0.75	1	
AZLD	1.07	0.64	0.72	0.27	1.23	1.03	0.74	1
	B_K	*CSC*	*SDF*	*NÓS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica 11 e 32/objectivo testar a separabilidade de nos

B_K	1							
CSC	0.81	1						
SDF	0.55	0.66	1					
NOS	1.03	0.82	0.99	1				
BRN	1.06	1.06	0.99	1.18	1			
BCH	0.74	0.75	0.48	1.13	0.75	1		
CRD	0.84	0.29	0.61	0.88	1.00	0.68	1	
AZLD	0.90	0.60	0.72	0.71	1.23	0.93	0.68	1
	B_K	*CSC*	*SDF*	*NÓS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica 32 e 40/objectivo testar a separabilidade de nós

B_K	1							
CSC	0.71	1						
SDF	0.53	0.49	1					
NOS	1.29	1.20	1.25	1				
BRN	0.89	0.50	0.79	1.17	1			
BCH	0.68	0.32	0.35	1.20	0.63	1		
CRD	1.03	0.72	0.85	0.97	0.82	0.71	1	
AZLD	0.69	0.36	0.33	1.18	0.67	0.22	0.67	1
	B_K	*CSC*	*SDF*	*NÓS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica 8 e 9/objectivo testar a separabilidade de BRN

B_K	1							
CSC	1.32	1						
SDF	1.34	0.95	1					
NÓS	1.35	0.38	0.84	1				
BRN	1.40	1.27	1.26	1.21	1			
BCH	1.28	0.60	0.95	0.77	1.33	1		
CRD	1.34	0.75	0.56	0.60	1.23	0.80	1	
AZLD	1.32	1.24	0.90	1.20	1.33	1.21	1.05	1
	B_K	*CSC*	*SDF*	*NÓS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica 12 e 21/ objectivo testar a separabilidade de BRN

B_K	1							
CSC	1.37	1						
SDF	1.29	0.80	1					
NÓS	1.36	0.74	0.70	1				
BRN	1.41	1.20	1.26	1.35	1			
BCH	1.41	1.08	1.21	1.23	0.74	1		
CRD	1.37	1.14	0.98	1.10	0.99	0.91	1	
AZLD	1.38	1.18	1.06	1.01	1.24	1.05	0.86	1
	B_K	*CSC*	*SDF*	*NÓS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica 12 e 26/ objectivo testar a separabilidade de bch

B_K	1							
CSC	1.39	1						
SDF	1.35	0.80	1					
NÓS	1.39	0.61	0.83	1				
BRN	1.41	0.95	1.21	1.21	1			
BCH	1.41	1.24	1.32	1.25	1.12	1		
CRD	1.38	0.79	0.79	0.74	0.98	1.22	1	
AZLD	1.38	0.57	0.71	0.44	1.27	1.31	0.89	1
	B_K	*CSC*	*SDF*	*NÓS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica 17 e 25/ objectivo testar a separabilidade de bch BCH

B_K	1							
CSC	1.35	1						
SDF	0.97	0.98	1					
NÓS	1.15	0.76	0.65	1				
BRN	1.34	0.66	1.04	0.67	1			
BCH	1.39	1.00	1.22	1.04	0.64	1		
CRD	1.26	0.96	0.71	0.60	0.75	0.98	1	
AZLD	1.10	0.82	0.67	0.57	0.91	1.16	0.89	1
	B_K	*CSC*	*SDF*	*NÓS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica 26 e 28/ objectivo testar a separabilidade de bch BCH

B_K	1							
CSC	1.38	1						
SDF	1.33	0.95	1					
NÓS	1.39	0.65	1.01	1				
BRN	1.40	0.97	1.30	0.95	1			
BCH	1.41	1.32	1.39	1.31	1.08	1		
CRD	1.37	0.97	0.80	0.83	1.11	1.34	1	
AZLD	1.35	0.77	0.60	0.93	1.20	1.36	0.75	1
	B_K	*CSC*	*SDF*	*NÓS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica 17 e 27 /objectivo testar a separabilidade de BCH

B_K	1							
CSC	1.35	1						
SDF	1.00	1.05	1					
NOS	1.17	0.83	0.63	1				
BRN	1.34	0.65	1.02	0.66	1			
BCH	1.39	0.95	1.21	1.01	0.61	1		
CRD	1.28	1.03	0.71	0.62	0.75	0.96	1	
AZLD	0.97	0.96	0.46	0.44	0.91	1.15	0.79	1
	B_K	*CSC*	*SDF*	*NÓS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica 6 e 14 /objectivo testar a separabilidade de Cardido

B_K	1							
CSC	1.19	1						
SDF	1.35	0.79	1					
NOS	1.24	0.32	0.58	1				
BRN	1.37	0.92	1.00	0.92	1			
BCH	1.34	0.70	0.69	0.62	0.57	1		
CRD	1.37	0.85	0.50	0.72	0.69	0.46	1	
AZLD	1.36	0.96	0.65	0.80	1.12	0.77	0.74	1
	B_K	*CSC*	*SDF*	*NÓS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica 11 e 33 /objectivo testar a separabilidade de Azulado

B_K	1							
CSC	1.02	1						
SDF	0.67	0.67	1					
NOS	0.88	0.81	0.83	1				
BRN	1.03	1.01	0.85	1.24	1			
BCH	0.53	1.00	0.63	0.93	0.93	1		
CRD	0.93	0.36	0.53	0.78	0.98	0.87	1	
AZLD	1.12	0.64	0.83	0.88	1.22	1.13	0.80	1
	B_K	*CSC*	*SDF*	*NÓS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica 10 e 33 /objectivo testar a separabilidade de Azulado

B_K	1							
CSC	1.09	1						
SDF	1.12	0.53	1					
NOS	0.89	0.79	0.66	1				
BRN	1.35	1.00	1.08	1.25	1			
BCH	1.12	0.94	0.79	0.84	0.96	1		
CRD	1.15	0.45	0.35	0.79	0.87	0.73	1	
AZLD	1.07	0.63	0.76	0.87	1.27	1.11	0.86	1
	B_K	*CSC*	*SDF*	*NÓS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica 22 e 33 /objectivo testar a separabilidade de Azulado

B_K	1							
CSC	1.34	1						
SDF	1.32	0.55	1					
NOS	1.35	0.80	0.59	1				
BRN	1.41	0.82	0.87	1.15	1			
BCH	1.40	0.96	0.69	0.77	0.96	1		
CRD	1.39	0.52	0.44	0.77	0.62	0.74	1	
AZLD	1.36	0.61	0.68	0.88	1.15	1.06	0.87	1
	B_K	*CSC*	*SDF*	*NÓS*	*BRN*	*BCH*	*CRD*	*AZLD*

II - SEPARABILIDADE TESTADA COM TRÊS VARIÁVEIS

Característica 10-11-33

B_K	1							
CSC	1.39	1						
SDF	1.37	1.12	1					
NOS	1.37	0.98	1.12	1				
BRN	1.41	1.23	1.21	1.32	1			
BCH	1.40	1.20	1.01	1.13	1.01	1		
CRD	1.40	0.76	1.06	0.83	1.15	1.10	1	
AZLD	1.39	0.79	1.20	0.94	1.34	1.26	0.94	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica 11-14-21

B_K	1							
CSC	1.40	1						
SDF	1.41	1.02	1					
NOS	1.40	0.65	1.02	1				
BRN	1.41	1.31	1.31	1.35	1			
BCH	1.41	1.24	1.25	1.25	0.74	1		
CRD	1.41	1.09	1.12	1.14	1.07	0.92	1	
AZLD	1.41	1.13	1.17	1.02	1.28	1.09	0.86	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica 0-15-16

B_K	1							
CSC	1.28	1						
SDF	1.39	1.28	1					
NOS	1.26	0.72	1.20	1				
BRN	1.40	0.76	1.26	0.69	1			
BCH	1.41	1.15	1.38	1.30	1.21	1		
CRD	1.40	0.89	1.12	0.89	0.59	1.22	1	
AZLD	1.36	1.20	0.76	1.14	1.19	1.36	1.07	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica 2 39 41

B_K	1							
CSC	1.30	1						
SDF	1.33	1.20	1					
NOS	1.23	1.20	0.70	1				
BRN	1.32	0.96	0.87	0.94	1			
BCH	1.28	1.09	0.56	0.65	0.74	1		
CRD	1.30	1.31	0.84	0.72	1.16	0.89	1	
AZLD	1.34	1.13	0.57	0.89	0.81	0.74	1.06	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica 5 10 16

B_K	1							
CSC	1.17	1						
SDF	1.24	1.03	1					
NOS	1.04	0.70	1.05	1				
BRN	1.38	1.05	1.33	1.27	1			
BCH	1.35	0.99	1.37	1.17	1.07	1		
CRD	1.28	0.71	0.91	0.93	1.13	1.20	1	
AZLD	1.02	0.91	0.81	0.59	1.32	1.28	0.95	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica		*35*	*38*	*40*				
B_K	1							
CSC	1.10	1						
SDF	1.05	1.05	1					
NOS	1.38	1.35	1.37	1				
BRN	1.09	0.86	0.74	1.35	1			
BCH	1.20	1.03	0.63	1.34	0.65	1		
CRD	1.28	1.15	1.17	1.10	1.07	1.03	1	
AZLD	1.15	0.91	0.78	1.32	0.49	0.52	0.90	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica		*11*	*32*	*35*				
B_K	1							
CSC	1.01	1						
SDF	0.74	0.81	1					
NOS	1.31	1.24	1.27	1				
BRN	1.15	1.12	1.10	1.34	1			
BCH	1.05	0.87	0.73	1.31	0.96	1		
CRD	1.21	0.98	0.99	1.09	1.20	0.89	1	
AZLD	1.10	0.69	0.88	1.17	1.26	0.94	0.85	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica		*8*	*21*	*24*				
B_K	1							
CSC	1.41	1						
SDF	1.41	0.74	1					
NOS	1.41	0.71	0.49	1				
BRN	1.41	1.34	1.34	1.35	1			
BCH	1.41	1.26	1.29	1.29	0.71	1		
CRD	1.41	1.24	1.23	1.24	0.90	0.44	1	
AZLD	1.41	1.25	1.23	1.22	1.34	1.01	1.09	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica		*9*	*21*	*30*				
B_K	1							
CSC	1.38	1						
SDF	1.37	0.83	1					
NOS	1.38	1.02	0.54	1				
BRN	1.41	1.27	1.33	1.34	1			
BCH	1.40	1.21	1.25	1.31	1.24	1		
CRD	1.40	1.20	1.02	0.97	1.25	1.18	1	
AZLD	1.37	1.24	1.15	1.21	1.36	1.18	1.06	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica		*14*	*20*	*32*				
B_K	1							
CSC	1.40	1						
SDF	1.39	0.81	1					
NOS	1.41	0.83	0.96	1				
BRN	1.41	1.03	1.13	1.17	1			
BCH	1.41	0.72	0.73	1.03	0.85	1		
CRD	1.41	0.76	0.61	0.92	0.82	0.49	1	
AZLD	1.40	0.94	0.65	0.94	1.21	0.92	0.74	1
	b_k	casc	sdf	nos	brn	bch	card	azld

CaCaracterística	*12*	*23*	*28*				
B_B_K	1						
CSCSC	1.36	1					
SDSDF	1.24	1.11	1				
NONOS	1.37	0.75	1.20	1			
BRBRN	1.41	1.16	1.35	1.24	1		
BCBCH	1.41	1.15	1.37	1.21	1.01	1	
CRCRD	1.30	0.91	0.85	0.95	1.28	1.30	1
AZAZLD	1.31	0.98	0.89	1.10	1.32	1.31	0.94	1
	b_k	casc	sdf	nos	bm	bch	card	azld

CaCaracterística	*17*	*26*	*29*				
B_B_K	1						
CSCSC	1.41	1					
SDSDF	1.38	1.20	1				
NONOS	1.41	0.99	1.15	1			
BRBRN	1.41	1.02	1.29	0.87	1		
BCBCH	1.41	1.08	1.33	1.11	-1.00	1	
CRCRD	1.41	1.12	1.16	0.95	0.52	0.97	1	
AZAZLD	1.39	1.00	1.02	0.69	1.11	1.23	1.00	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

CaCaracterística	*6*	*11*	*21*				
b_b_k	1						
CSCSC	1.33	1					
SDSDF	1.38	1.16	1				
NONOS	1.35	0.75	1.06	1			
BRBRN	1.40	1.31	1.35	1.34	1		
BCBCH	1.39	1.27	1.26	1.27	0.73	1	
CRCRD	1.40	1.16	1.12	1.14	1.14	0.97	1	
AZAZLD	1.40	1.16	1.14	1.09	1.30	1.12	0.80	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

CaCaracterística	*6*	*14*	*19*				
B_B_K	1						
CSCSC	1.36	1					
SDSDF	1.37	0.88	1				
NONOS	1.34	0.60	0.77	1			
BRBRN	1.41	1.04	1.15	0.97	1		
BCBCH	1.41	1.16	1.20	1.06	0.88	1	
CRCRD	1.41	1.04	0.84	0.83	0.75	0.90	1	
AZAZLD	1.39	0.98	0.68	0.88	1.18	1.18	0.91	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

CaCaracterística	*1*	*11*	*14*				
B_B_K	1						
CSCSC	1.31	1					
SDSDF	1.30	1.11	1				
NONOS	1.30	0.77	1.01	1			
BRBRN	1.38	1.24	1.14	1.28	1		
BCBCH	1.37	1.13	1.17	1.18	0.79	1	
CRCRD	1.34	0.87	0.82	0.87	1.10	1.16	1	
AZAZLD	1.39	1.25	1.26	1.16	1.27	1.17	1.26	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica	*11*	*14*	*24*				
B_K	1						
CSC	1.40	1					
SDF	1.40	1.05	1				
NOS	1.41	0.72	1.06	1			
BRN	1.41	1.22	1.16	1.28	1		
BCH	1.41	1.08	1.07	1.14	0.80	1	
CRD	1.41	0.71	0.88	0.81	1.05	0.88	1
AZLD	1.41	0.99	1.12	0.79	1.28	1.08	0.86	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica	*0*	*15*	*34*				
b_k	1						
CSC	1.22	1					
SDF	1.38	1.27	1				
NOS	1.25	0.60	1.22	1			
BRN	1.39	0.82	1.31	0.83	1		
BCH	1.29	0.94	1.30	1.16	1.03	1	
CRD	1.38	0.88	0.99	0.87	0.85	0.94	1
AZLD	1.36	1.17	0.58	1.11	1.22	1.28	0.97	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica	*3*	*15*	*16*				
B_K	1						
CSC	1.29	1					
SDF	1.25	1.05	1				
NOS	1.25	0.75	0.85	1			
BRN	1.40	1.07	1.31	0.94	1		
BCH	1.41	1.13	1.36	1.27	1.18	1	
CARD	1.39	0.82	1.07	0.84	0.86	1.19	1
AZLD	1.12	0.94	0.76	0.63	1.19	1.33	1.00	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica	*4*	*30*	*33*				
B_K	1						
CSC	1.23	1					
SDF	1.00	0.90	1				
NOS	0.83	1.15	0.78	1			
BRN	1.22	0.50	0.96	1.22	1		
BCH	1.26	0.91	1.07	1.14	1.04	1	
CRD	1.16	1.07	0.52	0.77	1.10	1.20	1
AZLD	1.15	0.79	0.62	1.00	0.89	1.14	0.77	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

CARACTERÍSTICA	*5*	*10*	*16*				
B_K	1						
CSC	1.17	1					
SDF	1.24	1.03	1				
NOS	1.04	0.70	1.05	1			
BRN	1.38	1.05	1.33	1.27	1		
BCH	1.35	0.99	1.37	1.17	1.07	1	
CRD	1.28	0.71	0.91	0.93	1.13	1.20	1
AZLD	1.02	0.91	0.81	0.59	1.32	1.28	0.95	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Cararacterística	*10*	*15*	*16*				
B_K _K	1						
CSC SC	1.28	1					
SDF DF	1.35	1.12	1				
NOS OS	1.28	0.85	1.10	1			
BRN RN	1.41	1.08	1.26	1.25	1		
BCH CH	1.41	1.20	1.35	1.28	1.26	1	
CRD RD	1.39	0.66	1.13	0.89	0.82	1.15	1
AZLZLD	1.19	0.84	0.97	0.54	1.28	1.31	0.99	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Cararacterística	*5*	*10*	*15*				
B_K _K	1						
CSC SC	1.20	1					
SDF DF	1.30	0.94	1				
NOS OS	1.12	0.63	1.04	1			
BRN RN	1.39	1.06	1.33	1.27	1		
BCH CH	1.34	0.97	1.37	1.04	0.98	1	
CRD RD	1.35	0.74	0.84	0.91	1.13	1.15	1
AZLZLD	1.09	0.82	0.80	0.61	1.32	1.25	0.92	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Cararacterística	*7*	*13*	*35*				
B_K _K	1						
CSC SC	1.40	1					
SDF DF	1.35	0.98	1				
NOS OS	1.41	1.25	1.23	1			
BRN RN	1.41	1.11	1.24	1.38	1		
BCH CH	1.41	0.99	1.05	1.34	0.98	1	
CRD RD	1.41	1.07	0.93	0.99	1.22	0.99	1
AZLZLD	1.37	0.76	0.89	1.14	1.32	1.17	0.84	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Cariaracterística	*11*	*35*	*40*				
B_K _K	1						
CSC SC	1.10	1					
SDF DF	0.83	0.79	1				
NOS OS	1.39	1.35	1.36	1			
BRN RN	1.28	1.09	0.92	1.38	1		
BCH CH	1.14	0.84	0.74	1.36	0.69	1	
CRD RD	1.34	1.09	1.14	1.16	1.15	1.04	1
AZLZLD	1.13	0.65	0.89	1.30	1.18	0.90	0.97	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Cariaracterística	*21*	*24*	*30*				
B_K _K	1						
CSC SC	1.41	1					
SDF DF	1.41	0.96	1				
NOS OS	1.41	1.11	0.53	1			
BRN RN	1.41	1.28	1.33	1.37	1		
BCH CH	1.41	1.27	1.33	1.36	-1.00	1	
CRD RD	1.41	1.32	1.23	1.23	1.14	0.99	1
AZLZLD	1.41	1.29	1.17	1.16	1.28	1.15	0.85	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica	*13*	*17*	*29*				
B_K	1						
CSC	1.36	1					
SDF	1.30	1.13	1				
NOS	1.31	0.99	0.92	1			
BRN	1.41	1.07	1.31	1.33	1		
BCH	1.41	1.02	1.30	1.22	1.05	1	
CRD	1.37	1.07	0.75	0.80	1.15	1.06	1
azld	1.19	1.03	0.74	0.52	1.32	1.26	0.85	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica	*14*	*21*	*24*				
B_K	1						
CSC	1.41	1					
SDF	1.41	0.77	1				
NOS	1.41	0.64	0.70	1			
BRN	1.41	1.28	1.30	1.32	1		
BCH	1.41	1.26	1.28	1.29	-1.00	1	
CRD	1.41	1.22	1.22	1.23	0.75	-1.00	1
AZLD	1.41	1.21	1.19	1.13	1.20	0.94	0.81	1
	B_K	*CSC*	*SDF*	*NOS*	*BRN*	*BCH*	*CRD*	*AZLD*

Característica	*11*	*14*	*22*				
B_K	1						
CASC	1.40	1					
SDF	1.40	1.17	1				
NOS	1.41	0.71	1.13	1			
BRN	1.41	1.24	1.15	1.31	1		
BCH	1.41	1.11	1.07	1.14	0.82	1	
CARD	1.41	0.87	0.83	0.93	1.04	0.89	1
AZLD	1.41	1.00	1.13	0.78	1.28	1.08	0.88	1
	B_K	*CASC*	*SDF*	*NOS*	*BRN*	*BCH*	*CARD*	*AZLD*

ANEXO 7

Anexo 7 SISTEMA MÍNIMO DE IMAGEM BIBLIOTECA IMPLEMENTADA

Não era propósito inicial do trabalho desenvolver uma biblioteca de funções de manipulação de imagem, mas acabou por se tornar necessário implementar algumas funções por necessidades de aprendizagem e de suporte da tese que se pretendia elaborar. O facto de não se estar apoiado por uma biblioteca de funções de base (sistema comercial) já desenvolvida, também contribuiu para que este trabalho tivesse de ser realizado. As limitações de tempo e dos objectivos, impõem portanto, que se olhe para esta biblioteca como uma mera ferramenta de apoio ao trabalho desenvolvido. Ela comporta as limitações derivadas do facto de estar orientada para o fim específico com que foi criada, e do facto de ter crescido aleatoriamente em função das necessidades sentidas. Neste anexo, apresentam-se as funções desenvolvidas para esse efeito e os menus de chamada utilizados. Os intuitos são mais de ilustração do que de demonstração. Ao sistema desenvolvido chamámos *Sistema Mínimo de Imagem*, para a *Localização e Reconhecimento de Defeitos em Madeira*

I - Descrição das funções contidas em cada um dos arquivo C

/* Índice de Ficheiros. Descrição das funções contidas em cada
um dos arquivo C do *Sistema Mínimo de Imagem* criado*/

TFUNCOES.C 328
TFUNC.C 329
TBINCT.C 330
PMAT.C 332
TCLEAN.C 333
TCORRL.C 333

TCREG.C 333
TC_FILE.C 334
TDIF.C 334
TE_LINHS.C 334
TOPL.C 336
TG_IMG.C 336
TM_U.C 336
TPTIMG00.C 337
TRCMDPNH.C 337
TRCPAD.C 337
TSEGMT1.C 338
TSEGMT2.C 339
TSEGMT3.C 340
TSEGQT.C 341
TSG.C 342
TTXTURAS.C.C 342
TFILTR0.C 343
TFILTR1.C 343
TFILTR2.C 344
TECT.C 344
TECT1.C 345
TGCTN.C 346
TMMT.C 347

TFUNCOEES.C

{**}

Xnumero(int it cdj,int cdi)
/*determina a a entrada de um número*/ linha 36

*void vlsegm1n1(unsigned int mhist[256],int *segm)* *linha 72*
/*determinação de *thresholds - método de Otsu* */

bnthresd(int st segm,char buff_er) *linha 122*
/* binarização io de imagens */

void tputimg1g1(int buff_er,int imgem) *linha 175*
/* faz o display de uma imagem numa àrea de écran */

void transfimmg(int buff_er,int imgem) *linha 267*
/* transfere immagem de um buffer para outro */

void trocaimgng(char img) *linha 312*
/* troca imagezens entre arquivos */

int carg_arq(q(void) *linha 336*
/* carrega um n ficheiro para o display*/
void carg_ararq1(char n) *linha 412*
/* carrega arququivo de uma imagem, numa file, para um buffer */

bntresd1(int segm1,int segm2,char buff_er,char var) *linha 450*
/* segmentação de uma imagem através de dois *threshods* */

bntresd2(int segm1,int segm2,char buff_er) *linha 432*
/* binariza a imagem a partir de dois threshold em que o primeiro tem o valor menor. Segmenta a imagem entre esses dois valores*/

void segtmad(char ch) *linha 561*
/* processo de segmentação com objectivos específicos, para a madeira*/

*float vlsegm2(unsigned int mhist[256],int *segm)* *linha 611*
/* versão de vlesgm1 */
{**********}

TFUNC.C

{**********}

/*calcula a média de uma imagem e a variancia-valor retornado*/
*int histgrm1(int *mdimg,float *smt,unsigned int math[256],int xj0,\ unsigned char yi0,int xj1,unsigned char yi1)* *linha 38*

/*calcula a média de uma imagem e a variancia-valor retornado*/
*int histgrm2(int *mdimg,float *smt,unsigned int math[256],int xj0,\ unsigned char yi0,int xj1,unsigned char yi1)* *linha 132*

*unsigned char variancia(unsigned char *f)* *linha 203*
/*calcula a variância na vizinhança dum pixel*/

thistogm(unsigned int mhist[256],int img) *linha 227*
/* Determina o histograma de uma imagem */

float SKewness(unsigned int math[256],int varcia,int media) *linha 294*
/* Estatística do 1ºgrau. Calcula o valor do skew dum histograma */

float Kurtosis(unsigned int math[256],int varcia,int media) *linha 312*
/* Estatística do 1ºgrau. Calcula o valor da kurtosis num histograma */

float Energ(unsigned int math[256]) *linha 327*
/* Estatística do 1ºgrau. Calcula o valor da energia duma imagem */

float Entrpia(unsigned int math[256]) *linha 341*
/* Estatística do 1ºgrau. Calcula o valor da energia duma imagem */

float As_m(unsigned int mcoov[32][32]) *linha 372*
/*Segundo momento angular para matrizes de covariância »ASM«medida de homogeneidade */

float Entp(unsigned int mcoov[32][32],unsigned int fact) *linha 388*
/*contraste para matrizes de coovariância, segundo diversas direcções*/
float contrast(unsigned int mcoov[32][32]) *linha 403*
/*contraste para matrizes de coovariância, segundo diversas direcções*/

float correlalaC(unsigned int mcoov[32][32]) *linha 419*
/*correlação o para matrizes de coovariância, segundo diversas direcções*/

*float difest(?(float *cont,float *sma,float *entrp,unsigned int mcoov[32][32])* *linha 476*
/*diferença e estatistica*/

void Rltorio{o() *linha 503*
/* Elabora relatatório estatístico */

void trphist{t(unsigned int math[256],char display) *linha 570*
/*display=1,1,para visualizar*/

/*diferença e estatistica*/
*float difest1{1(float *cont,float *sma,float *entrp,float *mdi,unsigned int mv[256])* *linha 620*

A_spvisd()) *linha 661*
/*aprendizagigem supervisada - lança os dados no monitor*/

A_spvisd1()() *linha 774*
/*aprendizagigem supervisada - lança os dados num ficheiro*/

void thestmmmn() *linha 887*

void thestm_n_n() *linha 985*

void dplaydedef(int bj,int ai,char def) *linha 1219*
/*faz o displalay do defeito/abrindo uma janela/ no
/ ponto indicacado ai-y;bj-x */

*unsigned chhar *dplaysaid(int bj,int ai)* *linha 1275*
/*faz o displalay de janela/abrindo uma janela/ no
/ ponto indicacado ai-y;bj-x */

void dplaydedef2(int aj,int ai,char def) *linha 1301*
/*faz o displalay do defeito no
ponto indicacado ai-y;bj-x */

void Rltrio1}1() *linha 1322*
/*faz um relaalatório estatístico da imagem. Lança os dados num ficheiro*/
{**********}

TBINCT.CC

{**********}

```
struct lista1
        {
                struct lista1 *pximo;
                int     i,j;
        };
```

```
struct lstpc1
        {
          struct lstpc1   *prox;
          struct lista1   *proximo;
          unsigned char   flag0:1,flag1:3,flag2:4;
          unsigned char   ai,bi;
          int             area,prt,num,aj,bj;
        };

/*unsigned _stklen=34321U;*/
```


*struct lista1 *nova1()* — *linha 53*
/*função que retorna uma struct*/

*struct lista1 *insere1(struct lista1 *pt)* — *linha 61*
/*insere um elemento na lista *lista1*/*

*struct lstpc1 *nov1()* — *linha 72*
/*função que retorna uma struct*/

*struct lstpc1 *isre1(struct lstpc1 *pt)* — *linha 81*
/*insere um elemento na lista *lstpc1*/*

*struct lista1 *alinhalista1(struct lista1 *ppt)* — *linha 89*

*struct lstpc1 *alinhalst1(struct lstpc1 *pt)* — *linha 105*

*unsigned int fill1(struct lista1 *llig,int num,char buff_er,* — *linha 124*
*unsigned int *ar,int *prt,int lmt[4])*

*fill3(struct lista1 *llig,int num,unsigned int *ar,int *prt,int lmt[4])* — *linha 279*
/* em relação a fill1 determina se a região é interior*/
/*ou exterior, e resolve o enchimento das regiões interiores*/

*struct lista1 *segct(char buff_er,int i,int j,int *n)* — *linha 373*

void topbin(char buff_er) — *linha 814*
/* Determina o contorno e faz o display dos valores de cada região */
/* àrea, perimetro da região, número, etc. */

void topb_n1(char buff_er) — *linha 889*
//em relação ao anterior não faz o display dos
//dados mas apresenta o contorno e o nome do
//defeito na saida

topb_n2(char buff_er) — *linha 1072*
//em relação ao anterior não faz o display dos
//dados mas apresenta o contorno e o nome do
//defeito na saida

*struct lista1 *segct1(char buff_er,int i,int j,int *n,int chncd[8])* — *linha 1273*
/*calcula a àrea, perimetro,e o número da região, */
/*para todas as superfícies segmentadas */
*struct lstpc1 *tpbin0(char buff_er)* — *linha 1733*

*struct lista1 1 *segct2(char buff_er,int i,int j,int *n)* *linha 1799*
/*Procura umm contorno numa situaçãoespecial em que a conectividade */
/* é sempre o ou quase sempre 4-contornos sempre horizontais ou verticais*/

/*calcula a àrárea, perimetro,e o número da região, */
/*para todas as as superfícies segmentadas */
/*em relação o a tpbin0, utiliza a função segmtc2 */
*struct lstpc1:1 *tpbin1(int *nmr)* *linha 2068*

/*calcula a àrárea, perimetro,e o número da região, */
/*para todas as as superfícies segmentadas */
/*em relação o a tpbin0, utiliza a função segmtc2 */
*struct lstpc1:1 *tpbin2(int *nmr)* *linha 2135*
{**********}

PMAT.C /* funções Matemáticas */

{**********}

mltmat(char i1,char j1,char i2,char j2,char mat1[10][10],
char mat2[10][10],int mat3[10][10]) *linha 34*
/*multiplicação de matrizes*/

*char pesquisa(int il,int *pivot)/*a[][] variável global*/* *linha 56*
/*inversão dele matrizes-operação interna*/

void trocalininhas(int li,int lp) *linha 86*
/*inversão dele matrizes-operação interna*/

void linhardzdz(char i,char j,float factor) *linha 99*
/*inversão dele matrizes-operação interna*/

void determmina() *linha 106*
/*inversão dele matrizes-operação interna*/

char Gauss_s_Jd(char dn,float m[20][20]) *linha 114*
/*inversão dele matrizes*/

int inversa()() *linha 165*
/* inverte mmatrizes através de Gauss_Jd(), preparando entrada */
/* interactiva ou por ficheiro, e gerindo a saida para o monitor */

int mt() *linha 344*
/* multiplicaca matrizes através de mltmat(),preparando entrada */
/* interactiva ou por ficheiro, e gerindo a saida para o monitor */

/*multiplica a matrizes de números reais*/
mltmat1(char i1,char j1,char i2,char j2,float mat1[20][20], *linha 526*
float mat2[20][20],float mat3[20][20])
{**********}

TCLEAN.C

{**********}

void suav(int fg) *linha 32*
/*Elimina através de operações lógicas, pixels isolados na imagem*/

void limpeza(char buff) *linha 81*
/*elimina pixels desconexos na imagem*/

void limpz1(char buff) *linha 184*
/*elimina pixels desconexos na imagem*/

void limpz2(char buff) *linha 200*
/*elimina pixels desconexos na imagem*/
{**********}

TCORRL.C

{**********}

int crlvar() *linha 38*

void czcrrl() *linha 123*
/*cruzamento de correlação*/

crrl() linha 207
/*cálculo da correlação entre caracteristicas*/

crrl1() linha 374
/*Cálculo da correlação entre caracteristicas.*/
/*Em relação a crrl só lança em ficheiro a correlação*/
/*para um conjunto de caracteristicas escolhidas*/
{**********}

TCREG.C

{**********}

```
struct lstreg
        {
                struct lstreg *pximo;
                unsigned char flg0:4,
                 flg1:4;
                unsigned char n0,md,ai,bi;
                int aj,bj;
                unsigned int  qtpxl;
        };

struct centrd
        {
                struct centrd *px;
                unsigned char smt_i;
                int smt_j;
        };
```

*struct lstreg *novrg()* *linha 60*
/*função que retorna uma struct*/

*struct lstreg *insrg(struct lstreg *pt)* *linha 68*

*struct lstreg *insrg2rd(struct lstreg *pt,unsigned char nivel)* *linha 79*
/*inserção ordenada*/

*struct lstreg *insrg_rd(struct lstreg *pt,unsigned char *nivel)* *linha 95*
/*inserção ordenada*/

*struct lstreg *alinhalrg(struct lstreg *ppt)* *linha 117*

*struct centrd *nvct()* *linha 133*
/*função que e retorna uma struct*/

*struct centrd *alinhact(struct centrd *ppt)* *linha 152*

c_rg(char buff_er) *linha 168*

void thcrg() *linha 482*
{**********}

TC_FILE.C

{**********}
int main (void) *linha 39*
/*função principal*/
{**********}

TDIF.C

{**********}
*difimg(char *img)* *linha 35*
/*diferença de imagem com determinação do contorno*/
{**********}

TE_LINHS.C

{**********}

```
struct lista
        {
                struct lista *pximo;
                int      i,j;
        };
struct lstpc
        {
                struct lstpc *prox;
                struct lista *proximo;
                char  flag;
                int      declive;
        } *inicio;
```

```
/*unsigned _stklen=34321U;*/
struct lista *lp; /*variável global*/

struct lista *nova()                                                    linha 51
/*função que retorna uma struct*/

struct lista *insere(struct lista *pt)                                  linha 59

struct lstpc *nov() /*função que retorna uma struct*/                   linha 70

struct lstpc *isre(struct lstpc *pt)                                    linha 78

struct lista *alinhalista(struct lista *ppt)                            linha 87

struct lstpc  *alinhalst(struct lstpc *pt)                              linha 103

struct lista *lg_linhs(struct lstpc *ptl)                               linha 122
/*ptl->flag:    0-ambos oextremos do segmento desligados;
                1-ligado o 1º extremo;
                2-ligado o segundo extremo;
                3-ambos os extremos ligados;*/

void putimg03(struct lista *ptt)                                        linha 316

struct lstpc *estrcont(unsigned char *bf)                               linha 358
/*estrutura um contorno de linhas*/

/*determina os pontos de uma linha com declive inferior a 45º,*/
/*pelo método  de Bressenham - *l_crt retorna um ponteiro para o */
/*inicio da lista e a função retorna um ponteiro para o fim da lista*/
struct lista *l_brsnam(int matriz[2][2],struct lista *l_crt,char colr)  linha 531

/*função que une através de segmentos de linha os pontos definidos
para um contorno.[uncponts]-unecontornodepontos.*lpont[] é um array
de ponteiros para lista ligada do tipo lista.*/
struct lista *uncponts(struct lista *lpont)                             linha 652

void buscpont(char modo,char curso,int matz[2][2],struct lista *crt)    linha 736

void bscpont1(char modo,char curso,int matz[2][2],struct lista *crt)    linha 825

void dt_cont(unsigned char *buf)                                        linha 906
/*determina contorno*/

Trshold(unsigned int math[256])                                         linha 1059
/*determina um threshold*/

Trshold1(unsigned int math[256],int *iii)                               linha 1163

Trshold2(unsigned int math[256],int *iii)                               linha 1231

{**********************************************************************}
```

TOPL.C

{**********}
void bitquad(i(char buff_er) *linha 34*
/* Determina a a quantidade e tipo de bitquad,e calcula
a àrea e perimetro e número de Euler com base nestes bitquad*/
{**********}

TG_IMG.C

{**********}

int gerd_img(i(int imgem) *linha 37*
/* lê uma imagem dum buffer */

int gd1_img(i(int imgem) *linha 109*
/* lê uma imagem do écran */

int gd2_img(i(int imgem) *linha 225*

{**********}

TM_U.C

{**********}
*t_bytint(nbyte , mbyte)/*converte char para unsigned int*/* *linha 28*

*menu(int *caracter)* *linha 38*
/*menu principal*/

*submenu1(int *caracter)* *linha 67*

*submenu2(int *caracter)* *linha 98*

*submenu3(int *caracter)* *linha 130*

*submenu4(int *caracter)* *linha 162*

*submenu5(int *caracter)* *linha 191*

*submenu6(int *caracter)* *linha 223*

*submenu7(int *caracter)* *linha 254*

*submenu8(int *caracter)* *linha 282*

*submenu9(int *caracter)* *linha 309*

*submnu10(int *caracter)* *linha 340*

*submenu11(int *caracter)* *linha 379*

*submnu11(int *caracter)* *linha 406*
{**********}

TPTIMG00.C

{**********}
int display() *linha 32*
/*prepara o modo gráfico no arranque do sistema*/
{**********}

TRCMDPNH.C

{**********}
void topb_n3(char buff_er,char fg) *linha 38*

RCMP() *linha 326*
/* Programa de reconhecimento de defeitos em madeira de pinho */
/* Detecta os defeitos:SDF,PPIC,C_Z,NO,BCH,BRN,BK.Retorna 0-se */
/* madeira de qualidade, ou 1 se sai pelo fim do programa */

topb_n4(char buff_er) *linha 428*
/*Apresenta o contorno e o nome do defeito de uma forma interactiva.É um*/
/*algoritmo de estudos das caracteristicas dos defeitos.Cada imagem contri*/
/*bui com um ou vários defeitos, donde se tira o valor das caracteristicas*/
/*Faz experiência com o rato */
{**********}

TRCPAD.C

{**********}

percetrn(char buff_er,int w[4]) *linha 40*

sdef(int valor[4]) *linha 277*
/*seleciona defeito a partir do deslocamento de duas rectas*/

percetrn1() *linha 377*
/*(float w1[4],float w2[4],float w3[4])*/

percetrn2() *linha 598*
/*(float w1[4],float w2[4],float w3[4])*/

lmse(char fg) *linha 871*
/*(float w1[4],float w2[4],float w3[4])*/

void supvis_d() *linha 1073*

MDPC()/(float w1[4],float w2[4],float w3[4])*/* *linha 1099*
/*faz uma classificação por distância-di(x)=x'zi-1/2zi'zi*/

Bayes1()/(float w1[4],float w2[4],float w3[4])*/* *linha 1227*
/*determina a função de decisão de bayes di=p(x/wi)p(wi),assumindo*/
/*uma didtribuição normal para as amostra*/
/*Estima a partir de protótipos a média(vector) e a covariância */
/*determina a inversa da matriz de covariância*/

Clssf() *linha 1517*
/* Programa dele classificação de uma amostra de dados */
/* Os dados são recolhidos num ficheiro RSPV.TXT */
/* Os vectores s de 4 componentes são recolhidos em PCTR.TXT */
/* São os hiperplanos de decisão d1,d2,d3 */
{**********}

TSEGMT1.C

{**********}

void thd1() *linha 45*
/*pretende-se segmentar a imagem através da determinação de thresold
por região, depois de uma segnmentação prévia*/

void thd2() *linha 237*
/*pretende-se segmentar a imagem através da determinação de thresold
por região, depois de uma segnmentação prévia*/

void eqlz() *linha 389*

void ivngtlog(char flag) *linha 425*
/*inversão de cor-imagem negativa,flag=1*/
/*e logaritimização da imagem, se flag=2*/

void thd3() *linha 460*
/*pretende-se segmentar a imagem através da determinação de thresold
por região, depois de uma segnmentação prévia*/

void rctt() *linha 563*
/*realce do contorno*/

void hypbl() *linha 614*
/*realce do contorno*/

void hypblCRR() *linha 672*
/*realce do contorno*/

void hexp() *linha 732*
/*realce do contorno*/

void thd4() *linha 792*
/*pretende-se segmentar a imagem através da determinação de thresold
por região, depois de uma segnmentação prévia*/

void thd5() *linha 904*
/*pretende-se segmentar a imagem através da determinação de thresold
por região, depois de uma segnmentação prévia*/

void thd6() *linha 967*
/*pretende-se segmentar a imagem através da determinação de thresold
por região, depois de uma segnmentação prévia*/

void esqltzcf() *linha 1049*
/*esqueletização*/

void esqltzcf2() *linha 1083*
/*Determina o contorno de àreas negras (Binarizadas),estando a imagem*/
/*original no buffer 2.*/

void thd7() *linha 1122*

void Anvrcia(int tt[7]) *linha 1325*
/*Análise de variância */

void esqltzcf1() *linha 1430*
/*retira o contorno de uma imagem-regiões negras em fundo claro*/
{**********}

TSEGMT2.C

{**********}

```
struct centrd1
        {
          truct centrd1 *px;
          unsigned char smt_i;
          int smt_j;
        };

struct centrd1 *nvct1()
        {
          struct centrd1 *x;
          x=ALCA;
          x->px=NULL;
          return(x);
        }
```

*struct centrd1 *insct1(struct centrd1 *pt)* *linha 53*

/*segmentação em imagem [binarizada]/ quatro niveis
através de crescimento por regiões*/
crib4() *linha 66*

void dsply() *linha 245*

/*faz a esqueletização.Devido ao facto das superfícies serem estreitas
a decisão de esqueletização torna-se dificil,pelo que necessitava de
ser mais elaborada.Nesto momento não faz nada, nesse aspecto.Posso re
tirar daqui um processo de definição de contorno.Descobri aqui também
um processo de gerar texturas.Vou substituir este por esqltzcf-que ape
nas define contornos.*/
void esqltzC(char buff_er) *linha 286*

void inspccreg(int th) *linha 726*
/*inspector de regiões, em função da sua uniformidade
valor elevado região com defeitos;valor baixo sem defeitos*/

void maxabsolutos() *linha 756*
/*determina os máximos numa imagem*/

void minabsolutos() linha 799
/*determina os máximos numa imagem*/

void inspcregV(int th) *linha 842*
/*inspector de e regiões, em função da sua uniformidade
variância elevada região com defeitos;valor baixo sem defeitos*/

*unsigned int inspcregV1(int *th,int *thn)* *linha 919*
/*inspector de e regiões, em função da sua uniformidade
variância elevada região com defeitos;valor baixo sem defeitos*/

*raizmadeira(int *thd,float *vc)* *linha 1008*
/*determina se há bakground numa amostra de madeira*/

*fretbk(int *thd,char ch)* *linha 1073*
/*função que retira o bakground numa amostra de madeira*/

*fretbk1(unsigned int mmh[256],int *vv0,int *thd)* *linha 1101*
/*função que retira o bakground numa amostra de madeira*/
/*decide se deve utilizar um ou dois thresolds em função*/
/*da existência ou não de background*/
{**********}

TSEGMT22.C

{**********}

/*função que determina as partições em que pode ser dividida uma imagem
utilizando projecções de segmentos da imagem.A matriz lmt[10][4] leva
os limites dala imagem e traz os limites das partições:coluna 0-ai,1-aj,
2-bi,3-bj; n lt leva a opção cinzentos '1' ou pretos '0' e traz o número
de partições-s- partc_im():partição da imagem.A função retorna -1 se não
for uniforme e retorna a densidade(nº de pixel/area)no caso contrário.*/
*partc_im(char *n,char lmt[151][4])* *linha 35*

*struct lstreg *est_img(char num)/*estrutura imagem*/* *linha 161*

/* Determina as regiões com conectividade 8 */
void conectividade() *linha 294*

void Fecho()() *linha 362*
/*fecha uma superfície dos seus espaços ocos*/

void Fecho1()() *linha 443*
/*fecha uma superfície dos seus espaços ocos*/

void Fecho2()() *linha 657*
/* fecha uma a superfície dos seus espaços ocos */
/* em relação ao fecho1(),faz uma dilatação maior 7x7 */
/* mas não aproxima a superfície dos limites da imagem */
/* imagem deve-se encontrar no buffer1 e sai no buffer2 */

void Erosao() *linha 810*
/*faz a erosão numa superfície que contenha espaços ocos*/
/*Para obter uma operação de abertura, bastava seguir esta*/
/*operação de uma de fecho */

{**********}

TSEGQT.C

{**********}

```
struct quadtree
{
        unsigned char   nivel,coordyi;
        int                             coordxj;
        struct quadtree  *esqalt,*esqbx,*drtalt,*drtbx;
};

struct quadt_e
{
        unsigned char   md,nivel,coordyi;
        int                             coordxj;
        struct quadt_e  *esqlt,*esqbxa,*drtlt,*drtbxa;
};

FILE *out,*out1;
char name[100];
unsigned char kk,k,*axbf1,*axbf2;
unsigned char A[8][2]={
                        100,160,
                        50,80,
                        25,40,
                        13,20,
                        7,10,
                        4,5,
                        2,2,
                        0,0
                        };

int t,ii,jj,axii,axjj;
unsigned int h[256],nn;
long int aux;
/*int ctrt[4];*/
```

*struct quadtree *inserearV(struct quadtree *arvq,unsigned char flag)* *linha 78*

*struct quadtree *arvHomg(struct quadtree *arvore,unsigned char qdrdo)* *linha 108*
/*(unsigned char *mbuf[200])*/

*struct quadtree *leAvqt(struct quadtree *tree,struct quadtree *ptvr)* *linha 289*

void l_aqt() *linha 358*

*struct quadt_e *arv_q(unsigned char centi,int centj,char nv,int x)* *linha 366*
/*Processo de segmentação,por divisão da imagem em zonas homogéneas acha o gradiente da imagem em função do contraste/ou da media e armazena os resultados numa estrutura em àrvore */

*struct quadt_e *arv_q2(unsigned char centi,int centj,char nv,int x)* *linha 541*
/*Processo de segmentação,por divisão da imagem em zonas homogéneas acha o gradiente da imagem em função do contraste/ou da media e armazena os resultados numa estrutura em àrvore */

*void readtreee(struct quadt_e *pt)* *linha 704*

*void dlttree(struct quadt_e *pt)* *linha 735*

void le_t(void) *linha 748*
{**********}

TSG.C

{**********}
m_etent(int img) *linha 36*

med1_im(int img) *linha 250*
{**********}

TTXTURAS.C

{**********}

mCooVr10(unsigned int matcv1_0[32][32]) *linha 34*
/*prepara matriz de covariância - *d=1 e θ =0*/

mCooVr20(unsigned int matcv1_0[32][32]) *linha 84*
/*prepara matriz de covariância - *d=2 e θ =0*/

void mCooVr90(unsigned int matcv90[32][32]) *linha 127*
/*prepara matriz de covariância - *d=1 e θ =90*/

mCooVr29(unsigned int matcv90[32][32]) *linha 168*
/*prepara matriz de covariância - *d=2 e θ =90*/

mCoVr245(unsigned int matcv45[32][32]) linha 209
/*prepara matriz de covariância - *d=2 e θ =45*/

void mCoVr145(unsigned int matcv45[32][32]) *linha 241*
/*prepara matriz de covariância - *d=1 e θ =45*/

unsigned int t mdif10(int vdd[4],unsigned int matdf1_0[256]) *linha 289*
/* prepara matriz de diferenças: d=1;θ=0º.*/

unsigned int t mdif20(int vdd[4],unsigned int matdf1_0[256]) *linha 322*
/* prepara matriz de diferenças: d=2;θ=0º.*/

unsigned int mdif190(int vdd[4],unsigned int matdf1_0[256]) *linha 341*
/* prepara matriz de diferenças: d=1;θ=90º.*/

unsigned int mdif290(int vdd[4],unsigned int matdf1_0[256]) *linha 360*
/* prepara matriz de diferenças: d=2;θ=90º.*/

void mrlgth0(int vdd[4],unsigned int matdf1_0[32][8]) *linha 379*
/* matriz de rlgth θ=0º */

void mrlgth90(int vdd[4],unsigned int matdf1_0[32][8]) *linha 413*
/* matriz de rlgth θ=90º */

void mr1lgth90(int tth,unsigned int matdf1_0[32][8]) *linha 444*
/* em relação à anterior, não permite que os pixels sejam */
/* menores do que tth-um limiar imposto pelo background- em TTXTURAS.C */

*unsigned int himdf(int vdd[4],unsigned int matdf1_0[256],int n,int *max,int *min)* *linha 478*
/*determina a matriz h[] dum defeito,delimitado por vdd[4], e */
/*na condição de no buffer 2 estar a imagem segmentada por cores.*/
/*A imagem deve-se encontrar no buffer 1.Retorna a àrea. */
/*Calcula a àrea e o histograma para a região delimitada */

unsigned int mdif10a(int vdd[4],unsigned int matdf1_0[256],int n) *linha 534*
/* Em relação a mdif10, tem apenas a diferença de controlar os pixels */
/* que pertencem à região dentro do quadrado circunscrito*/

unsigned int mdif190a(int vdd[4],unsigned int matdf1_0[256],int n) *linha 559*
/* Em relação a mdif10, tem apenas a diferença de controlar os pixels */
/* que pertencem à região dentro do quadrado circunscrito */
{**********}

TFILTR0.C

{**********}
*htgr_m2(int my,int mx,int img,unsigned char *f)* *linha 37*

unsigned char var_cia1(char buff_er,int i,int j) *linha 80*
/*calcula a variância*/

vc1(int i,int j) *linha 122*

sbel_ft3(int img) *linha 149*

lp_ft2(int img) *linha 214*
{**********}

TFILTR1.C

{**********}

*htgr_m1(int my,int mx,int img,unsigned char *f)* *linha 37*

*unsigned char var_cia(char buff_er,int i,int j,int *med)* *linha 80*

sbel_ft(int imag) *linha 124*

lp_ft(int img)) *linha 262*

rbert_ft(int imag) *linha 372*

media_im(int img) *linha 515*

svizmd(int imag) *linha 589*

sbel_ft1(int imag) *linha 638*

{**********}

TFILTR2.CC

{**********}

sbel_ft2(int imag) *linha 37*

fpalt(int img)) *linha 177*

{**********}

TECT.C /*escolha de caracteristicas*/

{**********}

/* função que determina os valores sre,lre,rln,nng e rp */
/* pelo métoddo GLRLM */
*float rlgthmd1(float *sre, float *lre,float *rln,float *nng,unsigned int m_h[32][8])* *linha 44*

/* função que determina os valores sre,lre,rln,nng e rp */
/* pelo métoddo GLRLM.faz o mesmo que rlgthmd1, de modo diferente */
float rlgthmd1(float *sre, float *lre,float *rln,float *nng,unsigned int m_h[32][8]) linha 91

/* determina um conjunto de caracteriaticas de uma imagem */
/* separando as caracteristicas de imagens com defeito das imagens */
/* sem defeito.Ficheiros "cdf.txt" e "sdf.txt" respectivamente */
/* A função selct1() retira da imagem 48 caracteristicas, com as quais */
/* se pretende e investigar a possibilidade de separar uma imagem sem */
/* defeitos dumma com defeitos-variáveis descritas no anexo1*/
selct1() *linha 136*

*spitc() /*cálculo do factor de separação interclasses[F]*/* *linha 480*
/*para a características globais SDF/CDF*/
/*factor *F* , ficheiro FCTF.TXT/médias MMSDFCDF.TXT*/

*difc() /*cálculo da distância Euclidiana de separação interclasses SDF/CDF*/* *linha 665*
/*através da escolha de seis características em quarenta e oito */

MMJfMt() *linha 910*
/*Determina o o máximo e o minimo por caracteristica, para o método JfMt*/

*McvJfMt(chaar flag,char md[4],char *ptt)* *linha 1087*
/*Determina a a média e a covariância de uma amostra padrões pertencentes */
/*a uma classe, lançando os dados (em variáveis globais) num ficheiro */

void DMdia(float mmdia[8][42]) *linha 1212*
/*Determina a média das 42 caracteristicas para todos os tipos */
/*de defeitos e apresenta-os através de mmdia[][], normalizafdas*/
/*a partir de ficheiro de protótipos*/

float Detn(char flag,float xx[20][20]) *linha 1310*
/* determina o determinante de uma matriz 2x2 ou 3x3(flag=2 ou flag=3)*/

FctJfMt(char flag) *linha 1326*
/*(float w1[4],float w2[4],float w3[4])*/
/*determina o factor 'a' de Jeffries-Matusita na determinação da */
/*separação entre classes em função de 2 ou três caracteristicas */
/*Retorna o valor do factor 'a' e leva como argumentos o nº de */
/*caracteristicas.Utiliza funções de multiplicação e inversão de matrizes*/

FBayes(char flag,float Vdtnt[8],float mxmn[2][3],char mdm[4]) *linha 1495*
/*(float w1[4],float w2[4],float w3[4])*/
/*determina os factores lnlcovrl e 1/cov para determinar a expressão*/
/*de decisão de bayes em função das variáveis que forem escolhidas. */
/*Utiliza funções de multiplicação e inversão de matrizes*/

*selctd2(char nm,struct lstpc1 *axlp)* *linha 1621*
/* determina um conjunto de caracteriaticas para regiões */
/* préviamente segmentadas numa imagem, enviando para ficheiros */
/* diferentes em função do tipo de defeito em presença */
/* Caracteristicas retiradas de cada defeito por selct2(....) descritas no anexo 2 */

/*determina um conjunto de caracteriaticas para regiões préviamente */
/*segmentadas numa imagem, enviando para um ficheiro de dados */
/* fd_d1(...) utiliza a matriz de dados mtd[42], seu argumento*/
*selctd3(struct lstpc1 *axlp,float mtd[42])* *linha 1900*

ordct(char flg) *linha 2204*
/*ordenação de caracteristicas através das suas propriedades*/
/*de aglomeração(usando a variância das caracteristicas)-por defeito*/
/*flg=1 ordena as caracteristicas em relação às outras caracteristicas*/
/*flg=2 ordena as caracteristicas em relação a ela mesma para cada um dos*/
/*defeitos_a variância foi normalizada e para os factores foi encontrada a média */

Cvmsic() *linha 2417*
/*cálculo de vectores de médias para separação interclasses*/
/* máximos e minimos.Ficheiro TIPDND.TXT com valores dos vectores*/
/*de separação */

void difc2() *linha 2609*
{**********}

TECT1.C /*medição e selecção de características*/

{**********}
medctVdft() *linha 40*
/*Esta função mede várias caracteristicas, em função dos defeitos que */
/*apresenta a imagem.Pergunta se existe background ou não. */

void fddnd() *linha 155*
/*função de deecisão se a imagem contém defeitos ou não contém defeitos*/
/*Ensaio em mmadeira de pinho para determinação */
/*da defeituosisidade da madeira ou não*/

void fd_d() *linha 281*
/*Função de ddecisão.Decide se a imagem contém defeitos e de que tipo*/

void fd_d1() *linha 447*
/*Função de ddecisão.Decide se a imagem contém defeitos e de que tipo*/

void fd_d2() *linha 609*
/*Função de ddecisão.Decide se a imagem contém defeitos e de que tipo */
/*Através de uum processo hierárquico, com controlo em cada um dos nós*/
/*da àrvore de decisão.Possibilidade de alterar o nº e o tipo de */
/*variáveis em cada um dos nós da àrvore. */

void fd_d3() *linha 876*
/*Função de ddecisão.Decide se a imagem contém defeitos e de que tipo */
/*Através de uum processo hierárquico/composto,*/
/* com controlo em cada um dos nós da àrvore ramificada de decisão */
/*Possibilidadde de alterar o nº e o tipo de */
/*variáveis em cada um dos nós da àrvore. */

void fd_d4() *linha 1212*
/*Função de ddecisão.Decide se a imagem contém defeitos e de que tipo */
/*Através de uum processo hierárquico/composto,*/
/* com controlo em cada um dos nós da àrvore ramificada de decisão */
/*Possibilidadde de alterar o nº e o tipo de */
/*variáveis em cada um dos nós da àrvore. */
{**********}

TGCTN.C /* Determinação do contorno */
{**********}
*struct lista1 *geractrn1(int inci,int incj,int *n)* *linha 31*
/* Gera um ccontorno numa imagem segmentada que não contenha ruído */
/* O contornoo é procurado na fronteira, entre o background e a região */
/* segmentadda */

*unsigned int t fill2(struct lista1 *llig,int num,char buff_er,* *linha 238*
*unsigned int * *ar,int *prt,int lmt[4])*

void tpct1(chaar buff_er) *linha 391*
{**********}

TMMT.C /* Gerador de momentos */
{**********}

*float mmcentt1(struct lista1 *llig ,int area,float *m1,* *linha 33*
float *m2,floatt *m3,float *m4,float *m5,float *m6)
/* Gera o momnentos centrais de uma região segmentada*/

*float mmcent(struct lista1 *llig ,int prt,float *m1,* *linha 90*
*float *m2,float *m3,float *m4,float *m5,float *m6)*
/* Gera o momentos centrais de uma região segmentada*/

*float mmcent2(int vdd[4],int n,float *m1,float *m2,float *m3,* *linha 144*
*float *m4,float *m5,float *m6)*
/* Gera o momentos centrais de uma região segmentada
a imagem depois de tratada deve encontrar-se no buf2 */
{**********}

TECT2.C /*localização e reconhecimento de defeitos em madeira*/
{**********}

float selctd4(struct lstpc1 *axlp,float *vr1,float *vr2, float *vr3)
/* retira características de imagem:Inclinação eixo maior,
% àrea não ocupada, eixo maior/eixo menor, P*P/A.
Se a região for estreita e vertical axlp->flag2=1, else '0' */

char grcdf(char opc,char ctd,char cnm)
/* referencia os defeitos contidos nas imagens */
/* processo de medição estatistica automática */
/* '1'-disk 3/vtl; '2'-disk 2/vtl1;'3'-disk 1/vtl2;*/
/* '4'-disk 4/vtl3;'5'-disk 5/vtl4;*/

fd_d5(char opcc)
/*Processo de detecção automático de defeitos em madeira*/
/*Função de decisão.Decide se a imagem contém defeitos e de que tipo */
/*Através de um processo hierárquico/composto,*/
/* com controlo em cada um dos nós da àrvore ramificada de decisão */
/*Possibilidade de alterar o nº e o tipo de */
/*variáveis em cada um dos nós da àrvore. */

{**********}

Índice Remissivo das funções

A_spvisd() ..... linha 661 .....TFUNC.C
A_spvisd1() ..... linha 774 .....TFUNC.C
alinhact(struct centrd *ppt) ..... linha 152 .....TCREG.C
alinhalista(struct lista *ppt)..... linha 87 .....TE_LINHS.C
alinhalista1(struct lista1 *ppt)..... linha 89 .....TBINCT.C
alinhalrg(struct lstreg *ppt)..... linha 117 .....TCREG.C
alinhalst(struct lstpc *pt)..... linha 103 .....TE_LINHS.C
alinhalst1(struct lstpc1 *pt)..... linha 105 .....TBINCT.C
Anvrcia(int tt[7]) ..... linha 1325 ....TSEGMT1.C
arv_q(unsigned char centi,int centj,char nv,int x)..... linha 366 .....TSEGQT.C

arv_q2(unsigned char centi,int centj,char nv,int x)..... linha 541 .....TSEGQT.C
arvHomog(struct quadtree *arvore,unsigned char qdrdo) ..... linha 108 .....TSEGQT.C
As_m(unnsigned int mcoov[32][32])..... linha 372 .....TFUNC.C
Bayes1()..... linha 1227 ....TRCPAD.C
bitquad(char buff_er)..... linha 34 .....TOPL.C
bnthresdd(int segm,char buff_er)..... linha 122 .....TFUNCOES.C
bntresd11(int segm1,int segm2,char buff_er,char var)..... linha 450 .....TFUNCOES.C
bntresd22(int segm1,int segm2,char buff_er)..... linha 432 .....TFUNCOES.C
bscpont11(char modo,char curso,int matz[2][2],struct lista *crt)linha 825..TE_LINHS.C
buscponnt(char modo,char curso,int matz[2][2],struct lista *crt)linha 736..TE_LINHS.C
c_rg(chaar buff_er)..... linha 168 .....TCREG.C
carg_arqq()..... linha 336 .....TFUNCOES.C
carg_arqq1(char n)..... linha 412 .....TFUNCOES.C
Clssf()..... linha 1517 ....TRCPAD.C
conectivvidade()..... linha 294 .....TSEGMT3.C
contrast(unsigned int mcoov[32][32])..... linha 403 .....TFUNC.C
correlaCC(unsigned int mcoov[32][32])..... linha 419 .....TFUNC.C
crib4()..... linha 66 .....TSEGMT2.C
crlvar()..... linha 38 .....TCORRL.C
Cvmsic()..... linha 2417 ....TECT.C
czcrrl()..... linha 123 .....TCORRL.C
determinna()..... linha 106 .....PMAT.C
difc()..... linha 665 .....TECT.C
difc2()..... linha 2609 ....TECT.C
difest(fldoat *cont,float *sma,float *entrp,\..... linha 476 .....TFUNC.C
float *mmdi,unsigned int mv[256])
difest1(float *cont,float *sma,float *entrp,\..... linha 620 .....TFUNC.C
int mcooov[32][32])
difimg(char img)..... linha 35 .....TDIF.C
display()..... linha 32 .....TPTIMG00.C
dlttree(struct quadt_e *pt) ..... linha 735 .....TSEGQT.C
DMdia(float mmdia[8][42]) ..... linha 1212 ....TECT.C
dplaydeef(int bj,int ai,char def) ..... linha 1219 ....TFUNC.C
dplaydeef2(int aj,int ai,char def) ..... linha 1301 ....TFUNC.C
dplaysaiid(int bj,int ai) ..... linha 1275 ....TFUNC.C
dsply()..... linha 245 .....TSEGMT2.C
dt_cont(t(unsigned char *buf)..... linha 906 .....TE_LINHS.C
Energ(uunsigned int math[256]) ..... linha 327 .....TFUNC.C
Entp(unnsigned int mcoov[32][32],unsigned int fact)..... linha 388 .....TFUNC.C
Entrpia(unsigned int math[256])..... linha 341 .....TFUNC.C
eqlz()..... linha 389 .....TSEGMT1.C
Erosao()..... linha 810 .....TSEGMT3.C
esqltzC(char buff_er)..... linha 286 .....TSEGMT2.C
esqltzcf(f()..... linha 1049 ....TSEGMT1.C
esqltzcf1()..... linha 1430 ....TSEGMT1.C
esqltzcf2()..... linha 1083 ....TSEGMT1.C
est_imgg(char num)..... linha 161 .....TSEGMT3.C
estrcontt(unsigned char *bf)..... linha 358 .....TE_LINHS.C
FBayes(s(char flag,float Vdtnt[8],float mxmn[2][3],\
char mddm[4])..... linha 1495 ....TECT.C
FctJfMtt(char flag)..... linha 1326 ....TECT.C
fd_d()..... linha 281 .....TECT1.C
fd_d1()..... linha 447 .....TECT1.C
fd_d2()..... linha 609 .....TECT1.C

fd_d3() ..... linha 876 .....TECT1.C
fd_d4() ..... linha 1212 ....TECT1.C
fd_d5(char opcc) ..... linha 336 .....TECT2.C
fddnd() ..... linha 155 .....TECT1.C
Fecho() ..... linha 362 .....TSEGMT3.C
Fecho1() ..... linha 443 .....TSEGMT3.C
Fecho2() ..... linha 657 .....TSEGMT3.C
fill1(struct lista1 *llig,int num,char buff_er,\ ..... linha 124 .....TBINCT.C
unsigned int *ar,int *prt,int lmt[4])
fill2(struct lista1 *llig,int num,char buff_er,\ ..... linha 238 .....TGCTN.C
unsigned int *ar,int *prt,int lmt[4])
fill3(struct lista1 *llig,int num,unsigned int *ar,int *prt,\
int lmt[4]) ..... linha 279 .....TBINCT.C
float Detn(char flag,float xx[20][20]) ..... linha 1310 ....TECT.C
fpalt(int img) ..... linha 177 .....TFILTR2.C
fretbk(int *thd,char ch) ..... linha 1073 ....TSEGMT2.C
fretbk1(unsigned int mmh[256],int *vv0,int *thd) ..... linha 1101 ....TSEGMT2.C
Gauss_Jd(char dn,float m[20][20]) ..... linha 114 .....PMAT.C
gd1_img(int imgem) ..... linha 109 .....TG_IMG.C
gd2_img(int imgem) ..... linha 225 .....TG_IMG.C
geractrn1(int inci,int incj,int *n) ..... linha 31 .....TGCTN.C
gerd_img(int imgem) ..... linha 31 .....TG_IMG.C
grcdf(char opc, char ctd, char cnm) ..... linha 200 .....TECT2.C
hexp() ..... linha 732 .....TSEGMT1.C
himdf(int vdd[4],unsigned int matdf1_0[256],int n,int *max,\
int *min) ..... linha 478 .....TTXTURAS.C
histgrm1(int *mdimg,float *smt,\ ..... linha 38 .....TFUNC.C
histgrm2(int *mdimg,float *smt,unsigned int math[256],int xj0,\
hmath[256],int xj0,unsigned char yi0,int xj1,unsigned char yi1)
htgr_m1(int my,int mx,int img,unsigned char *f) ..... linha 37 .....TFILTR1.C
htgr_m2(int my,int mx,int img,unsigned char *f) ..... linha 37 .....TFILTR0.C
hunsigned char yi0,int xj1,unsigned char yi1) ..... linha 132 .....TFUNC.C
hypbl() ..... linha 614 .....TSEGMT1.C
hypblCR() ..... linha 672 .....TSEGMT1.C
insere(struct lista *pt) ..... linha 59 .....TE_LINHS.C
insere1(struct lista1 *pt) ..... linha 61 .....TBINCT.C
inserarV(struct quadtree *arvq,unsigned char flag) ..... linha 78 .....TSEGQT.C
inspccreg(int th) ..... linha 726 .....TSEGMT2.C
inspcregV(int th) ..... linha 842 .....TSEGMT2.C
inspcregV1(int *th,int *thn) ..... linha 919 .....TSEGMT2.C
insrg(struct lstreg *pt) ..... linha 68 .....TCREG.C
insrg_rd(struct lstreg *pt,unsigned char *nivel) ..... linha 95 .....TCREG.C
insrg2rd(struct lstreg *pt,unsigned char nivel) ..... linha 79 .....TCREG.C
inversa() ..... linha 165 .....PMAT.C
isre(struct lstpc *pt) ..... linha 78 .....TE_LINHS.C
isre1(struct lstpc1 *pt) ..... linha 81 .....TBINCT.C
ivngtlog(char flag) ..... linha 425 .....TSEGMT1.C
Kurtosis(unsigned int math[256],int varcia,int media) ..... linha 312 .....TFUNC.C
l_aqt() ..... linha 358 .....TSEGQT.C
l_brsnam(int matriz[2][2],struct lista *l_crt,char colr) ..... linha 531 .....TE_LINHS.C
le_t() ..... linha 748 .....TSEGQT.C
leAvqt(struct quadtree *tree,struct quadtree *ptvr) ..... linha 289 .....TSEGQT.C
lg_linhs(struct lstpc *ptl) ..... linha 122 .....TE_LINHS.C
limpeza(char buff) ..... linha 81 .....TCLEAN.C

limpz1(char buff) ..... linha 184 .....TCLEAN.C
limpz2(char buff) ..... linha 200 .....TCLEAN.C
linhardzz(char i,char j,float factor)..... linha 99 .....PMAT.C
lmse(chaar fg)..... linha 871 .....TRCPAD.C
lp_ft(int t img) ..... linha 262 .....TFILTR1.C
lp_ft2(innt img) ..... linha 214 .....TFILTR0.C
m_etentt(int img) ..... linha 36 .....TSG.C
main ()..... linha 39 .....TC_FILE.C
maxabsolutos()..... linha 756 .....TSEGMT2.C
mCooVr10(unsigned int matcv1_0[32][32])..... linha 34 .....TTXTURAS.C
mCooVr20(unsigned int matcv1_0[32][32])..... linha 84 .....TTXTURAS.C
mCooVr29(unsigned int matcv90[32][32])..... linha 168 .....TTXTURAS.C
mCooVr90(unsigned int matcv90[32][32])..... linha 127 .....TTXTURAS.C
mCoVr145(unsigned int matcv45[32][32])..... linha 241 .....TTXTURAS.C
MevJfMMt(char flag,char md[4],char *ptt) ..... linha 1087 ....TECT.C
mdif10(int vdd[4],unsigned int matdf1_0[256]) ..... linha 289 .....TTXTURAS.C
mdif10aa(int vdd[4],unsigned int matdf1_0[256],int n)..... linha 534 .....TTXTURAS.C
mdif1900(int vdd[4],unsigned int matdf1_0[256])..... linha 341 .....TTXTURAS.C
mdif1900a(int vdd[4],unsigned int matdf1_0[256],int n)..... linha 559 .....TTXTURAS.C
mdif20(int vdd[4],unsigned int matdf1_0[256]) ..... linha 322 .....TTXTURAS.C
mdif2900(int vdd[4],unsigned int matdf1_0[256])..... linha 360 .....TTXTURAS.C
MDPC()..... linha 1099 ....TRCPAD.C
med1_inm(int img) ..... linha 250 .....TSG.C
medctVdft()..... linha 40 .....TECT1.C
media_idm(int img) ..... linha 515 .....TFILTR1.C
menu(innt *caracter)..... linha 38 .....TM_U.C
mltmat(char i1,char j1,char i2,char j2,char mat1[10][10],\
char maat2[10][10],int mat3[10][10]) ..... linha 34 .....PMAT.C
mltmat11(char i1,char j1,char i2,char j2,float mat1[20][20],\....linha 526 ..PMAT.C
float maat2[20][20],float mat3[20][20])
mmcentt(struct lista1 *llig ,int prt,float *m1,\ ..... linha 90 .....TMMT.C
float *m4,float *m5,float *m6)
mmcentt1(struct lista1 *llig ,int area,float *m1,\ ..... linha 33 .....TMMT.C
float *m2,float *m3,float *m4,float *m5,float *m6)
mmcentt2(int vdd[4],int n,float *m1,float *m2,float *m3,\.. linha 144 .....TMMT.C
float *m2,float *m3,float *m4,float *m5,float *m6)
MMJfMMt()..... linha 910 .....TECT.C
mr1lgth90(int tth,unsigned int matdf1_0[32][8])..... linha 444 .....TTXTURAS.C
mrlgth0()(int vdd[4],unsigned int matdf1_0[32][8])..... linha 379 .....TTXTURAS.C
mrlgth900(int vdd[4],unsigned int matdf1_0[32][8])..... linha 413 .....TTXTURAS.C
mt()..... linha 344 .....PMAT.C
nov() ..... linha 70 .....TE_LINHS.C
nov1() ..... linha 72 .....TBINCT.C
nova() ..... linha 51 .....TE_LINHS.C
nova1() ..... linha 53 .....TBINCT.C
novrg()..... linha 60 .....TCREG.C
nvct() ..... linha 133 .....TCREG.C
ordct(chhar flg)..... linha 2204 ....TECT.C
partc_inm(char *n,char lmt[151][4])..... linha 35 .....TSEGMT3.C
percetrnn(char buff_er,int w[4]) ..... linha 40 .....TRCPAD.C
percetrnn1()..... linha 377 .....TRCPAD.C
percetrnn2()..... linha 598 .....TRCPAD.C
pesquissa(int il,int *pivot)/*a[][] variável global*/..... linha 56 .....PMAT.C
putimg003(struct lista *ptt)..... linha 316 .....TE_LINHS.C

raizmadeira(int *thd,float *vc) ..... linha 1008 ....TSEGMT2.C
rbert_ft(int img) ..... linha 372 .....TFILTR1.C
RCMP() ..... linha 326 .....TRCMDPNH.C
rctt() ..... linha 563 .....TSEGMT1.C
readtree(struct quadt_e *pt) ..... linha 704 .....TSEGQT.C
rlgthmd(float *sre, float *lre,float *rln,\ ..... linha 44 .....TECT.C
rlfloat *nng,unsigned int m_h[32][8])
Rltorio() ..... linha 503 .....TFUNC.C
Rltrio1() ..... linha 1322 ....TFUNC.C
sbel_ft(int img) ..... linha 124 .....TFILTR1.C
sbel_ft1(int img) ..... linha 638 .....TFILTR1.C
sbel_ft2(int img) ..... linha 37 .....TFILTR2.C
sbel_ft3(int img) ..... linha 149 .....TFILTR0.C
sdef(int valor[4]) ..... linha 277 .....TRCPAD.C
segct(char buff_er,int i,int j,int *n) ..... linha 373 .....TBINCT.C
segct1(char buff_er,int i,int j,int *n,int chncd[8]) ..... linha 1273 ....TBINCT.C
segct2(char buff_er,int i,int j,int *n) ..... linha 1799 ....TBINCT.C
segtmad(char ch) ..... linha 561 .....TFUNCOES.C
selct1() ..... linha 136 .....TECT.C
selctd2(char nm,struct lstpc1 *axlp) ..... linha 1621 ....TECT.C
selctd3(struct lstpc1 *axlp,float mtd[42]) ..... linha 1900 ....TECT.C
selctd4(struct lstpc1 *axlp,float *vr1,float *vr2, float *vr3)...linha 42 .....TECT2.C
SKewness(unsigned int math[256],int varcia,int media)..... linha 294 .....TFUNC.C
spitc() ..... linha 480 .....TECT.C
struct centrd1 *insct1(struct centrd1 *pt) ..... linha 53 .....TSEGMT2.C
suav(int fg) ..... linha 32 .....TCLEAN.C
submenu1(int *caracter) ..... linha 67 .....TM_U.C
submenu11(int *caracter) ..... linha 379 .....TM_U.C
submenu2(int *caracter) ..... linha 98 .....TM_U.C
submenu3(int *caracter) ..... linha 130 .....TM_U.C
submenu4(int *caracter) ..... linha 162 .....TM_U.C
submenu5(int *caracter) ..... linha 191 .....TM_U.C
submenu6(int *caracter) ..... linha 223 .....TM_U.C
submenu7(int *caracter) ..... linha 254 .....TM_U.C
submenu8(int *caracter) ..... linha 282 .....TM_U.C
submenu9(int *caracter) ..... linha 309 .....TM_U.C
submnu10(int *caracter) ..... linha 340 .....TM_U.C
submnu11(int *caracter) ..... linha 406 .....TM_U.C
supvis_d() ..... linha 1073 ....TRCPAD.C
svizmd(int img) ..... linha 589 .....TFILTR1.C
t_bytint(nbyte , mbyte) ..... linha 28 .....TM_U.C
thcrg() ..... linha 482 .....TCREG.C
thd1() ..... linha 45 .....TSEGMT1.C
thd2() ..... linha 237 .....TSEGMT1.C
thd3() ..... linha 460 .....TSEGMT1.C
thd4() ..... linha 792 .....TSEGMT1.C
thd5() ..... linha 904 .....TSEGMT1.C
thd6() ..... linha 967 .....TSEGMT1.C
thd7() ..... linha 1122 ....TSEGMT1.C
thestm_n() ..... linha 985 .....TFUNC.C
thestmmn() ..... linha 887 .....TFUNC.C
thistogm(unsigned int mhist[256],int img) ..... linha 227 .....TFUNC.C
topb_n1(char buff_er) ..... linha 889 .....TBINCT.C
topb_n2(char buff_er) ..... linha 1072 ....TBINCT.C

topb_n3(char buff_er,char fg) ..... linha 38 ..... TRCMDPNH.C
topb_n4(char buff_er) ..... linha 428 ..... TRCMDPNH.C
topbin(char buff_er) ..... linha 814 ..... TBINCT.C
tpbin0(char buff_er) ..... linha 1733 ..... TBINCT.C
tpbin1(int *nmr) ..... linha 2068 ..... TBINCT.C
tpbin2(int *nmr) ..... linha 2135 ..... TBINCT.C
tpct1(char buff_er) ..... linha 391 ..... TGCTN.C
tputimg11(int buff_er,int imgem) ..... linha 175 ..... TFUNCOES.C
transfimgg(int buff_er,int imgem) ..... linha 267 ..... TFUNCOES.C
trocaimg(char img) ..... linha 312 ..... TFUNCOES.C
trocalinhaas(int li,int lp) ..... linha 86 ..... PMAT.C
trphist(unsigned int math[256],char display) ..... linha 570 ..... TFUNC.C
Trshold(unsigned int math[256]) ..... linha 1059 ..... TE_LINHS.C
Trshold11(unsigned int math[256],int *iii) ..... linha 1163 ..... TE_LINHS.C
Trshold22(unsigned int math[256],int *iii) ..... linha 1231 ..... TE_LINHS.C
uncpontss(struct lista *lpont) ..... linha 652 ..... TE_LINHS.C
unsigned char var_cia(char buff_er,int i,int j,int *med) ..... linha 80 ..... TFILTR1.C
var_cia1(char buff_er,int i,int j) ..... linha 80 ..... TFILTR0.C
varianciaa(unsigned char *f) ..... linha 203 ..... TFUNC.C
vc1(int i,int j) ..... linha 122 ..... TFILTR0.C
vlsegm1(unsigned int mhist[256],int *segm) ..... linha 72 ..... TFUNCOES.C
vlsegm2(unsigned int mhist[256],int *segm) ..... linha 611 ..... TFUNCOES.C
Xnumeroo(int cdj,int cdi) ..... linha 36 ..... TFUNCOES.C

II - Menus utilizados pelo Sistema Mínimo de Imagem

MENU PRINCIPAL

* Escolha a Operação			
* <case 49>>Manipulação de Imagem	submenu4	< 1 >	Carrega, Grava e Transfere Imagens
* <case 50>>Histograma.....	submenu8	< 2 >	Representa Histogramas de Imagem
* <case 51>>Segmentação/Contornos..	submenu2	< 3 >	Det. Contornos e Oprç. em Img. Binárias
* <case 52>>Binarização	submenu3	< 4 >	Operações de Segmentação
* <case 53>>Operç/Matrizes	submenu5	< 5 >	Multiplicação,Inv, Determnt., Corrlção
* <case 54>>Ensaios	submenu7	< 6 >	Testes Localizaç/identfç em Madeira
* <case 55>>filtros ..	submenu1	< 7 >	Operações de Filtragem
* <case 56>>l/sg-imagem		< 8 >	Operação Diferença de Imagem
* <case 57>>Relatório ..	submenu11	< 9 >	Relatórios Estatisticos da Imagem
* <case 65>>Aprendizagem ..	submenu6	< A >	Aquisição Características. Aprendizagem
* <case 66>>Enhancement ..	submenu9	< B >	Operações Transformação Histograma
* <case 67>>Ensaios/Pinho ..	submnu10	< C >	Operações Teste e Medição Características
* <case 68>>Pinho BV. ..	submnu11	< D >	*Localiz/Reconhecimento Defeitos Pinho*
* <DEFAULT:I>QUIT	<Esc>		Saída

submenu1(int *caracter) Filtragens

```
*****************************
*       Escolha a Operação
* FILTRO SOBEL ..... < 1 >   Operação de filtragem 'Sobel'
* LAPLACIANO ..... < 2 >     Operação de filtragem 'Laplaciano'
* Roberts ..... < 3 >        Operação de filtragem 'Roberts'
* media ..... < 4 >          Operação de filtragem 'Filtragem de Média'
* OP.F.SOBEL1 ..... < 5 >    Operação de filtragem 'Sobel-processo1'
* OP.F.SOBEL2 ..... < 6 >    Operação de filtragem 'Sobel-processo2'
* filtro lp/VC ..... < 7 >   Operação de filtragem 'laplaciano/threshold'
* filtro mediana ..... < 8 > Operação de filtragem 'Filtro de Mediana'
* SAIDA ..... < ESC >        Saida
*****************************
```

submenu2(int *caracter) Contornos e Segmentação

```
*****************************
*       Escolha a Operação
* CONTGIZ ..... < 1 >        Detecção de contorno marcado a giz
* CRESCREG ..... < 2 >       Segmentação por crescimento de regiões
* CONTGIZ3 ..... < 3 >       Detecção de contorno marcado a giz/3
* ARV/MEDIA ..... < 4 >      Árvore quaternária/segmentação
* MEDIA ..... < 5 >          Operação de homogenização
* MEDIA/varc ..... < 6 >     Operação de homogenização/med+varcia
* cont/circl ..... < 7 >     Determinação de orlas/op de fecho, abertura, etc...
* SAIDA ..... <ESC>          Saida
*****************************
```

submenu3(int *caracter) Segmentação

```
*****************************
*       Escolha a Operação
* thresold1 ..... < 1 >      Operação de segmentação/pcesso 1
* thresold2 ..... < 2 >      Operação de segmentação/pcesso 2
* Topologias ..... < 3 >     Medição de parâmetros Topológicos na imagem
* Topolog1 ..... < 4 >       Medição de parâmetros Topológicos na imagem/1
* thresold3 ..... < 5 >      Operação de segmentação/pcesso 3
* thresold4 ..... < 6 >      Operação de segmentação/pcesso 4
* thresold5 ..... < 7 >      Operação de segmentação/pcesso 5
* SAIDA ..... <ESC>          Saida
*****************************
```

Submenu4(int *caracter) Operações Manipulação de Imagem

```
*********************************
*       Escolha a Operação
* GRAVA IMAGEM ..... < 1 >   Grava imagem a partir de um buffer
* LOAD IMAGEM ..... < 2 >    Carrega imagem de um ficheiro
* TRANSFERE IMAGEM ..... < 3 > Transfere imagens entre buffers
* GRAVA/INTG IMAGEM ..... < 4 > Grava imagem a partir da memória video
* SAIDA ..... <ESC>          Saida
*********************************
```

Submenu5(int *caracter) Operações matemáticas

```
*****************************
*       Escolha a Operação
* INVERS/MATZ ..... < 1 >    Inversão de matrizes
* MULTPLC/MTZ ..... < 2 >    Operações com matrizes
* CORRELAÇÃO ..... < 3 >     Correlação entre variáveis
* SAIDA ..... <ESC>          Saida
*****************************
```

submenu6(int *caracter) Aprendizagem

```
*******************************
*        Escolha a Operação
* RCPa'adrões-Teste ..... < 1 > Função de decisão/teste sobre protótipo
* Apdizizgem/Svzda-Teste ... < 2 > Recolha de variáveis padrão nas imagens
* Apdzzg/Svzada_IMG ..... < 3 > Recolha de variáveis padrão nas imagens/interactiva
* Classssificação1 ..... < 4 > Aprendizagem do classificador por processo Perceptron
* Classssificação2[Bayes] .... < 5 > Aprendizagem do classificador por processo Bayes
* Classssificação3 ..... < 6 > Ensaio de Classificação/utilização dos hiperplanos obtidos
* SAIIIDA ..... <ESC> Saida
*******************************
```

submenu7(int *caracter) Ensaios de Classificação

```
*******************************
*        Escolha a Operação
* ENSδAIO 1 ..... < 1 > Ensaio de classificação em madeira, não pinho/ad hoc
* ENSδAIO 2 ..... < 2 > Ensaio de Segmentação por 'edges'
* ENSδAIO em Pinho < 3 > Ensaio de classificação em pinho/ad hoc
* SAIIIDA ..... <ESC> Saida
*******************************
```

submenu8(int *caracter) Histogramas

```
*******************************
*        Escolha a Operação
* HIS3TGR 1 ..... < 1 > Histograma da imagem com display
* HIS3TGR 2 ..... < 2 > Histograma da imagem com display/de texturas
* SAIIIDA ..... <ESC> Saida
*******************************
```

submenu9(int *caracter) Pré Tratamentos

```
*******************************
*        Escolha a Operação
* Filtroro P. Alto ..... < 1 > Filtro passa alto
* Realdce 1 ..... < 2 > Operação de Realce/1
* Realdce 2 ..... < 3 > Operação de Realce/2
* Realdce 3 ..... < 4 > Operação de Realce/3
* Equalalização ..... < 5 > Equalização histograma
* Inv_)_logrtz ..... < 6 > Transformação histograma/logaritmização e inversão
* SAIIIDA ..... <ESC> Saida
*******************************
```

submnu10(int *caracter) Medição de Características e Caracterização da imagem

```
*******************************
*        Escolha a Operação
* ENSδAIO PINHO 1 ..... < 1 > Ensaios com vários threshjold
* ENSδAIO PINHO 2 ..... < 2 > Ensaios com vários threshjold e projecções
* ENSδAIO PINHO 3 ..... < 3 > Àrvore quaternária
* ENSδAIO PINHO 4 ..... < 4 > Operações no histograma e medição das àreas de segmentação
* ENSδAIO PINHO 5 ..... < 5 > Operação de segmentação
* ENSδAIO PINHO 6 ..... < 6 > Medição de parâmetros na imagem
* Cla. v. valor caract. globais ... < 7 > Medição de características globais
* Cla. f. fact. sep. caract. gb. < 8 > Escolha de características globais/cálculo do factor F-spitc( )
                                    Determinação das médias das características globais
* Correrelação entre crcterist. < 9 > Correlação entre características
* Distàtância entre classes < A > Distância Euclidiana entre classes globais SDF/CDF- Anexo 3
* Caracteristicas por Defeito < B > Medição de características em defeitos individuais
* Factctor ordenação/Defeito < C > Medição dos factores de ordenação por classe/defeito
* Factctor ordenação/JfMt < D > Medição fact. ordenação classe/defeito-Jeffries-Mattusita
* Factctor ordn. crt/Dist. < E > Medição fact. ordenação classe/defeito - por distância
* SAIIIDA < ESC> Saida
*******************************
```

submenu11(int *caracter) Medição Texturas

```
*****************************
*       Escolha a Operação
* Relt 1  ....          < 1 >   Medição de valores de textura em imagens / Relatório/vers.1
* Relt 2/fich.          < 2 >   Medição de valores de textura em imagens / Relatório/vers.2
* SAIDA  ....  <ESC>    *
*****************************
```

submnu11(int *caracter) Classificação

```
************************************
*       Escolha a Operação
* TESTE B_K  .....          < 1 >   Teste da presença do backup na imagem
* TESTE V/intClss .....     < 2 >   Ensaios e opções de localização e identificação de defeitos
* TESTE SDF/CDF  ....       < 3 >   Separação de imagens com defeito das imagens sem defeito
* TESTE DEF/DEF  ...        < 4 >   Identificação de defeito a defeito nas imagens
* T.v2  DEF/DEF  .....      < 5 >   Identificação de defeito a defeito nas imagens/versão 2
* T.hier def  .....         < 6 >   Identificação de defeito a defeito /processo hierárquico
* T.hier glbal  .....       < 7 >   Idtificação global e defeito a defeito /com processo
                                    hierárquico
* SAIDA  .....              <ESC>   Saida
************************************
```

ANEXO 8

Anexo 8 Exemplo de Implementação de *processo de localização e reconhecimento de defeitos em madeira*, através do Programa fd_d6(...), desenvolvido em *C*.

O programa apresentado neste anexo, fornece a listagem de um conjunto de procedimentos que permitem executar as operações necessárias à implementação das decisões descritas no esquema da fig. 7.19.

Não são apresentados no anexo a listagem de todas as funções referenciadas pelo programa, dado que o tornaria muito extensa, e sairia fora dos objectivos pretendidos, já que não se pretende dissertar sobre os procedimentos informáticos, mas fundamentalmente sobre os processos algoritmicos de processamento de imagem utilizados. O objectivo do Anexo 8, consiste desse modo, exclusivamente em adequar o que foi exposto no Capítulo VII, a um exemplo de aplicação que possa ilustrar a sua implementação de uma forma concreta. Não serão fornecidos detalhes sobre o significado de algumas variáveis e processos, mas apenas comentários pontuais quando considerados justificados.

```
/*>>*************************************************************************
* Projecto: Parte3          Modulo:capitulo 07          13/09/93            *
*.....                                                                      *
* Nome do arquivo:TECT2.C, Tese: escolha de características, e classificação - 2ºArquivo *
* Funções no arquivo: selctd4(...), grcdf(...), fd_d5(...), selctd2(...), fd_d6(...).  *
* Ultima alteração: 12/1/94                                                  *
* Autor:Rui Bebiano                                                          *
*.....                                                                      *
* Finalidade: localização e identificação de defeitos em madeira            *
```

```
                                        .....
*  Parametros de entrada: Arquivos de imagens de madeira com ou sem defeitos   *
*                         como, ... P21c04.imd, ... . O parâmetro de entrada opcc,  *
*                         permite seleccionar entre um processo automático e   *
*                         um processo interactivo.                             *
*                                                                              *
*  Saida: Display de imagens com ou sem defeito no écran, devidamente classificadas *
*  Retorno: zero - sem significado                                             *
*                                                                              *
*  Funções accionadas pela função fd_d6( ) [não pertencentes à biblioteca do C]:  *
*                         só são referenciadas as funções visiveis do programa *
*                         DMdia( ), gd2_img(...), carg_arq(...), carg_arq1(...),  *
*                         bntresd(...), bnthresd( ), tputimg1(...), conectividade(...)  *
*                         Fecho1(...), limpz1(...), esqltzcf2(...), tpbin2(...), selct2(...),  *
*                         selctd4(...), dplaydef2(...), apglst1(...), selctd3(...), grcdf(...).  *
*  Observaçães:                                                                *
*                   Detecção do background, segmentasção da imagem, extracção  *
*                   de contornos(armazena em listas os contornos das           *
*                   regiões.Extracção de características de textura e de contorno -  *
*                   de acordo com fig. 7.19 e classificação global e hierárquica,  *
*                   com display do nome dos defeitos.                          *
*                                                                              *
********************************************************************************/


#include <fcntl.h>
#include <io.h>
#include <stdio.h>
#include <allloc.h>
#include <process.h>
#include <graphics.h>
#include <conio.h>
#include <dos.h>
#include <math.h>
#include <string.h>
#include "propria1.h"


#define ESC  27
#define ENTTER  13
#define BACKSPC 8

/*variáveis globais*/

/*unsigned _stklen = 64000U;*/
extern int inndc[4];/*variáveis de forma das regiões não normalizadas*/
extern char iM_g;
extern char string[50];
extern unsiggned char *buf1[200],*buf2[200],*buf3[200];
extern int maxx,maxy;

/*****************************************/
```

```
..... /* funções não representadas */ .....

fd_d6(char opcc)
/*Processo de detecção automático de defeitos em madeira*/
/*Função de decisão.Decide se a imagem contém defeitos e de que tipo */
/* O primeiro processo de decisão é global, procurando enquadrar a */
/* imagem na classe/defeito SDF ou CDF. No segundo caso a análise */
/* continua através de exploração defeito a defeito.*/
/*Através de um processo hierárquico/composto,*/
/* com controlo em cada um dos nós da àrvore ramificada de decisão */
/*Possibilidade de alterar o nº e o tipo de */
/*variáveis em cada um dos nós da àrvore. */
{
        int gd2_img(int);
        FILE *out,*out1;
        struct lista1 *axpt;
        struct lstpc1  *axlp,*axlpc;
        unsigned char *axbf0,*axbf1,*axbf2,*axbf3;
        char ch,tt=1,ctdor=0,ctnm,cttrl,flag,ndef,vtst,ms[90];

        char nome[16][13]={
                                "a:p21c04.imd",
                                "a:p21c03.imd",
                                "a:p11c01.imd",
                                "a:p11c00.imd",
                                "a:p11f01.imd",
                                "a:p12f00.imd",
                                "a:p12f01.imd",
                                "a:p12c00.imd",
                                "a:p12c01.imd",
                                "a:p13f00.imd",
                                "a:p13f01.imd",
                                "a:p13c00.imd",
                                "a:p13c01.imd",
                                "a:p13c02.imd",
                                "a:p14f00.imd",
                                "a:p14f01.imd"
                           };

        ..... /* arquivos não representados*/ .....

        char op[4][6]={  5,0,3,4,5,2,
                         4,3,4,5,2,0,
                         3,4,5,2,0,0,
                         2,5,2,0,0,0
                       };

        char vrv[4][6]={
                         4,0,15,21,29,0,
                         4,11,32,35,38,0,
                         4,20,24,30,34,0,
                         4,10,15,19,23,0
                        },
                         seq[8]={0,3,4,5,2,0,0,0};
```

```
                char stg[8][6]={
                                "b_k",
                                "csc",
                                "sdf",
                                "nos",
                                "brn",
                                "bch",
                                "crd",
                                "azld"
                                };

/* vrv[][] cotém o indíce das variáveis a utilizar em cada nó */
/* seq[[] contém sequência de defeito de saida em cada nó     */

int i,j,nj,k,th1,th2,nrg,v0,v1,v2;
int clssg[8][8],clssdd[8][8],vl[4],mkuad[6];
/* variáveis globais:rlgth90/44;varcia/4;nºobj/15;nºobj>200/12?.*/
/* 1ªclasse-SDF;2ªclasse-CDF */

int crtmd[2][4]={
                55,35,78,78,
                37,60,36,30
                },
vlct[4];;

unsigned int mh[256],mrlgth[32][8],n0,n1;
float mdos[48],md1s[42],res1[8],vmdia[8][42],mgm[6][4],mdia,varc,vxl,mid;

setfillstyle(SOLID_FILL,14);
bar(0,240,640,480);
for(i=0;i<8;i++)
for(j=0;j<8;j++)
{clssdd[i][j]=0;clssg[i][j]=0;}

DMdiaa(vmdia);

 if((outt=fopen("clss1.txt","wt"))==NULL)
          {
          fprintf(stderr,"cannot open output file.\n");
          exit(1);
          }

           if((out1=fopen("crct1.txt","wt"))==NULL)
          {
          fprintf(stderr,"cannot open output file.\n");
          exit(1);
          }
```

```
while(tt)
{
        switch(opcc)
        {
                case 0:
                if(ctdor)
                {
                        setfillstyle(SOLID_FILL,14);
                        bar(0,240,640,480);
                        outtextxy(20,260,"Quer gravar a imagem 1?[S][N]");
                        ch=getch();
                        if(ch=='S' || ch=='s')
                        gd2_img(1);
                        bar(0,240,640,480);
                        outtextxy(20,260,"Quer continuar a análise?[S][N]");
                        ctdor=getch();ch=ch-48;
                        sprintf(ms,"%c",ctdor);
                        outtextxy(20,330,ms);
                        getch();
                        if(ctdor=='S' || ctdor=='s')
                        carg_arq();
                        else {fclose(out);fclose(out1);return(0);}
                }
                else ctdor=1;
                break;

                case 1: if(ctdor==16) goto SAID;
                        stpcpy(string,nome[ctdor]);
                        carg_arq1(1);
                        ctdor++;
                        break;
                case 2: if(ctdor==22) goto SAID;
                        stpcpy(string,nom1[ctdor]);
                        carg_arq1(1);
                        ctdor++;
                        break;
                case 3: if(ctdor==20) goto SAID;
                        stpcpy(string,nom2[ctdor]);
                        carg_arq1(1);
                        ctdor++;
                        break;
                case 4: if(ctdor==22) goto SAID;
                        stpcpy(string,nom3[ctdor]);
                        carg_arq1(1);
                        ctdor++;
                        break;
                case 5: if(ctdor==16) goto SAID;
                        stpcpy(string,nom4[ctdor]);
                        carg_arq1(1);
                        ctdor++;
                        break;
                }
        th1=fretbk1(mh,&k,&th2);
        /***/
```

```
        if(th2!=0)
        flag=bmtresd2(th2,th1,1);
        else flag=bnthresd(th1,1);
        /*setfillstyle(SOLID_FILL,15);
        bar(0,220,639,440);
        sprintf(ms,"th2 %d th1 %d",th2,th1);
        outtextxy(0,245,ms);
        getch();*/
        tputimg1(flag,1);
        vl[0]=0;vl[1]=0;vl[2]=199;vl[3]=319;
        mkuad[0]=0;mkuad[1]=0;mkuad[2]=0;
        mkuad[3]=0;mkuad[4]=0;mkuad[5]=0;
/************determinação dos bitquad***********/
        for(i=0;i<maxx-1;i++)
        {
         axbf0=buf1[i];
         axbf1=buf1[i]+1;
         axbf2=buf1[i+1];
         axbf3=buf1[i+1]+1;
         for(j=0;j<maxy-1;j++)
         {
          k=((imt)(*axbf0)+(*axbf1)+(*axbf2)+(*axbf3))/255;
          switch(k)
          {
                case 0:mkuad[0]++;
                       break;
                case 1:mkuad[1]++;
                        break;
                case 2:if((*axbf0==*axbf3)||(*axbf1==*axbf2)) mkuad[5]++;
                       else mkuad[2]++;
                       break;
                case 3:mkuad[3]++;
                        break;
                case 4:mkuad[4]++;
                        break;
          }
          axbf0++;axbf1++;axbf2++;axbf3++;
         }
}
                n0=mkuad[0]+mkuad[1]+mkuad[2]+mkuad[3]+mkuad[4]+mkuad[5];
                mdos[5]=((float)(mkuad[0])/n0)*100;
                mdos[6]=((float)(mkuad[1])/n0)*100;
                mdos[7]=((float)(mkuad[2])/n0)*100;
                mdos[8]=((float)(mkuad[3])/n0)*100;
                mdos[9]=((float)(mkuad[4])/n0)*100;
                mdos[10]=((float)(mkuad[5])/n0)*100;
/***********fim do cálculo de bitquad*********/
        conectividade();
        tputimg1(2,1);
        Fecho1();
        limpz11(2);
        tputimg1(2,1);
```

```
        esqltzcf2();
        tputimg1(1,1);
        axlp=axlpc=tpbin2(&nrg);
        tputimg1(1,2);
        carg_arq1(1);
/** determinação dos limites do background - prenchimento de vl[] **/
        if(th2!=0)/* 22 */
        {
                vl[1]=vl[3]=v1=v2=v0=0;
                axbf0=buf1[100];
                for(j=0;j<320;j++)
                {
                        if(*axbf0<=th2) {v1++;v2++;}
                        else v2=0;
                        if(v2==2 && v0==0) {vl[1]=j-2;v0=1;}
                        axbf0++;
                }
                vl[3]=vl[1]+v1;
                if(v1>6)
                {
                j=vl[1]+v1/2;
                v1=v2=v0=0;
                for(i=0;i<200;i++)
                {
                 axbf0=buf1[i]+j;
                 if(*axbf0<=th2) {v1++;v2++;}
                 else v2=0;
                 if(v2==2 && v0==0) {vl[0]=i-2;v0=1;}
                }
         vl[2]=vl[0]+v1;
         /*if(vl[3]>(320-vl[3]))*/
         if(vl[3]>300)
         {
                vl[3]=vl[1];
                vl[1]=0;
         }
          else
                {
                  vl[1]=vl[3];
                  vl[3]=319;
                }
        vl[0]=0;vl[2]=199;
        }
        else /* 11 */
                {
                 j=160;
                 v1=v2=v0=0;
                 for(i=0;i<200;i++)
                 {
                        axbf0=buf1[i]+j;
                        if(*axbf0<=th2) {v1++;v2++;}
                        else v2=0;
                        if(v2==2 && v0==0) {vl[0]=i-2;v0=1;}
                 }
```

```
                vl[2]=vl[0]+v1;
                i=vl[0]+v1/2;
                v1=v2=v0=0;
                axbf0=buf1[i];
                for(j=0;j<320;j++)
                {
                        if(*axbf0<=th2) {v1++;v2++;}
                        else v2=0;
                        if(v2==2 && v0==0) {vl[1]=j-2;v0=1;}
                        axbf0++;
                }
                vl[3]=vl[1]+v1;
                /*if(vl[2]>(200-vl[2]))*/
                if(vl[2]>180)
                {
                        vl[2]=vl[0];
                        vl[0]=0;
                }
        else
                {
                        vl[0]=vl[2];
                        vl[2]=199;
                }
                vl[1]=0;vl[3]=319;
                }/*fim do else 11*/
        }/*fim do if 22 */
/**fim escolha wl[]***/
        /*bar(0,240,640,480);
        sprintf(ms,"vl %d %d %d %d mh %d %d",\
        vl[0],vl[1],vl[2],vl[3],mh[100],mh[140]);
        outtextxy(20,360,ms);
        getch();;*/
        /*determina o valor de algumas características globais*/
        selct2(th2,vl,mh,mdos);
        /*outtextxy(20,460,"fim");
        getch();;*/
        vlct[1]=(int)(mdos[4]/10);
        /*variância-variável global*/
        vlct[0]=(int)(mdos[44]*12);/*run length-lre90, vrvel global*/
        v1=v2=0;
        for(i=0;i<4;i++)
        {
                v1=v1+crtmd[0][i]*crtmd[0][i];
                v2=v2+crtmd[1][i]*crtmd[1][i];
        }
 fprintf(out,"%ss ",string);
 n0=0;n1=0;cttrl=0;
 while(axlpc!=NULL)
 {
        n0=n0+++;
        if(axlpoc->area>200)
        {
          n1++;;
        mdia=sselctd4(axlpc,&varc,&vxl,&mid);
```

```
            if(axlpc->flag2==1) cttrl++;
            fprintf(out,"%2.2f;%2.2f;%2.2f;%2.2f:: ",mdia,varc,vxl,mid);
            if(n1<=6)
            {
                    mgm[n1-1][0]=mdia;
                    mgm[n1-1][1]=varc;
                    mgm[n1-1][2]=vxl;
                    mgm[n1-1][3]=mid;
            }
          }
  axlpc=axlpc->prox;
}
vlct[2]=n0*3;/*número de objectos*/
vlct[3]=n1*12;
mdos[15]=n0;mdos[14]=n1;
mdos[11]=n0-n1;mdos[13]=0;mdos[12]=0;
mdia=(float)(cttrl)*100/n1;
fprintf(out,"\n");
fprintf(out1,"\n");
fprintf(out1,"%s \n",string);
for(i=0;i<48;i++)
{
          if(i%8==0 && i!=0) fprintf(out1,"\n");
          fprintf(out1,"%3.2f; ",mdos[i]);
}
fflush(stdin);
v0=k=0;
for(i=0;i<4;i++)
{
           v0=v0+crtmd[0][i]*vlct[i];
           k=k+crtmd[1][i]*vlct[i];
}
v1=v0-v1/2;
v2=k-v2/2;
fprintf(out,"%d %d %d %d; v1 %d v2 %d ct %2.2f\n",\
vlct[0],vlct[1],vlct[2],vlct[3],v1,v2,mdia);
/*ct % de regiões rectângulares*/
if(v1>v2 || (vlct[0]<22 && vlct[2]>60))/* tratamento global */
{
          if(vlct[1]<=25)
          {
          /*vtst=0;vtst=0;variável indicação de saida*/
          dplaydef2(152,94,2);
          bar(0,240,640,480);
          ch=grcdf(opcc,ctdor,9);
          fprintf(out,"no-00 %s \t",stg[ch]);
          fprintf(out,"clss:%s \n",stg[2]);
          clssg[ch][2]++;
          apglst1(axlp);
          if(th2==0)
          continue;
          else goto BCKG;
          }
```

```
        if(mdia>=60)
        {
        dplaydef2(152,94,2);
        bar(0,240,640,480);
        ch=grcdf(opcc,ctdor,9);
        fprintf(out,"no-01 %s \t",stg[ch]);
        fprintf(out,"clss: %s \n",stg[2]);
        clssg[ch][2]++;
        apglst1(axlp);
        if(th2==0)
        continue;
        else goto BCKG;
        }
for(j=0;j<1;j++)
{
/*int crtmd[2][4]={
                        15,45,30,30,
                        90,85,60,70
                };
/+ Anel global '02'-caract 1,2,4,29 +/
vlct[0]=(int)(mdios[1]*4);/+ medidas centrais sdf-15;cdf-90+/
vlct[1]=(int)(mdios[2]*5);/+ medidas centrais sdf-45;cdf-85+/
vlct[2]=(int)(mdios[4]/10);/+ medidas centrais sdf-30;cdf-60+/
vlct[3]=(int)(mdios[29]/5);/+ medidas centrais sdf-30;cdf-70*/

int crtmd[2][4]={
                        35,60,40,45,
                        60,75,20,55
                };
/* Anel global '02'-caract 4,16,29,31 */
vlct[0]=(int)(mdios[4]/10);/* varcia,medidas centrais sdf-35;crd-60*/
/*vlct[1]=0;/+ valor anterior 35...55 */
vlct[1]=(int)(mdios[45]*100);/*skewness,medidas centrais sdf-35;crd-55*/
vlct[2]=(int)(mdios[44]*12);/*lre rte,medidas centrais sdf-40;crd-20*/
vlct[3]=(int)(mdios[31]*10);/*entp,medidas centrais sdf-45;crd-55*/

v0=k=0;
for(i=0;i<4;i++)
{
        v0=v0+crtmd[0][i]*vlct[i];
        k=k+crtmd[1][i]*vlct[i];
}
v1=v2=0;
for(i=0;i<4;i++)
{
        v1=v1+crtmd[0][i]*crtmd[0][i];
        v2=v2+crtmd[1][i]*crtmd[1][i];
}
        v1=v0-v1/2;
        v2=k-v2/2;
        }/*fim do for*/
```

```
        if(v2>v1)
        {
         dplaydef2(152,94,2);
         bar(0,240,640,480);
         ch=grcdf(opcc,ctdor,9);
         fprintf(out,"%d; %d; %d; %d ::v1 v2: %d %d \n",\
         vlct[0],vlct[1],vlct[2],vlct[3],v1,v2);
         fprintf(out,"no-02 %s \t",stg[ch]);
         fprintf(out,"clss: %s \n",stg[6]);
         clssg[ch][6]++;
         apglst1(axlp);
         if(th2==0)
         continue;
         else goto BCKG;
         }
         /*anel '03'*/
for(j=0;j<1;j++)
{
        /*int crtmd[2][4]={
                          55,30,0,0,
                          30,60,0,0
                         };
/+ Anel global '03'-caract 4,44 +/
vlct[0]=mdos[44]*12;/+ medidas centrais sdf-55;cdf-30+/
vlct[1]=mdos[4]/10;/+ medidas centrais sdf-30;cdf-60*/
int crtmd[2][4]={
                 20,20,16,40,
                 50,30,24,50
                 };
/* Anel global '03'-caract 0,2,29,31*/
vlct[0]=(int)(mdos[0]);/*v.minimo,medidas centrais azd-20;sdf-50*/
vlct[1]=(int)(mdos[4]/10);/*variância,medidas centrais azd-30;sdf-40*/
vlct[2]=(int)(mdos[28]*2);/*media/diif,medidas centrais azd-45;sdf-70*/
vlct[3]=(int)(mdos[31]*10);/*entrpia,medidas centrais azd-40;sdf-50*/

v0=k=0;
for(i=0;i<4;i++)
{
        v0=v0+crtmd[0][i]*vlct[i];
        k=k+crtmd[1][i]*vlct[i];
}
v1=v2=0;
for(i=0;i<4;i++)
{
        v1=v1+crtmd[0][i]*crtmd[0][i];
        v2=v2+crtmd[1][i]*crtmd[1][i];
}
       v1=v0-v1/2;
       v2=k-v2/2;
       }/*fim do for*/
       if(v1>v2)
       {
        dplaydef2(152,94,7);
        bar(0,240,640,480);
```

```
        ch=grcdf(opcc,ctdor,9);
         fprintf(out,"%d; %d; %d; %d ::v1 v2: %d %d \n",\
        vlct[0],vlct[1],vlct[2],vlct[3],v1,v2);
        fprintf(out,"no-03 %s \t",stg[ch]);
        fprintf(out,"clss: %s \n",stg[7]);
        clssg[ch][7]++;
        apglst1(axlp);
        if(th2==0)
        continue;
        else goto BCKG;
        }
        else
        {
        dplaydef2(152,94,6);
        bar(0,240,640,480);
        ch=grcdf(opcc,ctdor,9);
        fprintf(out,"%d; %d; %d; %d ::v1 v2: %d %d \n",\
        vlct[0],vlct[1],vlct[2],vlct[3],v1,v2);
        fprintf(out,"no-04 %s \t",stg[ch]);
        fprintf(out,"clss: %s \n",stg[2]);
        clssg[ch][2]++;
        apglst1(axlp);
        if(th2==0)
        continue;
        else goto BCKG;
        }
}/*fim do if do 1º grupo de classificação-imagem global*/

/*Análise Defeito a Defeito*/
 axlpc=axlp;ctnm=0;
 while(axlpc!=NULL)
 {
        if(axlpc->area<=200 || axlpc->flag0==1)
        {axlpc=axlpc->prox;continue;}
        selctd3(axlpc,md1s);
 vtst=-2;
 nrg=-1;
 for(k=0;k<4;k++)
 {
                if(mgm[ctnm][3]>105) {nrg=1;ndef=2;break;}
                for(n=0;n<8;n++)
                res1[n]=0;
                for(i=0;i<op[k][0];i++)
                {
                 varc=0;mdia=0;
                for(n=0;n<vrv[k][0];n++)
                {
                bar(0,240,640,480);
                sprintf(ms,"i %d n %d op %d vrv %d vmd %3.2f md %3.2f",\
                i,n,op[k][i+1],vrv[k][n+1],vmdia[op[k][i+1]][vrv[k][n+1]],md1s[vrv[k][n+1]]);
                outtextxy(20,360,ms);
                varc=varc+vmdia[op[k][i+1]][vrv[k][n+1]]*vmdia[op[k][i+1]][vrv[k][n+1]];
                mdia=mdia+vmdia[op[k][i+1]][vrv[k][n+1]]*md1s[vrv[k][n+1]];
                }
```

```
                res1[op[k][i+1]]=mdia-varc/2;
                }
        ndef=op[k][1];
        vxl=res1[op[k][1]];
        for(i=1;i<op[k][0];i++)
        if(res1[op[k][i+1]]>vxl)
        {
                vxl=res1[op[k][i+1]];
                ndef=op[k][i+1];
        }
        if(ndef==seq[k])
        {
                if(ndef==3 && axlpc->area>8000 && mgm[ctnm][1]<40 &&\
                mgm[ctnm][0]>80 && mgm[ctnm][3]<50)
                {
                        fprintf(out,"anel area %d \n",axlpc->area);
                        nrg=1;ndef=5;vtst=-1;
                        break;
                }
        if((ndef==3 && mgm[ctnm][2]>3 && mgm[ctnm][3]>40)|| mgm[ctnm][3]>60)
        continue;
        if(ndef==0 && axlpc->ai>2 && axlpc->bi<198 && axlpc->aj>2 && axlpc->bj<317)
        {nrg=1;ndef=3;vtst=-1;break;}
        nrg=1;
        break;
        }
 }/*fim do for k*/
axpt=axlpc->proximo;
while(axpt!=NULL)
{
        putpixel(axpt->j,axpt->i,15);
        axpt=axpt->pximo;
}
if(nrg==1)
{
        varc=mdia=0;
        /*fprintf(out,"%d \t",ndef);*/
        switch(ndef)
        {
                case 2:if(indc[0]<=40) {ndef=1;vtst=1;break;}
                        varc=varc+vmdia[1][0]*vmdia[1][0]+vmdia[1][5]*vmdia[1][5]+\
                        vmdia[1][6]*vmdia[1][6];
                        mdia=mdia+vmdia[1][0]*md1s[0]+vmdia[1][5]*md1s[5]+\
                        vmdia[1][6]*md1s[6];
                        res1[1]=mdia-varc/2;
                        varc=mdia=0;
                        varc=varc+vmdia[2][0]*vmdia[2][0]+vmdia[2][5]*vmdia[2][5]+\
                        vmdia[2][6]*vmdia[2][6];
                        mdia=mdia+vmdia[2][0]*md1s[0]+vmdia[2][5]*md1s[5]+\
                        vmdia[2][6]*md1s[6];
                        res1[2]=mdia-varc/2;
                        if(res1[1]>res1[2]) ndef=1;
                        vtst=2;
                        break;
```

```
ccase 3: if(mgm[ctnm][0]>85 && mgm[ctnm][2]>3 && axlpc->area<3800)
                {vtst=20;ndef=4;break;}
                if(vtst==-1) {vtst=21;break;}
                varc=varc+vmdia[3][23]*vmdia[3][23]+vmdia[3][28]*vmdia[3][28]+\
                vmdia[3][32]*vmdia[3][32];
                mdia=mdia+vmdia[3][23]*md1s[23]+vmdia[3][28]*md1s[28]+\
                vmdia[3][32]*md1s[32];
                res1[3]=mdia-varc/2;
                varc=mdia=0;
                varc=varc+vmdia[2][23]*vmdia[2][23]+vmdia[2][28]*vmdia[2][28]+\
                vmdia[2][32]*vmdia[2][32];
                mdia=mdia+vmdia[2][23]*md1s[23]+vmdia[2][28]*md1s[28]+\
                vmdia[2][32]*md1s[32];
                res1[2]=mdia-varc/2;
                if(res1[2]>res1[3]) ndef=2;
                vtst=3;
                break;
 < case 4:if(axlpc->area>3800) {ndef=2;vtst=6;break;}
                if(mgm[ctnm][0]>80 && mgm[ctnm][2]>3 && axlpc->area<3800)
                {vtst=5;break;}
                 varc=varc+vmdia[4][10]*vmdia[4][10]+vmdia[4][13]*vmdia[4][13]+\
                vmdia[4][16]*vmdia[4][16];
                 mdia=mdia+vmdia[4][10]*md1s[10]+vmdia[4][13]*md1s[13]+\
                 vmdia[4][16]*md1s[16];
                 res1[4]=mdia-varc/2;
                 varc=mdia=0;
                 varc=varc+vmdia[2][10]*vmdia[2][10]+vmdia[2][13]*vmdia[2][13]+\
                 vmdia[2][16]*vmdia[2][16];
                 mdia=mdia+vmdia[2][10]*md1s[10]+vmdia[2][13]*md1s[13]+\
                 vmdia[2][16]*md1s[16];
                 res1[2]=mdia-varc/2;
                 if(res1[2]>res1[4]) ndef=2;
                 vtst=7;
                 break;
   case 5: if(vtst==-1) {vtst=22;break;}
                 if(mgm[ctnm][0]>80 && mgm[ctnm][2]>3 && axlpc->area<3800)
                 {vtst=8;ndef=4;break;}
                 varc=varc+vmdia[5][15]*vmdia[5][15]+vmdia[5][19]*vmdia[5][19]+\
                 vmdia[5][25]*vmdia[5][25];
                 mdia=mdia+vmdia[5][15]*md1s[15]+vmdia[5][19]*md1s[19]+\
                 vmdia[5][25]*md1s[25];
                 res1[5]=mdia-varc/2;
                 varc=mdia=0;
                 varc=varc+vmdia[2][15]*vmdia[2][15]+vmdia[2][19]*vmdia[2][19]+\
                 vmdia[2][25]*vmdia[2][25];
                 mdia=mdia+vmdia[2][15]*md1s[15]+vmdia[2][19]*md1s[19]+\
                 vmdia[2][25]*md1s[25];
                 res1[2]=mdia-varc/2;
                 if(res1[2]>res1[5]) ndef=2;
                 vtst=9;
                 break;
        }
if(ndef===0) vtst=10;
dplaydeef2(axlpc->aj,axlpc->ai,ndef);
```

```
            setfillstyle(SOLID_FILL,14);
          bar(0,240,640,480);
          setcolor(0);
          ch=grcdf(opcc,ctdor,ctnm);
          fprintf(out," no-1%d: %s;",vtst,stg[ch]);
          fprintf(out,"clss %s \t",stg[ndef]);
          clssdd[ch][ndef]++;
}
else
{
dplaydef2(axlpc->aj,axlpc->ai,2);
bar(0,240,640,480);
ch=grcdf(opcc,ctdor,ctnm);
fprintf(out,"no-100: %s;",stg[ch]);
fprintf(out,"%s \t",stg[2]);
clssdd[ch][2]++;
}
axlpc->proximo=apglista1(axlpc->proximo);
ctnm++;
axlpc=axlpc->prox;
}
apglst1(axlp);
BCKG:if(th2!=0)
{
flag=bnthresd(th2,1);
transfimg(flag,1);
conectividade();
transfimg(2,1);
Fecho1();
limpz1(2);
esqltzcf2();
axlp=axlpc=tpbin2(&nrg);

while(axlpc!=NULL)
 {
                    if(axlpc->area<=200 || axlpc->flag0==1)
                    {axlpc=axlpc->prox;continue;}
 axpt=axlpc->proximo;
 while(axpt!=NULL)
 {
        putpixel(axpt->j,axpt->i,15);
        axpt=axpt->pximo;
 }
 dplaydef2(axlpc->aj,axlpc->ai,0);
 axlpc->proximo=apglista1(axlpc->proximo);
 axlpc=axlpc->prox;
}
fprintf(out,"no-18: %s;",stg[0]);
fprintf(out,"%s \t",stg[0]);
clssdd[0][0]++;
apglst1(axlp);
}
fprintf(out,"\n");
}/*fim do while*/
```

```
SAID:fclose(out);

if((out=fopen("md1.txt","wt"))==NULL)
                    {
                     fprintf(stderr,"cannot open output file.\n");
                     exit(1);
                     }
for(n=0;n<8;n++)
{
         i=0;
         fprintf(out,"%s \t",stg[n]);
         for(j=0;j<8;j++)
         {
          fprintf(out," %d ",clssdd[n][j]);
          i=i+clssdd[n][j];
          }
          if(i==0) {fprintf(out,"\n"); continue;}
         mid=(float)(clssdd[n][n])*100/i;
         fprintf(out," :: %f \n",mid);
         }
 for(n=0;n<8;n++)
 {
         i=0;
         fprintf(out,"%s \t",stg[n]);
         for(j=0;j<8;j++)
         {
           fprintf(out," %d ",clssg[n][j]);
           i=i+clssg[n][j];
         }

         if(i==0) {fprintf(out,"\n"); continue;}
         mid=(float)(clssg[n][n])*100/i;
         fprintf(out," :: %f \n",mid);
         }
 fclose(out);
 fclose(out1);
 return(0);
}
/*************************************/
```

BIBLIOGRAFIA

REFERÊNCIAS BIBLIOGRÁFICAS

LIVROS [secç-A]

[1] Anil K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall International Editions, 1989.

[2] Albino de Carvalho, *Defeitos da Madeira-Formação e importância biológica*, Ministério da Economia-Secretaria de Estado da Indústria-Instituto Nacional de Investigação Indústrial, Lisboa 1970.

[3] Ballard, D. H. and C. M. Brown, *Computer Vision*, Prentice Hall, 1982

[4] Barret, Anyhony *Computer Vision and Image Processing* Chapman and Hall, 1989.

[5] Baxes, Gregory A. *Digital Image Processing*, Prentice Hall, 1984

[6] Bekkhoucha, Abdelkrim *Traitement d' Images et Analyse de Texture.Application a l'Imagerie Satellitaire*, These pour l'obtention du grade de docteur, Universite de Technologie de Compiegne, 1992

[7] Besançon,Jackes E. *Vision par Ordinateur en deux et trois dimensions* EYROLES ,1988.

[8] Bolc, L. Z. Kulpa, *Digital Image Processing System*, Springer Verlag, 1981.

[9] Bow, Sing-Tze *Pattern Recognition and Image Preprocessing* Marcel Dekker,Inc,1992

[10] Bugrov, Ia. S., Nikolski, S. M., *Matemática para Engenharia - Elementos de Álgebra e de Geometria Analítica*, Editora Mir, 1984.

[11] Cooley, W. P. Lohnes, *Multivariate Data Analysis*, JohnWiley and Sons,1971

[12] Dagnelie, Pierre, *Estatistica-Teoria e Métodos,* 1º e 2º V., Publicações Europa-América, 1973.

[13] Duda, R. P. Hart, *Pattern Classification and Scene Analysis,* Academic Press, 1972

[14] Faugeres, Q. D. Ed., *Fundamentals in Computer vision - Advanced Course*, Cambridge, UK:University Press, 1983

[15] Geoffrey J. Mclcchlan, *Discriminant Analysis and Statistical Pattern Recognition*, John Wiley & Sons, Inc., 1992.

[16] Gonzalez, R. P. Wintz, *Digital Image Processing*, ADDISON - WESLEY, AND SONS ed. 1987.

[17] Hall, E. L. *Computer Image Processing and Recognition,* New York, NY: Academic Press, 1979.

[18] Hanson, A. R. and E. M. Riseman, Eds., *Computer Vision Systems*, New York, NY:Academic Press, 1978

[19] Haralick, Robert M. Linda G. Shapiro *Computer And Robot Vision,* Addison-Wesley Publshing Company, 1992.

[20] James, Mike *Pattern Recognition,* John Wiley & Sons, 1988

[21] Kovalevsky, V. A. *Image Pattern Recognition,* Springer-Verlag, 1988.

[22] Leewis, Rhis *Practical Digital Image Processing* Ellis Horwood,1990

[23] Levine, M. D. *Vision in Man and Machine*, New York, NY:McGraw-Hill, 1985.

[24] Patrick Martin, *bois & Produtique,* Cépadues Éditions, juin 1992.

[25] Plastock, A. Roy, *Computação gráfica* Mc Graw-Hill, 1991.

[26] Pratt, W. K. *Digital Image Processing* , John Wiley & Sons, INC. New York , *second edition* 1991.

[27] Rosenfeld, A. and A. C. Kak, *Digital Picture Processing, second edition*, Academic Press, San Diego,CA, 1986.

[28] Rogers, David F., *Procedural Elements for Computer Graphics,* Mc Graw-Hill, 1985.

[29] Schalkoff, Robert J. *Digital Image - Processing and Computer Vision* John Wiley & Sons, 1989.

[30] Serra, J. *Image Analysis and Mathematical Morfhology* Academic Press Inc., London, 1982.

[31] Tomita, Fumiaki SaburoTsuji *Computer Analysis of Visual Textures* Kluwer Academic Publishers, 1990.

[32] Tou, J. T. R. C. Gonzalez, *Pattern Recognition Principles*, Addison Wesley, 1974.

[33] T. Y. Young and K. S. Fu, Eds.,*Handbook of Pattern Recognitionand Image Processing*, Orlando, FL:Academic Press, 1986.

[34] Wayne Niblack, *Digital Image Processing* Prentice/Hall International,1988

[35] Wirth, Niklaus *Algorithms + Data structures = Programs* Prentice-Hall,1987.

ARTIGOS EM REVISTAS [secç-B]

[1] Algazi, V. R. "Useful Approximations to optimum quantization", *IEEE Trans. Commun. technol.* Vol. COM - 14, p.297, June 1966.

[2] Andrews, H. C., A. G. Tesher, and R.P. Kruger, "Image Processing by Digital Computer", *IEEE Spectrum*, 9, 7 Julho 1972.

[3] Bourdier, L. M., "Capteurs pour le Bois:Comment les choisir", *CTBA*(Centre Technique du Bois et de l'Assemblement), nº24, Mai de 1989.

[4] Ciccotelli, J. et Bussy R., "Detection des Singularités du bois:Le point sur les systemes de vision", *CTBA* (Centre Technique du Bois et de l'Assemblement), nº25, Juillet de 1989.

[5] Ciccotelli, "Lambris de Pin Maritime:Une Machine de Vision pour les Classer", *CTBA*(Centre Technique du Bois et de l'Assemblement), nº25, Juillet de 1989.

[6] Cho, Tai-Hoon, Conners, R. W., Araman, P.A., "A Computer Vision System for Analyzing Images of Rough Hardwood Lumber", *IEEE*, 1990

[7] Choon-Woo Kim, Koivo, A. J.,"Hierarchical Classification of Surface Defects on Dusty Wood Boards," IEEE, 1990.

[8] Conners, R. W. and C. A. Harlow, "A theoretical comparison of texture algorithms,"*IEEE, Trans. Pattern Anal. machine Intell* Vol. PAMI - 2, pp. 204 - 222, May 1980.

[9] Conners, R. W., McMillin, C. W., Kingyao L., Vasquez-Espinosa, "Identifying and Locating Surface Defects in Wood:Part of an Automated Lumber Processing System", *IEEE, PAMI* Vol.5, nº6, 1983.

[10] Elias, P "Bounds on performances of optimum quantizers", IEEETrans. Inform. Theory, Vol. IT - 16, nº2, pp. 172 - 184, Mar. 1970.

[11] Fellow, A. R."Computer Vision: Basic Principles", *IEEE*, Vol. 76 nº8, August 1988.

[12] Frei, W. "Image Enhancement by Histogram Hyperbolization", *Computer Graphic and Image Processing*, 6, 3 June 1977

[13] Foley, D. H."Considerations of sample and feature size", IEEE Trans. Inform. Theory, Vol. IT - 18, pp. 618 - 626, Sept.1972.

[14] Laws, K. I., "Textured Image Segmentation" University of Southern California, Image Processing Institute, *USCIPI* Report 940, January 1980.

[15] Lego, J. P. "1989:Les Systemes de vision artificielle dans l' Industrie du bois", *CTBA*(Centre Technique du Bois et de l'Assemblement), nº22, Janvier de 1989.

[16] Haralick, R. M. K. Shanmugam, and I. H. Dinstein,"Textural features for image classification", *IEEE Trans. Syst; Man, Cybern*; Vol. SMC-3,pp 610 - 621, Nov.1973.

[17] Haralick, R. M. ,"Statistical and Structural Approches to Texture ", *proc. IEEE 67*, pp 786 - 809, May.1979.