

Faculdade de Engenharia da Universidade do Porto

Mestrado em Inteligência Artificial e Sistemas Inteligentes

Sistema de Visão para Controlo de Cadeira de Rodas Inteligente

Um sistema de visão adaptado para reconhecimento de
expressões faciais com algoritmos inteligentes

José Carlos Pinto Miranda

Dissertação submetida à Universidade do Porto para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Inteligência Artificial e Sistemas Inteligentes, elaborada sob a orientação do Professor Doutor Luís Paulo Gonçalves dos Reis do departamento de Engenharia Informática da Universidade do Porto

Porto, 2009

Resumo

A interacção homem-máquina tem se tornado cada vez mais um tema de importante relevo no desenvolvimento de sistemas inteligentes. A noção de que as máquinas podem e devem auxiliar o homem na multitude de tarefas que constituem a sua actividade quotidiana é um dado adquirido incontestável.

Por outro lado, a tecnologia informática, pelo seu progresso, não só se tem expandido a quase todos os domínios da actividade humana, abrangendo um maior número de pessoas, como também se tem tornado acessível no que respeita à relação preço/qualidade dos produtos que com ela se criam. É devido a este último aspecto que se gera uma “bola de neve” tecnológica onde a massificação de produtos tecnologicamente inovadores leva à criação de novas necessidades que por sua vez exigem novas aplicações.

Elementos de hardware/software que outrora pertenciam apenas à lavra de laboratórios de investigação conseguem-se hoje encontrar em trânsito nas ruas, miniaturizados na forma de aparelhos de última geração dos quais, alguns, conseguem lançar modas, e alterar hábitos sociais.

Esta banalização, alarga o espectro de aplicabilidade da tecnologia, podendo abrangerem-se áreas onde as limitações humanas são um obstáculo à qualidade de vida do cidadão comum. No caso dos fisicamente diminuídos (por causas genéticas, doenças crónicas ou mesmo por acidentes), o auxílio dedicado de sistemas inteligentes pode recuperar de forma significativa a qualidade de vida e auto estima deste tipo de utilizadores.

Em concreto, para os paraplégicos, um sistema de visão computacional capaz de interpretar expressões faciais e de as converter em acções (como por exemplo o comando de uma cadeira de rodas) baseado numa *Webcam* simples constitui um trabalho de interesse actual.

Este trabalho aponta procedimentos para a realização de um reconhecimento de expressões através: da introdução do conceito de máscara de utilizador, que permite uma orientação do esforço computacional para áreas bem definidas da imagem captada, em vez do todo; do conceito do filtro de macro-visão, que insensibiliza o sistema ao excesso de ruído; da estimativa de posicionamento tri-dimensional do utilizador por técnicas de regressão e do conceito de *reality checks*, por estimativa da continuidade do movimento.

Apresentam-se ainda os resultados obtidos quando aplicadas a um modelo de utilizador diferentes técnicas de inteligência artificial de classificação (para determinar o que os elementos expressivos do rosto estão a realizar) e de regressão (para estimar posicionamentos e ângulos). Apresenta-se também uma forma prática de recolher dados de um utilizador para construção dum modelo através do emprego de marcadores

Abstract

Man-machine interaction has become an increasingly more important topic in development of intelligent systems. The notion that machines can, and should, assist man in the multitude of tasks that constitute its everyday activity, is a fact beyond dispute.

Moreover, computer technology, and its progress, not only has expanded its applicability to almost all areas of human activity, and therefore covering a larger number of users, but has also become more accessible to the average family income. Due to the later, a snowball effect in technology development is deployed, where massification of technology leads to creation of new needs, which in turn, require new applications.

The average consumer can now use miniaturized equipments of the last generation, some of which powered new techno-fashions, and changing social habits, with hardware and software that once was only to be found in research laboratories. This trivialization extends the technology spectrum of applicability to areas where the physically challenged fail to achieve life quality standards that ordinary people take for granted. Dedicated assistance of intelligent systems can boost, in a significant way, their life quality and enhance their self-esteem.

Our Work aims specifically the issue of paraplegics assistance with a computerized vision system, based on a Webcam, able to interpret facial expressions and convert them into commands (such as the command of a wheelchair).

It points procedural guidelines to achieve facial expression recognition by: introducing a *usermask* concept, that allows algorithm efforts to be directed to well defined areas of interest within the captured image; proposing a *macro-vision* image filter that improves immunization on noisy environments; indicating estimation techniques to determine users tri-dimensional position and rotation angles; introducing *reality checks* techniques based on continuous motion estimation.

Artificial Intelligence techniques are applied trough a user model database construction, in order to study their results in classification (of the various face expression elements) and regression (in estimation of tri-dimensional positioning, movement and angles of the users face). In this context it is presented a simple method to collect data, based on face markers, during database construction.

Agradecimentos

São diversas as pessoas que tiveram um impacto na realização deste trabalho. De entre as que tiveram um impacto positivo, é difícil estabelecer uma hierarquia que permita com justiça uma enunciação coerente dos mesmos.

Passarei a mencionar estes marcos de motivação e apoio à medida que me ocorrerem e peço desculpa se omito alguém da lista. Estou certo que por mais que me esforce neste sentido, após a impressão, irei ser sempre confrontado com a incómoda constatação de que deixei alguém de fora e que, se pudesse voltar a reimprimir o(a) acrescentaria inquestionavelmente a esta lista. Passemos aos agradecimentos por isso.

Agradeço à minha esposa, Li Xin, pela sua imensurável paciência e apoio concedido ao longo de todo este longo e tortuoso projecto. Sem ela, de certeza que não teríamos este trabalho.

À minha família em geral, Li Danchan, Ana Sofia, por suportarem as variações de humor a que este trabalho me sujeitou e também um agradecimento especial pelas longas horas às quais me subtraí das respectivas vidas para o realizar, sem que disso se tivessem queixado.

Ao Alexandre Eduardo Miranda e ao Carlos Augusto Miranda, motivo pelo qual comecei a tirar o Mestrado já no distante ano de 1997, ainda na altura sob o nome de Mestrado em Inteligência Artificial e Computação e que lançou as raízes que mais tarde levaram à criação do Mestrado ao qual me proponho a obter o título correspondente com esta dissertação.

Aos vários Professores da componente lectiva do mestrado, que me proporcionaram um acesso mais rápido e seguro à panóplia de conhecimentos de que disponho actualmente e que me abriram a visão para a necessidade constante de procura de conhecimento face à insignificância do que recolhi.

Com um agradecimento especial ao orientador das teses que não defendi, Professor Doutor Eugénio Costa Oliveira, pela sua paciência.

Ao meu orientador, Professor Doutor Luís Paulo Reis, pela perseverança e fé de que em mim existe potencial para levar a bom porto uma tese de mestrado. Agradeço-lhe ainda pela simpatia, simplicidade e amizade que me ofereceu, sem nunca me apontar os inúmeros defeitos de inconstância na lavra da tese a que a vida familiar e profissional me moldou.

À direcção actual da Escola Secundária Carolina Michaëlis por, nesta fase tão crítica, ter aliviado a carga de trabalhos de forma a libertar física e mentalmente para a conclusão da tese.

A todos os meus amigos que, sempre com um sorriso, me disseram que sou capaz e que nunca duvidaram que mais tarde ou mais cedo haveria de acabar. Ao Inácio, por me ter oferecido a impressora laser onde imprimir o primeiro rascunho desta tese.

Aos que se ofereceram para me auxiliar nas várias tarefas deste trabalho.

A todos esses, sem excepção, ainda que em graus diferentes,

Muito Obrigado.

Índice

1.	Introdução	1
1.1	Enquadramento	1
1.2	Motivação.....	2
1.3	Objectivos	4
1.4	Estrutura da tese	4
2.	Descrição do Problema	5
2.1	A Expressão Facial	5
2.2	Sistema FACS	6
2.3	Restrições	12
2.3.1	Expressões Faciais Detectáveis e Expressões Faciais Utilizáveis	12
2.3.2	Limitações do Utilizador.....	13
2.3.3	Limitações de Processamento e Temporais.....	14
2.3.4	Meio Físico	16
2.4	Linguagem de Comando	17
2.5	Conclusões	19
3.	Cadeiras de Rodas	21
3.1	Evolução	21
3.2	Projecto Intellwheels	24
3.3	Conclusões	29
4.	Visão e Aprendizagem Computacional	31
4.1	Fases da visão computacional	32
4.1.1	Captura de imagem.....	32
4.1.2	Pré processamento	33
4.1.2.1	Transformações de espaço	34
4.1.2.2	Filtros.....	35
4.1.3	Extracção de componentes	37
4.1.4	Detecção, segmentação e processamento de Alto Nível.....	37

4.2	Aplicação de sistemas de visão computacional no controlo de cadeiras ..	38
4.3	Aprendizagem computacional.....	39
4.3.1	Ferramentas de exploração.....	40
4.3.1.1	RATTLE	41
4.3.2	Modelos de classificação.....	43
4.3.2.1	Árvores de classificação	44
4.3.2.2	Discriminantes lineares	45
4.3.2.3	Random forest	45
4.3.2.4	Redes neuronais.....	46
4.3.2.5	Máquinas de vectores de suporte.....	47
4.3.3	Avaliação de modelos de classificação.....	48
4.3.4	Modelos de regressão.....	49
4.3.4.1	– Regressão linear	50
4.3.4.2	- Árvores de regressão.....	50
4.3.4.3	- Redes neuronais de regressão.....	50
4.3.5	Avaliação de modelos de regressão	51
4.4	Conclusões	53
5.	O Sistema proposto	54
5.1	Pressupostos	54
5.2	<i>Normalização da máscara</i>	58
5.3	<i>Macro-visão</i>	62
5.4	Treino usando Marcadores.....	65
5.4.1	Factores de normalização	67
5.4.2	Regressão na estimativa das coordenadas dos marcadores.....	68
5.4.3	Estimativa do centróide inicial	70
5.4.4	Determinação dos ângulos.....	72
5.5	Reality Checks	73
5.5.1	Regressão para estimativa de rasto	75
5.6	Classificação individual dos componentes expressivos	77

5.6.1	Assistente de classificação	78
5.6.2	Uso de classificadores com o Rattle	81
5.6.2.1	Classificações para os olhos.....	81
5.6.2.2	Classificações do cenho.....	85
5.6.2.3	Classificações da boca.....	86
5.6.2.4	Classificações pelos ângulos	87
5.6.3	Classificação da expressão.....	88
5.7	Perspectiva global do sistema de visão.....	91
5.8	Conclusões	92
6.	Conclusão	94
6.1	Resultados principais.....	94
6.2	Trabalho futuro.....	95
6.3	Comentários finais	95
	Bibliografia	97

Lista de Figuras

FIGURA 2.1 - MUSCULATURA DO ROSTO (ADAPTADO DE [32])	6
FIGURA 3.1 - CADEIRA DO SÉC. XVIII (ADAPTADO DE [10])	21
FIGURA 3.2 - INTERNATIONAL SYMBOL OF ACCESS.....	22
FIGURA 3.3 -IMAGEM DE CADEIRAS MOTORIZADAS (ADAPTADOS DE [23]).....	23
FIGURA 3.4 - EXEMPLOS DE CADEIRAS DE MOBILIDADE ALARGADA (ADAPTADAS DE [25] E [34]).....	23
FIGURA 3.5 - CADEIRAS BÍPEDES E EXOESQUELETOS (ADAPT. [19],[33] E [18])	23
FIGURA 3.6 - CADEIRA COMANDADA POR INTERPRETAÇÃO DE IMPULSOS MENTAIS (ADAPTADO DE [14])	24
FIGURA 3.7 - ALGUNS PROTÓTIPOS NACIONAIS DE CADEIRAS MULTI-SENSORIAIS (ADAPTADO DE [38] E [4])	24
FIGURA 3.8 - COMPONENTES DO SISTEMA INTELLWHEELS (ADAPTADO DE [4]).....	25
FIGURA 3.9 - ARQUITECTURA DO PROJECTO INTELLWHEELS (ADAPTADO DE [4])	26
FIGURA 3.10 – DEFINIÇÃO DE SEQUÊNCIAS (ADAPTADO DE [21]).....	27
FIGURA 3.11 – INTERFACES ESPECÍFICAS (ADAPTADO DE [21])	28
FIGURA 3.12 – APLICAÇÃO PRINCIPAL (ADAPTADO DE [21])	28
FIGURA 4.1 PROJECTO DO INSTITUTO NACIONAL JAPONÊS DE CIÊNCIA E TECNOLOGIA INDUSTRIAL AVANÇADA (ADAPTADO DE [1]).....	32
FIGURA 4.2 – EXEMPLO DE UM FILTRO GAUSSIANO	36
FIGURA 4.3 – PONTOS DE VIZINHANÇA	36
FIGURA 4.4 - DEFINIÇÃO DE DADOS (DATA).....	41
FIGURA 4.5 - ÁRVORES BINÁRIAS V.S. ÁRVORES NÃO BINÁRIAS	44
FIGURA 4.6 - DISCRIMINANTE APLICADO A DUAS CLASSES	45
FIGURA 4.7 - NEURÓNIO ARTIFICIAL DE MCCULLOCH E WALTER PITTS.....	46
FIGURA 4.8 - REDE NEURONAL MULTICAMADA (ADAPTADO DE [30])	47
FIGURA 4.9 - SEPARAÇÃO POR MAXIMIZAÇÃO DA FRONTEIRA.....	47
FIGURA 5.1 – IMUTABILIDADE	55
FIGURA 5.2 – POSICIONAMENTO RELATIVO DOS ELEMENTOS A ANALISAR	56
FIGURA 5.3 – HOMEM VITRUVIANO (ILUSTRAÇÃO DE LEONARDO DA VINCI).....	57
FIGURA 5.4 – HOMEM VITRUVIANO – PROPORÇÕES DO ROSTO.....	58
FIGURA 5.5 – OCULTAÇÃO DE PORMENORES	58
FIGURA 5.6 - PERSPECTIVA	59

FIGURA 5.7 – EIXOS RELATIVOS	60
FIGURA 5.8 - ROTAÇÃO INDIVIDUAL DO EIXO ZZ	60
FIGURA 5.9 - ROTAÇÃO INDIVIDUAL DO EIXO XX	60
FIGURA 5.10 - ROTAÇÃO INDIVIDUAL DO EIXO YY	61
FIGURA 5.11 – NORMALIZAÇÃO DAS ÁREAS DE INTERESSE	61
FIGURA 5.12 – MACRO-VISÃO.....	63
FIGURA 5.13 – PONTOS DE RECOLHA NA MACRO-VISÃO	65
FIGURA 5.14 - EXEMPLOS DE MARCADORES	65
FIGURA 5.15 - MARCADORES IDENTIFICADOS	66
FIGURA 5.16 - MARCADORES NUMERADOS	66
FIGURA 5.18 - IMAGEM REMOVIDOS OS MARCADORES	67
FIGURA 5.19 - MACRO-VISÃO, REMOVIDOS OS MARCADORES	67
FIGURA 5.20 - ALTURA E LARGURA PARA EFEITO DE NORMALIZAÇÃO	68
FIGURA 5.21 – ESTIMATIVA DE POSICIONAMENTO	74
FIGURA 5.22 - EXEMPLO DE PORMENOR RECOLHIDO PARA CLASSIFICAÇÃO.....	78
FIGURA 5.17 - ÁREAS DE INTERESSE	78
FIGURA 5.23 - ASSISTENTE DE CLASSIFICAÇÃO	79
FIGURA 5.24 - EXEMPLO DE ÁRVORE DE DECISÃO	89
FIGURA 5.25 - ARQUITECTURA GLOBAL DO SISTEMA DE VISÃO	91

Lista de Tabelas

TABELA 2.1 – SOBRANCELHAS, PÁLPEBRAS E NARIZ (ADAPTADO DE [11])	7
TABELA 2.2 - FACS, AU DETECTÁVEIS ATRAVÉS DOS LÁBIOS, BOCA E QUEIXO (ADAPTADO DE [11]) ..	7
TABELA 2.3 - FACS, AU DETECTÁVEIS ATRAVÉS DOS LÁBIOS, BOCA E QUEIXO (ADAPTADO DE [11]) ..	8
TABELA 2.4 - FACS, AU DETECTÁVEIS POR DESLOCAÇÃO DOS OLHOS (ADAPTADO DE [11]).....	9
TABELA 2.5 - FACS, AU DETECTÁVEIS POR DESLOCAÇÃO DAS PÁLPEBRAS (ADAPTADO DE [11]).....	9
TABELA 2.6 – FACS, MOVIMENTOS DA CABEÇA (ADAPTADO DE [11]).....	10
TABELA 2.7 – EXEMPLO DE COMANDOS PARA UMA LINGUAGEM BASE (ADAPTADO DE [11]).....	18
TABELA 4.1 - MATRIZ DE CONFUSÃO: EXEMPLO PARA DUAS CLASSES	49
TABELA 5.1 - CORRELAÇÃO PARA A ESTIMATIVA DAS COORDENADAS DOS MARCADORES	69
TABELA 5.2 - ERRO NORMALIZADO NA ESTIMATIVA DAS COORDENADAS DOS MARCADORES.....	70
TABELA 5.3 - RESULTADOS DE REGRESSÃO – ROTAÇÃO SEGUNDO OS EIXOS	73
TABELA 5.4 - REGRESSÃO APLICADA AO RASTO DOS MARCADORES.....	76
TABELA 5.5 - DEGRADAÇÃO DO NÍVEL DE CONFIANÇA UTILIZANDO REDES NEURONAIS	76
TABELA 5.6 - DEGRADAÇÃO DO NÍVEL DE CONFIANÇA UTILIZANDO O MODELO LINEAR GENERALIZADO	77
TABELA 5.7 - DEGRADAÇÃO DO NÍVEL DE CONFIANÇA UTILIZANDO AS ÁRVORES DE REGRESSÃO.....	77
TABELA 5.8 - CARACTERIZAÇÃO DO CENHO	80
TABELA 5.9 - CARACTERIZAÇÃO DA BOCA.....	80
TABELA 5.10 - CARACTERIZAÇÃO DO OLHO ESQUERDO.....	80
TABELA 5.11 - CARACTERIZAÇÃO DO OLHO ESQUERDO.....	80
TABELA 5.12 – VARIÁVEIS COMBINADAS PARA CLASSIFICAÇÃO	81
TABELA 5.13 - UTILIZADOR MODELO - AU 61, 62, 63 E 43.....	82
TABELA 5.14 - UTILIZADOR MODELO - AU 5 E 6	82
TABELA 5.15 - CLASSIFICAÇÃO DO OLHO DIREITO.....	83
TABELA 5.16 - CLASSIFICAÇÃO DO OLHO ESQUERDO.....	83
TABELA 5.17 - RESUMO COMPARATIVO (100%/100%).....	84
TABELA 5.18 - RESUMO COMPARATIVO (70%/30%)	84
TABELA 5.19 - UTILIZADOR MODELO, AU 4	85
TABELA 5.20 - CLASSIFICAÇÃO DO CENHO	85
TABELA 5.21 - UTILIZADOR MODELO, AU 18,27 E 28	86

TABELA 5.22 – RESULTADOS DE CLASSIFICAÇÃO DA BOCA.....	87
TABELA 5.23 - UTILIZADOR MODELO - AU 55, 56, 51 E 52.....	88
TABELA 5.24 - CONFIANÇA ESPERADA NUM COMANDO.....	90

Capítulo 1

1. Introdução

1.1 Enquadramento

Todas as deficiências físicas afectam em maior ou menor grau a qualidade de vida de quem delas padece. Em particular as que afectam a parte motora do ser humano, reduzem as pessoas à dependência de terceiros perdendo-se também nessa dependência não só qualidade de vida mas também alguma da dignidade humana à qual cada um de nós tem direito. No sentido de mitigar este problema, que tanto pode ter causas de nascença, acidentais como mesmo degenerativas, foram-se criando ao longo do tempo soluções mais ou menos complexas consoante o tipo de restrição física.

No caso de paralisia dos membros inferiores ou inexistência dos mesmos, o recurso a cadeiras de rodas é a opção mais corrente para colmatar a falta de mobilidade. Aqui, a impulsão pelos membros superiores é a forma mais usual de conferir mobilidade à cadeira. A força nestes casos é exercida ou directamente nas rodas maiores da cadeira ou, mais raramente, num sistema parecido ao das engrenagens das tradicionais bicicletas, tendo os pedais (neste caso *manuals*) à altura dos braços. Contudo, levantam-se logo duas dificuldades. A primeira é que o recurso à impulsão pela força física dos braços, pode não ser opção viável em pacientes que, apesar de não sofrerem de paralisia a esse nível, já não dispõem da energia dos tempos da juventude. A segunda, é que muitas vezes a própria mobilidade dos membros superiores pode ser reduzida. Para estes casos, ou então simplesmente por disponibilidade financeira, existem as soluções motorizadas equipadas com baterias e motor eléctrico, que conferem à cadeira uma autonomia satisfatória. O controlo fica nestes casos a cargo de sistemas de comando como teclas direccionais ergonómicas adaptadas à mão, ou as tradicionais alavancas (*Joystick*) de fácil manuseio.

Todas estas soluções caem por terra pensando na sua utilização por indivíduos tetraplégicos. Apenas o recurso a sistemas computadorizados pode auxiliar este tipo de incapacitados. O reconhecimento de padrões de voz é uma das soluções possíveis. Mas mesmo nestes casos subsistem dois problemas. O primeiro advém do facto de poder não se tratar de opção viável, uma vez mais devido ao tipo de deficiência, ou pela especificidade do idioma nativo. O segundo, pela possível interferência ao nível de ruído, o que reduz a fidelidade do sistema, podendo mesmo ser comandado por acidente por terceiros contrariando as directivas do utilizador/piloto.

Chegando a este ponto, a solução seguinte passa pelo reconhecimento de comandos onde os problemas neurológicos atacam com maior raridade, ou seja na expressão facial e eventualmente através de movimentos da cabeça (se possíveis). Pelo estudo das expressões faciais intencionais que o doente pode realizar pode se construir uma tabela de equivalência entre elas e os comandos da cadeira. Com intuito de eliminar comandos erroneamente reconhecidos por uso de expressões reflexivas de estados emocionais alheios à intenção de comando da cadeira, a tabela deve ser lavrada com as que não são de se esperar que o doente realize num contexto corrente. Alternativamente poderá ser elaborada usando sequências simples de expressões de forma a garantir a intencionalidade do controlo.

Não importa a complexidade final do modelo de tradução das expressões em acções de controlo da cadeira de rodas (de alto ou baixo nível) todo o sistema falha se for excessivamente sensível a variações decorrentes do meio ambiente onde a iluminação e até a própria mobilidade do utilizador/piloto jogam um papel fundamental.

No respeitante às questões de iluminação a utilização dum filtro de imagem *canny*, permite insensibilizar, em grau razoável, o sistema às variações de luz. Em câmaras de baixa qualidade, com um mínimo de software de ajuste à luminosidade, o efeito da falta de iluminação é a produção de ruído excessivo ao passo que o excesso leva a um esbatimento dos pormenores por efeito de sobre exposição. Esses factores, após o filtro *canny*, traduzem-se em imagens com eventuais contornos excessivos (pelo efeito de ruído) ou na falta de contornos de pormenor (excesso de iluminação). Como por sua vez a imagem antes de aplicada à classificação ainda é submetida a um filtro de macrovisão, esses elementos pouco alteram a informação que se usa para efeito de classificação e regressão, e mesmo as variações introduzidas esbatem-se nos conjuntos de dados de treino dos diferentes modelos utilizados.

No que toca à questão da mobilidade do utilizador, o sistema propõe formas de estimar o posicionamento do utilizador face à câmara e aponta mecanismos de *reality check* que permitem aferir a qualidade da informação gerada face ao esperado tendo em linha de conta o rasto de movimentação do utilizador.

1.2 Motivação

Tenta-se com este trabalho alargar o tipo de interfaces existentes entre homem máquina, introduzindo uma interface de controlo não mecânico, para alcançar uma interacção mais profunda no comando natural de artefactos. A motivação principal do trabalho está na criação de tecnologia de auxílio a paraplégicos no sentido de lhes proporcionar maior autonomia através de um sistema de visão e reconhecimento de expressões faciais para o controlo duma cadeira de rodas.

Existem diversos sistemas de controlo já em prática que permitem a pessoas de mobilidade reduzida, o controlo parcial deste tipo de dispositivos. Nestes, encontramos *joysticks*,

reconhecimento de voz, reconhecimento de movimentos 3d entre outros. No caso concreto de sistemas de reconhecimento facial, estes sistemas utilizam a captura de imagem para processamento integral e em cada momento realizar o reconhecimento da expressão e respectivo comando. Sendo que estes tipos de algoritmos são pela sua natureza *computacionalmente pesados* implicam um compromisso entre o detalhe e a velocidade de resposta. Esse compromisso vai no sentido de permitir um reconhecimento de comandos no mais curto espaço de tempo sem que por isso se perca a qualidade/exactidão no reconhecimento do comando em si.

Este binómio, pode ser sempre resolvido com a elevação da capacidade computacional do sistema de apoio, o que se resolve facilmente pelo emprego de tecnologia de ponta, paralelismo de processos, etc. Em casos de necessidade mais extrema ainda existe o recurso à utilização de chips específicos que através de instruções de código máquina conseguem realizar uma boa parte do que habitualmente se faz através de programas convencionais.

De notar que no caso do controlo de uma cadeira de rodas o tempo de resposta do sistema tem de ser baixo e inversamente proporcional à velocidade de deslocação da própria cadeira. Para melhor se compreender esta relação basta entender que uma cadeira que se desloque a um metro por segundo se tiver o tempo de resposta de um segundo leva a que haja um desfasamento entre o utilizador e a reacção da cadeira de pelo menos um metro (distância de movimento descontrolado). Se a isso acrescentarmos a própria velocidade de reacção do utilizador a imprevistos (que dependendo da incapacidade física pode ser de bem mais demorada que 1 segundo) a distância de movimento descontrolado aumenta na razão directa e, neste caso, em um metro por segundo. Quanto maior for essa distância maior é a probabilidade de o utilizador sofrer ou causar um acidente.

Assim o referido binómio é de importância crucial ao ponto de fazer sentido em desenvolver trabalhos para conquistar tempo. Contudo este binómio, quando se emprega tecnologia de ponta como a acima indicada, lança um problema novo que impede a utilização deste tipo de sistemas de forma generalizada – tecnologia de ponta implica um aumento do custo global da solução.

Não é lógico esperar-se que um cidadão de posses modestas, depois de sofrer um acidente incapacitante de larga escala, veja ao seu alcance uma solução perfeitamente funcional que custa milhares de *euros*.

Se o custo de hardware é um factor que dispara com o aumento da qualidade e potencialidade do hardware (o que mistura diferentes variáveis como desempenho, autonomia e robustez), já o custo de utilização de *software* para o mesmo hardware não faz variar o custo da solução global no produto final.

A motivação deste trabalho vai no sentido de conceber um sistema com capacidade de resposta rápida, baseado em técnicas de reconhecimento de expressões adaptativas ao próprio movimento do utilizador, sem requerer o emprego de hardware demasiado específico. Esta introdução de tecnologia mais rápida ao nível dos algoritmos faz com que o custo a nível de hardware específico se possa reduzir até estar ao nível de pouco mais que um simples

computador pessoal de secretária (e quem sabe se num futuro próximo ao de um simples *microcontrolador*).

Assim, uma cadeira de rodas com *software* desta natureza, pouco mais cara é do que uma com eléctrica equipada com computador convencional e respectivos actuadores, passando a ser acessível a um leque muito mais alargado de utilizadores.

1.3 Objectivos

O principal objectivo principal deste trabalho é a análise da aplicabilidade de metodologias de inteligência artificial no âmbito de um sistema de visão computacional que permita o reconhecimento de comandos veiculados numa linguagem baseada em expressões faciais.

O trabalho tem igualmente como meta apontar as linhas orientadoras para a implementação deste tipo de sistema num cenário real - o controlo de uma cadeira de rodas de interface multimodal.

Nesse sentido é objectivo estudar técnicas de tratamento de imagem para possibilitar a extracção dos elementos expressivos referentes a um utilizador, que possam ser submetidos a um processamento subsequente por técnicas de inteligência artificial em classificação ou em estimativa de posicionamento.

É ainda objectivo abordar a questão da variância do meio envolvente, no sentido de tornar o sistema de visão mais imune à múltiplas dimensões de interferência que sucedem nos dispositivos de aquisição de imagem, quando aplicados a um contexto real.

Por último, é objectivo também a definição de formas práticas de lidar com as imperfeições do sistema de reconhecimento de expressões, com recurso a técnicas de inteligência artificial, para ser possível a implementação de um sistema capaz de uma resposta em tempo real útil com algum nível de confiança.

1.4 Estrutura da tese

Esta tese inicia-se com uma introdução, que constitui o presente capítulo. De seguida no capítulo 2, apresenta-se o problema da expressão facial. No terceiro capítulo cobre-se a temática das cadeiras de rodas e apresenta-se sumariamente o sistema onde será integrado o presente trabalho. O capítulo quarto detalha aspectos sobre a visão computacional em geral e aspectos sobre a respectiva aplicação ao trabalho presente. No capítulo cinco aborda-se o sistema de visão que é proposto por este trabalho nos seus aspectos de interesse geral e particular. No capítulo sexto, apresentam-se as conclusões e bem como propostas de trabalho futuro.

Capítulo 2

2. Descrição do Problema

Como referimos anteriormente o problema a tratar neste trabalho é o reconhecimento da expressão facial do utilizador de forma a ser possível a tradução em comandos.

Neste capítulo aborda-se a questão da expressão facial como um elemento natural do acto de comunicar e analisa-se um sistema de classificação dessas expressões. Ainda se referem alguns aspectos relacionados com as limitações de processamento, associadas a este tipo de problema e no final do capítulo apresenta-se um exemplo duma linguagem mínima baseada em expressões faciais.

2.1 A Expressão Facial

Sendo o elemento de eleição para este trabalho a expressão facial contém uma linguagem rica que permite comunicar a vários níveis. Ao nível consciente, consegue-se enfatizar emoções que se pretendem transmitir. Ao nível inconsciente, a expressão oferece inúmeros sinais de variação do estado emocional muitas das vezes apenas através de simples micro expressões. Não é por acaso que os bons jogadores de *Poker* são exímios em manter expressões neutras, sendo que só os melhores o conseguem na totalidade. Contudo, este trabalho está preocupado em lidar com as expressões que o utilizador pretende mostrar e como tal apenas com as expressões forçadas.

Como se pode observar pela , a musculatura do rosto é formada por diversos conjuntos que se podem controlar de forma independente, tal como na maioria da anatomia humana, sendo que na maioria dos casos esse controlo pode ser realizado intencional e individualmente. A diferença essencial para a maioria da restante anatomia humana é que esta estrutura não tem por objectivo a realização de esforços de potência (com a excepção do controlo maxilar), pelo que reagem em velocidade.

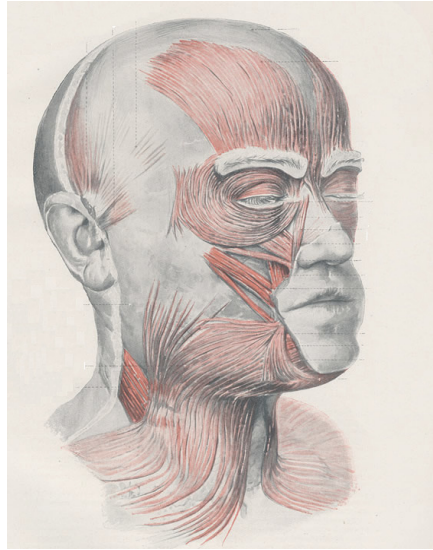


Figura 2.1 - Musculatura do rosto (adaptado de [32])

O essencial é que a expressão é constituída pela articulação de diversos elementos que no seu todo permitem uma variedade considerável de diferentes conjuntos.

Uma caracterização correcta desses conjuntos, separando as expressões inconscientes das conscientes, permite construir uma tabela de correspondências expressão/comando que pode ser utilizada no controlo duma cadeira de rodas.

2.2 Sistema FACS

O sistema *FACS – Facial Action Coding System* (sistema de codificação da acção facial) – foi desenvolvido por Ekman e Friesen em 1978 [12], tem por objectivo classificar alguns aspectos das expressões faciais de forma a construir uma espécie de vocabulário sobre a matéria.

Este sistema contempla aspectos de pormenor excessivo e demasiado extenso para que se possam inserir neste contexto, pelo que nos vamos limitar apresentar as tabelas reduzidas de unidades de acção relativas aos elementos expressivos que nos interessam em particular.

Nestas a Tabela 2.1, refere-se à parte superior do rosto compreendendo as sobrancelhas, as pálpebras e ainda a parte superior do nariz. Na tabela a coluna “AU” refere o número da *Action Unit*, ou seja da unidade de acção em causa, identificador único dentro do sistema FACS. A descrição, encontra-se indicada em com referência ao original em *italico*, pela pobreza que se introduz no idioma alvo, após a tradução de alguns dos termos. A imagem exemplo serve para complementar a descrição e foi extraída das ilustrações publicadas do próprio sistema FACS.

AU	Descrição	Imagem exemplo
1	Elevação das sobrancelhas por dentro <i>Inner Brow Raiser; Frontalis, pars medialis</i>	
2	Elevação das sobrancelhas por fora <i>Outer Brow Raiser; Frontalis, pars lateralis</i>	
4	Baixas das sobrancelhas <i>Brow Lowerer; Corrugator supercilii, Depressor supercilii</i>	
5	Elevar das palpebras <i>Upper Lid Raiser; Levator palpebrae superioris</i>	
6	Elevar das bochechas <i>Cheek Raiser; Orbicularis oculi, pars orbitalis</i>	
7	Semicerrar as palpebras <i>Lid Tightener; Orbicularis oculi, pars palpebralis</i>	
9	Enrugar o nariz <i>Nose Wrinkler; Levator labii superioris alaeque nasi</i>	

Tabela 2.1 – Sobrancelhas, pálpebras e nariz¹ (adaptado de [11])

Tabela 2.2 - FACS, AU detectáveis através dos lábios, boca e queixo (adaptado de [11])

AU	Descrição	Imagem exemplo
10	Elevar do lábio superior <i>Upper Lip Raiser; Levator labii superioris</i>	
11	Descida naso labial <i>Nasolabial Deepener; Zygomaticus minor</i>	
12	Puxador dos cantos dos lábios <i>Lip Corner Puller; Zygomaticus major</i>	

¹ Na tabela a coluna AU são as siglas de *Action Unit*

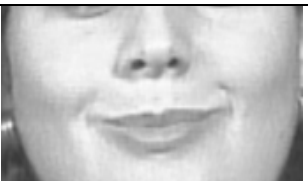
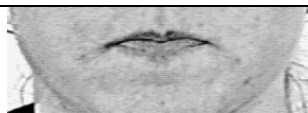
AU	Descrição	Imagem exemplo
13	Inchar das bochechas <i>Cheek Puffer; Levator anguli oris (a.k.a. Caninus)</i>	
14	Covinhas <i>Dimpler; Buccinator</i>	
15	Depressor dos cantos labiais <i>Lip Corner Depressor; Depressor anguli oris (a.k.a. Triangularis)</i>	
18	Franzidor de lábios <i>Lip Puckerer; Incisivii labii superioris and Incisivii labii inferioris</i>	
20	Esticador de lábios <i>Lip stretcher; Risorius w/ platysma</i>	
22	Funilador de lábios <i>Lip Funneler; Orbicularis oris</i>	
23	Estreitador de lábios <i>Lip Tightener; Orbicularis oris</i>	

Tabela 2.3 - FACS, AU detectáveis através dos lábios, boca e queixo (adaptado de [11])

AU	Descrição	Imagem exemplo
24	Compressor de lábios <i>Lip Pressor; Orbicularis oris</i>	
25	Depressor do lábio inferior <i>Lips part*; Depressor labii inferioris or relaxation of Mentalis, or Orbicularis oris</i>	
26	Queixo caído <i>Jaw Drop; Masseter, relaxed Temporalis and internal Pterygoid</i>	

(Continuação)

AU	Descrição	Imagem exemplo
27	Boca aberta <i>Mouth Stretch; Pterygoids, Digastric</i>	
28	Sucção dos lábios <i>Lip Suck; Orbicularis oris</i>	

A Tabela 2.2 e a Tabela 2.3 ilustram as unidades de acção, realizáveis através dos lábios, boca e queixo. Também incluem as bochechas, ainda que em conjunção com os lábios e por consequência de impossibilidade de movimentação autónoma dos outros elementos.

Na Tabela 2.4 ilustra as unidades de acção que resultam da movimentação da pupila dos olhos.

A Tabela 2.5 refere-se às unidades de acção identificáveis pela deslocação das pálpebras.

Tabela 2.4 - FACS, AU detectáveis por deslocação dos olhos (adaptado de [11])

AU	Descrição	Imagem exemplo
61	Olhos para a esquerda <i>Eyes turn left;</i>	
62	Olhos para a direita <i>Eyes turn right;</i>	
63	Olhos para cima <i>Eyes up;</i>	
64	Olhos para baixo <i>Eyes down;</i>	

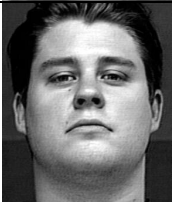
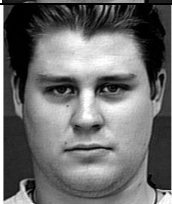
Tabela 2.5 - FACS, AU detectáveis por deslocação das pálpebras (adaptado de [11])

AU	Descrição	Imagem exemplo
41	Relaxamento das pálpebras <i>Lid droop; Relaxation of Levator palpebrae superioris</i>	
42	Fresta aberta dos olhos <i>Slit; Orbicularis oculi</i>	

(Continuação)

AU	Descrição	Imagem exemplo
43	Olhos fechados <i>Eyes Closed; Relaxation of Levator palpebrae superioris Orbicularis oculi, pars palpebralis</i>	
44	Olhos Semicerrados <i>Squint; Orbicularis oculi, pars palpebralis</i>	
45	Bater as palpebras <i>Blink; Relaxation of Levator palpebrae superioris Orbicularis oculi, pars palpebralis</i>	
46	Piscar dum olho <i>Wink; Relaxation of Levator palpebrae superioris Orbicularis oculi</i>	

Tabela 2.6 – FACS, movimentos da cabeça (adaptado de [11])

AU	Descrição	Imagem exemplo
51	Cabeça voltada à esquerda <i>Head turn left;</i>	
52	Cabeça voltada à direita <i>Head turn right;</i>	
53	Cabeça para cima <i>Head up;</i>	
54	Cabeça para baixo <i>Head down;</i>	

(Continuação)

AU	Descrição	Imagem exemplo
55	Inclinação à esquerda <i>Head tilt left;</i>	
56	Inclinação à direita <i>Head tilt right;</i>	
57	Inclinação para a frente <i>Head forward;</i>	
58	Inclinação para trás <i>Head back;</i>	

A Tabela 2.6 já é mais controversa. A razão está no simples facto de não podermos propriamente designar movimentos da cabeça como expressões faciais. Em todo o caso, e porque este tipo de movimentos acompanha com frequência o discurso humano normal, alguns de forma instintiva outros simplesmente para vincar a ênfase do discurso, é aceitável que figurem como unidades de acção próprias.

Como se pode perceber pela tabela acima indicada, nem todas as particularidades do rosto humano se prestam à formação de vocábulos.

Por exemplo o nariz pouco ou nada traz para a expressão facial. Apenas poderia ser incluído pelo enrugar do cenho. Eventualmente ainda poderíamos detectar a acção de dilatação das narinas o que seria um elemento difícil de discernir computacionalmente. Contudo, mesmo que fosse incluído para a produção dum vocábulo ainda subsiste o facto de ser praticamente impossível produzir essa dilatação sem denunciar a mesma pela movimentação involuntária de outros elementos no rosto.

Aos elementos do rosto que se podem movimentar de forma isolada² passamos a designar, para efeitos deste trabalho, por elementos expressivos ou componentes expressivos.

² Note-se que o emprego da palavra “isolada” não é de todo rigorosa neste contexto, uma vez que a musculatura do rosto está de tal forma interligada que é quase impossível acção numa expressão apenas o que pretendemos. Qualquer destas acções conscientes produz micro

Note-se que no sistema FACS existem unidades de acção muito difíceis de distinguir entre si, como é o caso de algumas baseadas nas pálpebras (Au42 e Au43) ou nos lábios (Au17 com Au20 e ainda Au25 com Au26).

Por este motivo, ainda que a classificação se sirva do sistema FACS para representação interna nas redes neuronais, o importante é que a acção que o utilizador deseja realizar possa ser discernida sem dificuldade. Se alguma das unidades de acção pela sua proximidade face ao utilizador da cadeira se confundem, em termos de comando da cadeira pouca ou nenhuma diferença fazem pois o importante é o resultado final.

O sistema pode identificar as AU consoante as condições em que se apresente o modelo, duma ou de outra forma, a associação final destas com o comando de alto nível é que constitui a configuração do utilizador, e como tal não são relevantes esses pormenores. A diferença é se a acção de alto nível será desencadeada através de uma única Au, ou de todas as que se confundem para um dado utilizador.

Por outro lado, importa sempre não esquecer de que estamos perante um sistema multimodal, onde cada entrada dum dispositivo de captação pode ser complementada ou sobreposta pelas demais, consoante as determinações de alto nível desses sistema multimodal.

Não convém também esquecer que tanto é importante reconhecer um comando resultante da variação da entrada num dispositivo dos disponíveis para a cadeira num dado momento, como a sequencia dessas variações, o que minimiza a possibilidade de comando por erro de percepção.

2.3 Restrições

2.3.1 Expressões Faciais Detectáveis e Expressões Faciais Utilizáveis

Do conjunto de componentes expressivos indicados, nem todos tem aplicabilidade neste contexto. A necessidade de distinção de uma expressão intencional e de uma resultante de uma reacção involuntária, leva a que seja necessário que se mantenha fixa por um determinado lapso mínimo de tempo. Esta característica elimina as expressões resultantes por exemplo dos olhos fechados uma vez que o contexto de aplicação é dinâmico exigindo um controlo reactivo por parte do utilizador.

Por outro lado, há elementos expressivos que são particularmente difíceis de distinguir entre si pelo menos a nível computacional, pelo que a totalidade das acima apresentadas terá de

espasmos musculares noutras zonas do rosto. Por outro lado, na zona onde se insere a estrutura base dos músculos que comandamos voluntariamente da face produz-se à volta movimentações da face resultantes da elasticidade (ou falta de) da pele.

ser reduzida a um conjunto menor criando um maior contraste. Note-se que o sistema FACS apresenta uma lista extensiva de todo o tipo de expressões que é possível identificar num utilizador abstraindo-nos das particularidades desse mesmo utilizador. Convém por isso não esquecer que o que resulta como uma expressão controlada em alguém pode mais não ser que uma expressão natural noutro indivíduo. Essas expressões são normalmente reflexo do estado de espírito dominante do indivíduo, pelo que ainda que sejam detectáveis para um utilizador em particular, o contraste expressivo das mesmas não é suficiente para discernir a expressão controlada que o utilizador quer impor. Ainda que seja possível assinalar para esse utilizador uma determinada unidade de acção, de acordo com o FACS, a intencionalidade dessa acção pode ser nula, o que leva a considerar o problema das unidades de acção utilizáveis.

Contudo, uma vez distinguido um subconjunto que possa ser facilmente discriminado a nível algorítmico, entramos numa linguagem com carácter bem definido, e como sucede com qualquer linguagem com dois ou mais símbolos, abre-se caminho para a representação para a formação dum código de comunicação completo.

2.3.2 Limitações do Utilizador

Cada utilizador é um caso diferente. Desde os que não tem qualquer incapacidade fora do normal aos que nitidamente necessitam de correcções específicas para poderem comunicar.

De entre estes existem múltiplas variantes que dentro da incapacidade motora os tornam casos especiais. Por exemplo vítimas de acidentes de viação, para além de paraplégicos, podem apresentar deformações no rosto que tornam as expressões pouco identificáveis, já para não falar dos que sofreram de queimaduras dos tecidos ao ponto de mudarem de pigmentação ou de forma.

Para além destes casos existem ainda os derivados de enfermidades degenerativas quer dos tecidos musculares, quer do sistema nervoso que os controla quer ainda de ambos. Paralisia parcial ou total dos músculos da cara, motivada sobretudo por doenças como a paralisia de bell, a trombose, o tumor cerebral ou a doença de Parkinson; ou movimentos descontrolados da cara, que se devem sobretudo a doenças neurológicas como a esclerose múltipla, paralisia cerebral e coréia³. [24]

Se reflectirmos nestas variantes, começamos a entender que as limitações podem ser comprometedoras se o sistema for concebido com demasiada rigidez. Numa fase de melhoramento de um sistema de visão adaptado a um utilizador com este tipo de problema, é

³ A coréia é caracterizada por movimentos rápidos, amplos, irregulares e imprevisíveis comprometendo os músculos da face, os músculos bulbares, as partes próximas das extremidades e os dedos dos pés e das mãos. Estas características podem ser agravadas mediante *stress*, excitação e febre. É um dos três grupos em que se tipifica a Paralisia Cerebral Discinética.

possível criar modelos de inteligência artificial (por exemplo treinar séries temporais) para determinar a diferença entre um movimento descontrolado dos demais. Mesmo o simples registo do rasto do utilizador com a respectiva classificação em forma de *voluntário* e *involuntário* para criar conjuntos de dados adaptados a cada doença, permite ultrapassar o problema do reconhecimento por erro de um comando devido a acções involuntárias do utilizador.

2.3.3 Limitações de Processamento e Temporais

Um dos problemas mais severos do processamento de imagem é o tempo necessário para a execução da maioria dos algoritmos que fazem reconhecimento de características. A complexidade destes algoritmos pode ser maior ou menor, sendo que em alguns casos inviabilizam a sua utilização em tempo real, exigindo não só imagens estáticas como tempos avultados para a respectiva identificação de elementos.

Ora sendo o objectivo o de criar uma interface com capacidade de controlo em tempo real numa cadeira de rodas, temos de levar em conta que a simplificação destes processos tem de ser uma das directivas principais do projecto.

Essa simplificação contudo, não se pode fazer tendo por custo o grau de rigor dos resultados obtidos, uma vez que para além de determinado grau de confiança o sistema torna-se inútil. Convém sempre não esquecer que estamos a entregar uma pessoa incapaz de acções e reacções motoras às deliberações de uma *prótese muscular* assistida por computador, com todas as consequências que isso implica.

Portanto um compromisso entre tempo de resposta e qualidade de resposta é a essência para o sucesso do sistema.

Esse compromisso terá de ser obtido pela estimativa de margem de segurança que se pretende obter e depende não só do utilizador mas também da velocidade máxima a imprimir à cadeira.

Assim para chegar a um valor aceitável podemos considerar as seguintes variáveis:

$V_{\max}$ - Velocidade máxima da cadeira em condições de segurança;

D_{seg} - Distância máxima que a cadeira pode percorrer sem controlo do utilizador dentro dum limite de tolerância (margem de segurança de movimento descontrolado);

T_a - Maior mais longo de resposta do sistema de actuadores mecânicos/electrónicos quando a cadeira se desloca à velocidade máxima;

T_e - Tempo mais longo que o paciente leva a esboçar uma das expressões de controlo do sistema;

T_{ri} - Tempo médio de reacção do utilizador entre a ocorrência de um elemento imprevisto, selecção consciente da expressão de controlo da cadeira e o início de formação da expressão;

A forma de determinar o tempo de resposta do sistema T_r é dada pela fórmula:

$$T_r = \frac{D_{seg}}{V_{max}} - T_a - T_e - T_{ri} \quad (Exp. 2.1)$$

Em termos de exemplo, e por conveniência de cálculo, podemos considerar o caso de uma velocidade máxima aceitável para a cadeira de $V_{max} = 0.5 \text{ m/s}$, o que se pode considerar já como uma velocidade vertiginosa para um paraplégico.

Podemos ainda assumir (por mera convenção) que seria interessante que a cadeira não percorresse mais de um metro à velocidade máxima sem poder ser controlada pelo utilizador, ou seja um $D_{seg} = 1 \text{ m}$.

Para um tempo de actuação *electromecânico* de por exemplo $T_a = 0.5 \text{ s}$, temos já um sistema que consegue vencer a inércia global da massa envolvida em tempo bastante reduzido.

Se colocarmos como tempo mais longo que o paciente leva a esboçar uma das expressões de controlo do sistema num valor razoável de $T_e = 0.2 \text{ s}$ e para tempo médio de reacção $T_{ri} = 0.5 \text{ s}$ temos a seguinte expressão⁴:

$$T_r = \frac{D_{seg}}{V_{max}} - T_a - T_e - T_{ri} = \frac{1}{0.5} - 0.5 - 0.2 - 0.5 = 0.8 \quad (Exp. 2.2)$$

Ou seja, para o cenário definido, o tempo máximo de processamento associado a um comando teria de ser de 0.8 s ⁵.

⁴ Contributo do autor

⁵ Na recolha de dados para efeitos de criação dos conjuntos de dados, a utilização do filtro *canny* e o filtro de *macrovisão*, para uma máquina portátil *monoprocessador Sempron 2600+ (1,6Ghz)* os tempos médios de execução para cada frame situaram-se bem abaixo desse limiar, estando em média próximos dos $0,05 \text{ s}$. Note-se que os algoritmos utilizados não careciam de optimização para a criação dos conjuntos de dados de treino, e entre outras funcionalidades apresentam processamentos desnecessários resultantes da exibição de janelas com a informação tratada, e como tal ainda poderão ser optimizados para melhorar o desempenho global. Mesmo assim, situam-se bem abaixo da fasquia dos $0,8$ segundos. Claro que no sistema final ainda é necessária a aplicação dos classificadores e métodos de regressão. Na maioria dos casos são operações praticamente de cálculo directo sem iterações, com tempos de execução quase insignificantes neste contexto.

É evidente que se podem calcular valores diferentes ajustando as variáveis, sendo que importante é manter esses valores dentro de parâmetros válidos.

Por exemplo o tempo de actuação da cadeira poderia ser desprezado em alguns dos comandos. Contudo o comando de paragem, se não estivesse associado a um tempo de actuação mínimo, equivaleria a uma paragem abrupta e por muito baixa que fosse a velocidade, redundaria no perigo de projecção do ocupante, o que seria inaceitável.

Caso não se consiga chegar a um tempo de reacção positivo, ou suficiente, resta apenas uma de três soluções. A primeira é reduzir a velocidade máxima da cadeira, a segunda é aumentar a velocidade de processamento. A terceira é utilizar algoritmos mais eficientes.

No que respeita aos algoritmos, a maioria deles é parametrizável para que o tempo de execução diminua, sacrificando a qualidade de resultado. No entanto o que verdadeiramente consome mais tempo é aquele que se perde ao processar elementos que não são estritamente o alvo dos algoritmos de reconhecimento finais. Nestas tarefas os algoritmos têm de separar o que interessa do que é simples ruído ocasional da captura de imagem ou elemento eventual extra do meio envolvente.

E é aqui que se tem de garantir tempos aceitáveis com qualidade mínima.

2.3.4 Meio Físico

O meio de aplicação é uma cadeira de rodas robótica pelo que o sistema tem de ser suficientemente pequeno para poder ser nela integrado. Por outro lado, a alimentação do sistema tem de garantir ao mesmo uma autonomia suficiente para que o conjunto possa manter-se operacional durante tempo útil entre duas recargas sucessivas, o que implica baixo consumo e por consequência emprego de tecnologia de desempenho menor.

Convém destacar que o consumo energético para componentes de igual tecnologia de integração é proporcional à frequência de operação (o chamado *System Clock*), o que leva a que máquinas de topo de gama apresentem consumos exagerados de energia. Assim uma boa solução tem de ser pequena, leve e de preferência de baixo consumo, para além de manter apesar disso um desempenho aceitável.

Outro aspecto é que o sistema tem de operar em condições completamente imprevisíveis em vários aspectos:

- Piso – podendo este ter as mais variadas características desde os irregulares que causam macrovibrações na estrutura da cadeira, aos padronizados que introduzem ressonâncias no chassis como ainda aos de padrões aleatórios cujos efeitos podem ser quaisquer a nível de impacto na cadeira. Estas vibrações causam instabilidade óbvia na recolha de imagem devido a variações súbitas quer no posicionamento do rosto, quer pela deslocação da câmara produzindo efeitos esborratadas nos mapas de bits recolhidos;

- Iluminação – sendo este um factor de bastante impacto na maioria dos algoritmos de processamento de imagens, devido às variações que provocam ao nível da recolha da matriz base. Aqui o cenário pode ser de espaço abundantemente iluminado como por exemplo campo aberto sem sombras, ou o oposto, salas escuras sem iluminação. Podem ainda ocorrer situações de iluminação artificial que colidam com a taxa de refrescamento da imagem recolhida, como é o caso de luzes baseadas em lâmpadas de Xenon (estroboscópicas em que caso a recolha de imagem for sincronizada com a frequência da própria lâmpada, e porque nas câmaras não existe retina, o efeito é de ausência de luz). Existe ainda a iluminação resultante de lâmpadas coloridas, como é o caso da maioria da iluminação nocturna urbana, dum laranja mortiço. Ainda temos por exemplo a iluminação caótica resultante da travessia da luz por gradeados, arbustos, árvores, postes, etc. . E por último ainda há a contar o efeito de luzes parasitas que podem incidir directamente nas próprias câmaras, como lâmpadas de automóveis, candeeiros, feixes infravermelhos, etc.;

- Atmosfera – aqui locais com fumo, nevoeiro ou mesmo a simples chuva, podem distorcer por completo a recolha de informação. Note-se que não se pode criar um sistema que funcione na perfeição quando o paraplégico sai de casa e de repente ao primeiro sinal de chuva deixe por completo de reconhecer o dono. O fumo é igualmente um dos factores possíveis de influenciar a fiabilidade do sistema. Ainda poderíamos referir outros imprevistos como pó de obras e outros elementos atmosféricos parasitas;

- Utilizador – mesmo aqui, podem ocorrer elementos imprevisíveis como é o caso de surgimento de barba, suor no rosto ou mesmo até lágrimas, correntes de vento que despenteiem o utilizador projectando-lhe cabelos à frente do rosto, etc... .

2.4 Linguagem de Comando

Um sistema que tenha pretensões adaptativas ao utilizador deve passar por um processo de aprendizagem ou pelo menos de pré-configuração para que se possa estabelecer uma interface eficaz.

Por exemplo se numa configuração da cadeira o inclinar a cabeça para a direita corresponder a uma viragem para a direita da cadeira, de forma fixa, no caso da paralisia abranger o pescoço, todo o sistema se tornaria inútil.

Assim, tem de existir um processo de aprendizagem prévio adaptado ao utilizador.

Apenas a título demonstrativo deixamos uma sugestão para essa linguagem base, conforme a Tabela 2.7.

Evidentemente que nada é perfeito. Basta imaginar um prurido no olho esquerdo, resultante por exemplo duma poeira, e eis que a cadeira inicia uma sucessão de círculos obedientes para a esquerda uma vez que o utilizador não pára de piscar o olho.

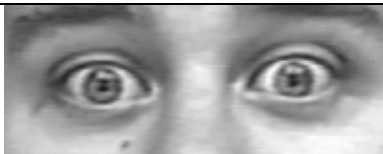
 <i>piscar olho esquerdo (7+42)</i>	<i>Virar direita</i>
 <i>piscar olho direito (42+7)</i>	<i>Virar esquerda</i>
 <i>Olhos esbugalhados 5</i>  <i>Lábios comprimidos 28</i>	<i>encetar deslocamento para frente</i>
 <i>Olhos esbugalhados</i>  <i>Boca aberta 27</i>	<i>Parar</i>
 <i>Baixas das sobrancelhas 4</i>	<i>Começar a aceitar instruções de comando ou Imobilizar e parar de aceitar instruções</i>

Tabela 2.7 – Exemplo de comandos para uma linguagem base (adaptado de [11])

Em todo o caso as expressões escolhidas duma maneira geral raramente resultam como expressões reflexas, com a excepção possivelmente da instrução de paragem que é praticamente a mesma de espanto e que por isso mesmo pode ser interpretada como situação imprevista onde o utilizador não tem tempo de pensar para conscientemente comandar a cadeira (por consequência o melhor é parar). Importante é também uma escolha de expressões que em todo o

momento garantam que o utilizador possa ver o que tem pela frente, excluindo-se por isso qualquer uma que seja resultante dos olhos fechados.

A necessidade de um comando para o sistema começar e parar de aceitar instruções surge, como facilmente se entende, da possibilidade de serem “mal interpretadas” expressões feitas quer involuntariamente, quer voluntariamente mas no decorrer de uma conversa.

A necessidade de um comando para o sistema começar e parar de aceitar instruções surge, como facilmente se entende, da possibilidade de serem “mal interpretadas” expressões feitas quer involuntariamente, quer voluntariamente mas no decorrer de uma conversa.

Estes comandos são os de controlo imediato ou seja de baixo nível e por isso baseiam-se em unidades expressivas directas. Contudo a linguagem de controlo pode ser alargada permitindo a construção de comandos mais complexos por encadeamentos adicionais com prefixos e terminadores.

Assim desde que haja um mínimo de duas expressões diferentes que possam ser reconhecidas entre as expressões delimitadoras, existe a possibilidade de construção de frases completas (que no pior de todos os casos recorrem a uma linguagem binária). Usando este princípio esta extensão da linguagem poderá permitir comandar a cadeira com instruções de alto nível, desde que haja algoritmos de navegação e controlo respectivos (bem como sensores adequados) como os que se encontram num vulgar robô autónomo móvel:

Seguir parede à direita

Passar pela porta em frente

Seguir caminho pré-definido para casa

Seguir elemento

...

Qualquer um destes comandos de alto nível, tirando os que exigem sensores específicos, não elevam o custo do hardware associado, pelo que podem constituir um trabalho interessante para desenvolvimento futuro de cadeiras inteligentes.

Numa fase posterior, quando o reconhecimento do sistema alcança níveis de confiança/qualidade que permitem discernir as indicações do utilizador com precisão, podem aplicar-se treinos de sequências a fim de garantir a exclusão de erros de interpretação devidos a reacções instintivas ou inadvertidas do utilizador.

2.5 Conclusões

Neste capítulo apresentou-se o conjunto de factores que tipificam o problema da utilização de expressões faciais no contexto da construção de uma linguagem de controlo.

Iniciou-se a exposição com a apresentação de um sistema de classificação de expressões faciais e dentro dele apontou-se um conjunto de elementos que posteriormente se agruparam tendo por critério o tipo de elemento que se utiliza neles para delinear a expressão.

Enalteceu-se a diferença entre a classificação de expressões no sentido de constituir um dicionário de expressões reconhecíveis e as expressões que são utilizáveis para um sistema de reconhecimento automático.

Apontaram-se ainda as dificuldades que advém de um sistema de reconhecimento de expressões faciais, destacando-se a necessidade de reacção dum sistema em tempo útil, com a formulação de uma métrica de definição do lapso de tempo máximo aceitável em função de algumas propriedades do sistema.

A título exemplificativo, desenhou-se uma mini-linguagem, hipotética, para controlo simples duma cadeira de rodas, seleccionando para esse efeito as expressões de maior contraste entre si que se encontrem dentro das possibilidades de exequibilidade de um utilizador normal.

Neste sentido apontaram-se as expressões a trabalhar nos capítulos subsequentes na medida em que implicitamente se definiram elementos expressivos nelas. Esses serão por sua vez trabalhados ao nível da classificação no capítulo 5.

Capítulo 3

3. Cadeiras de Rodas

3.1 Evolução

A ideia de uma cadeira que permita aos incapacitados mobilidade (autónoma ou assistida) é já bastante antiga. Consta-se que nos anos 500 depois de Cristo, já os chineses tinham aplicado rodas a cadeiras com o intuito de transportar enfermos.

Já no continente europeu, há registos de que em 1595 o rei Filipe II de Espanha, já se servia de uma cadeira de rodas empurrada pelos seus criados [6].

A primeira cadeira de rodas que assumiu a forma aproximada das actuais nasceu no século XVIII, pelas mãos de *John Dawson* em 1783, com rodas de madeira e um contrapeso na traseira [7]. O seu aspecto era bastante semelhante ao que se mostra na .



Figura 3.1 - Cadeira do séc. XVIII (adaptado de [10])

Na década de 80 (1881) tornou-se comum que as cadeiras tivessem nas rodas Jantes para permitirem ao doente empurrarem a cadeira pelos seus próprios meios. Em 1900, as rodas de madeira foram substituídas pelas ainda hoje populares rodas raiadas de metal. Desde então, e pela necessidade que adveio principalmente do resultado de diversas guerras, das quais a maior contribuinte foi a primeira guerra mundial, surgiram diversos modelos de cadeiras de estrutura rígida que foram incorporando elementos adicionais como o repouso para braços, repouso independente para os pés.

Só em 1932 *Herbert Everest* e *Harold Jennings* criaram a primeira cadeira de estrutura encartável resultando daí uma empresa (E&J) que ainda hoje tem o nome dos dois engenheiros.

Posteriormente surgiram outras patentes de cadeiras encartáveis, mas foi novamente a empresa acima citada que introduziu a primeira cadeira de rodas comercial motorizada, já nos inícios da década de 50, quando os seus autores aplicaram o controlo transistorizado de motores ao desenho da sua cadeira manual.

O desenvolvimento de cadeiras, desde então tomou diversas frentes consoante a tónica dos projectos em questão. Na óptica da ergonomia procurou-se uma melhor adaptação às deficiências do utilizador. Na vertente da propulsão, procuraram-se novas formas de minimizar o esforço que o utilizador tinha de realizar de forma a mover a cadeira, com a incorporação de sistemas mecânicos alternativos, como pedais (mas para as mãos) e uso de motores eléctricos (e mais tarde até de motores de combustão de pequena cilindrada, para cadeiras de locomoção exterior).

A cadeira de rodas passa a ser um acessório banal que leva à integração social de muitos fisicamente diminuídos. Em 1952 nascem os primeiros jogos de cadeiras de rodas em Inglaterra. Sendo que só doze anos mais tarde aparecem os primeiros jogos *Paralímpicos* em 1964 em Tóquio, no Japão [2].

Esta tendência de alargamento de actividades de elevada mobilidade a pessoas com deficiências motoras severas (nos membros inferiores), levou à necessidade de criação de cadeiras de materiais leves, facilitando por isso a locomoção por minimização do esforço físico necessário para as mover.



Figura 3.2 - International Symbol of Access

A banalização e aceitação da sociedade ante este tipo de acessório, já era de tal forma grande que em 1968, desenhado por *Susanne Koefoed*, nasce a imagem do *International Symbol of Access* (ISA)⁶ como se indica na Figura 3.2.

Com o aparecimento dos microcontroladores no início da década de 80 nascem as primeiras cadeiras de rodas cujos controlos se adaptam melhor às necessidades do utilizador.

Desde então, deu-se a revolução de desenho, de controlo, de autonomia, de conforto etc. . As imagens seguintes ilustram alguns desses produtos:

⁶ Imagem de copyright da *International Commission on Technology and Accessibility* (ICTA)



Figura 3.3 -Imagem de cadeiras motorizadas (adaptados de [23])



Figura 3.4 - Exemplos de cadeiras de mobilidade alargada (adaptadas de [25] e [34])

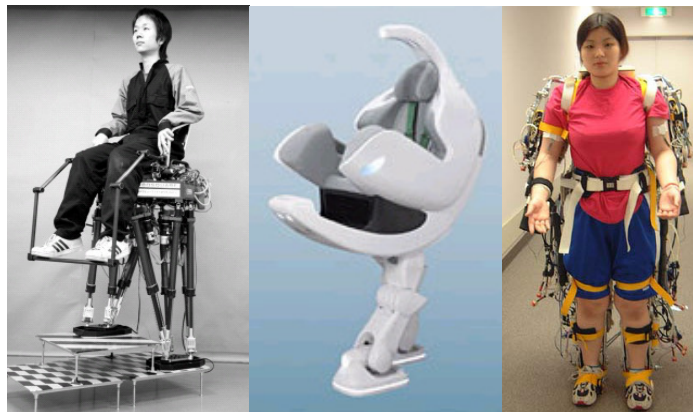


Figura 3.5 - Cadeiras bípedes e exoesqueletos (adapt. [19],[33] e [18])



Figura 3.6 - Cadeira comandada por interpretação de impulsos mentais (adaptado de [14])



Figura 3.7 - Alguns protótipos nacionais de cadeiras multi-sensoriais (adaptado de [38] e [4])

3.2 Projecto Intellwheels

À medida que se incorporam sistemas mais complexos de condução de cadeiras de rodas, com a utilização de autênticos robôs no campo do apoio aos incapacitados motores, o problema do interface é um dos mais importantes a estudar. Nesta vertente existem duas linhas de desenvolvimento essenciais.

A primeira prende-se com o aperfeiçoamento e criação de novos dispositivos de entrada de dados, que permitam comunicação mais eficaz com o utilizador. Neste contexto a inclusão de dispositivos capazes de recolher elementos biométricos do utilizador, é cada vez mais frequente.

A título de exemplo podemos mencionar *joysticks*⁷, microfones, câmaras e mais recentemente por leitura de impulsos nervosos.

⁷ Estes podem se apresentar como manetas convencionais dos jogos de computador ou adaptações complexas comandados por exemplo pela língua, pelo queixo, etc.. Essas dependem do grau de imobilização do utilizador.

A segunda vertente dos esforços de melhorar a interface com o utilizador é a de integrar vários dispositivos no controlo em simultâneo. É esse o objecto de trabalho do desenvolvimento de interfaces multimodais.

Intellwheels – Hardware

O projecto *Intellwheels*, onde se integra o presente trabalho, é um exemplo em que se aplica uma interface multimodal. O objectivo é o de alargar ao máximo a aplicabilidade da mesma cadeira a deficiências físicas de diferentes graus pela versatilidade de configuração que apresenta ao nível do controlo. A Figura 3.8, dá uma panorâmica geral do sistema.

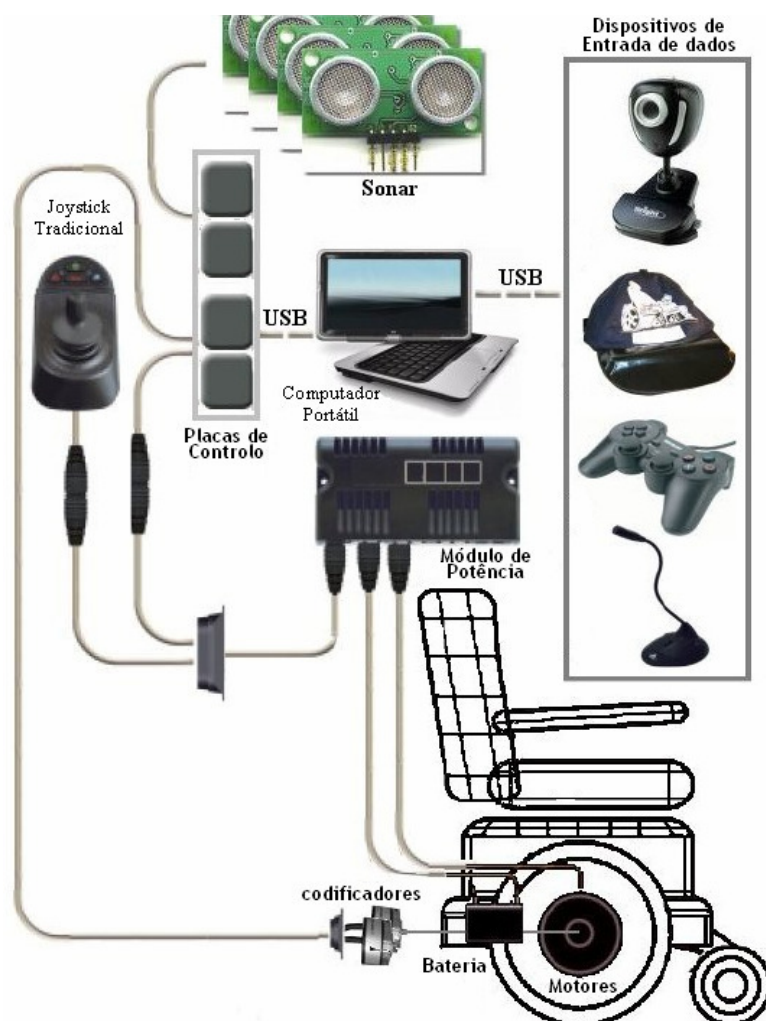


Figura 3.8 - Componentes do sistema Intellwheels (adaptado de [4])

No que respeita aos dispositivos de entrada, o objectivo do projecto, como uma cadeira multimodal, é o de numa fase posterior integrar qualquer tipo de dispositivo de entrada que consiga ser comandado por um utilizador e que produza saídas discretas que posteriormente poderão ser associadas a comandos da cadeira constituindo assim um sistema aberto.

Dos apresentados na figura anterior, temos o *Joystick* tradicional, microfone para comandos de voz, consola tipo *playstation*, um detector de movimento da cabeça e uma câmara de vídeo.

Intellwheels – Software

Na interface de controlo (*Control Interface*), nesta arquitectura (Figura 3.9), situa-se o controlo da cadeira de rodas e a interligação com o hardware.

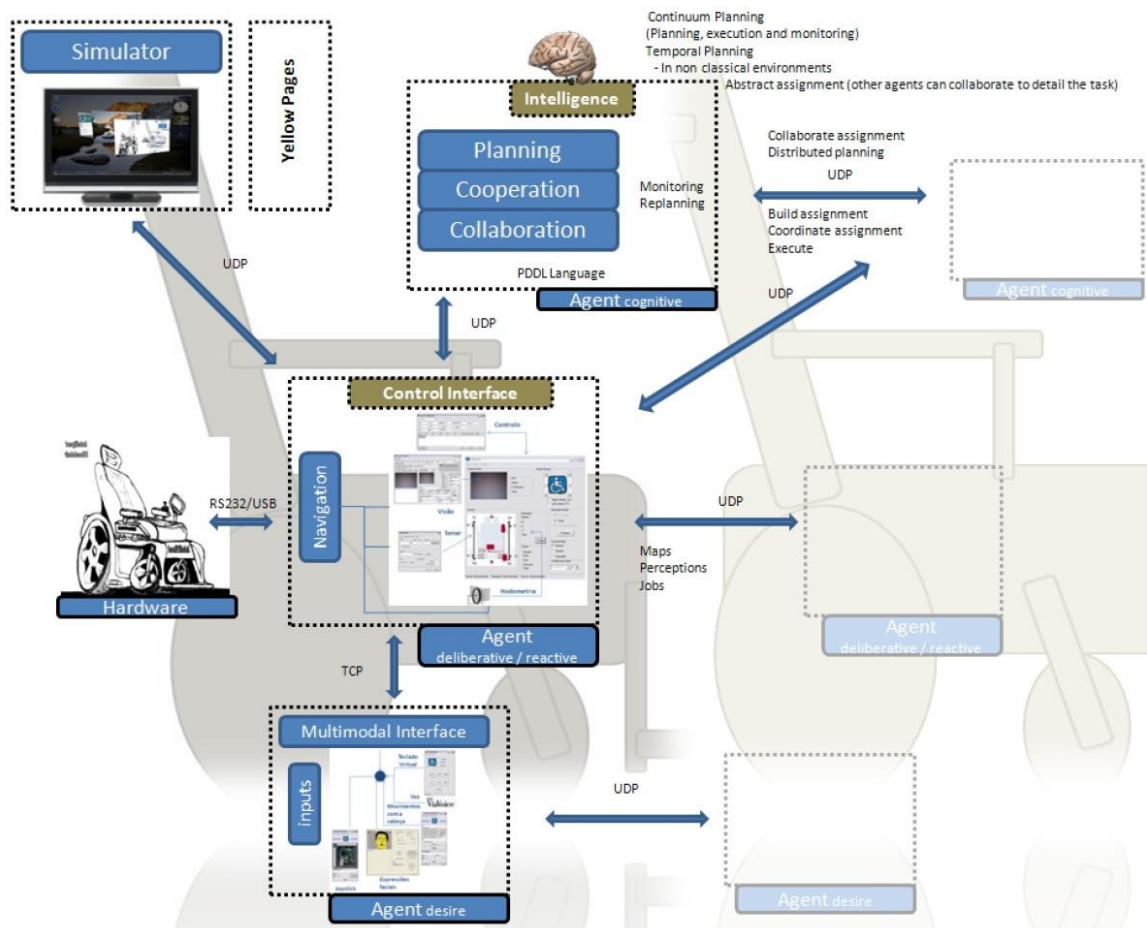


Figura 3.9 - Arquitectura do Projecto Intellwheels (adaptado de [4])

O simulador é utilizado para testar as rotinas de controlo da cadeira de rodas num ambiente simulado. Este módulo pode também ser utilizado para activar o modo de realidade virtual aumentada, que gera uma interacção real e virtual do mundo.

O módulo de inteligência (*Intelligence*) é o módulo de planeamento de alto nível. É aqui se realiza o planeamento para a conversão de acções de alto nível em comando directos da cadeira.

A interface *multimodal* interage com interface de controlo através de rede baseadas no protocolo TCP recebendo deste informações várias sobre acções em curso e o estado de planos em execução ou pendentes.

O utilizador por sua vez interage com a interface *multimodal* também através de conexões de rede baseadas no protocolo TCP que são fornecidas aos diversos de entrada que permitem a

codificação de acções através de sequências que os mesmos emanam e que são reconhecidas pela interface *multimodal*.

Interface multimodal

Esta é a interface do projecto para o utilizador. Serve para mostrar de forma gráfica informações sobre acções emanadas do módulo de controlo e dos módulos de entrada como o tipo e nome da acção ou da entrada de dados. Também mostra as sequências de acções definidas para controlo (Figura 3.10).

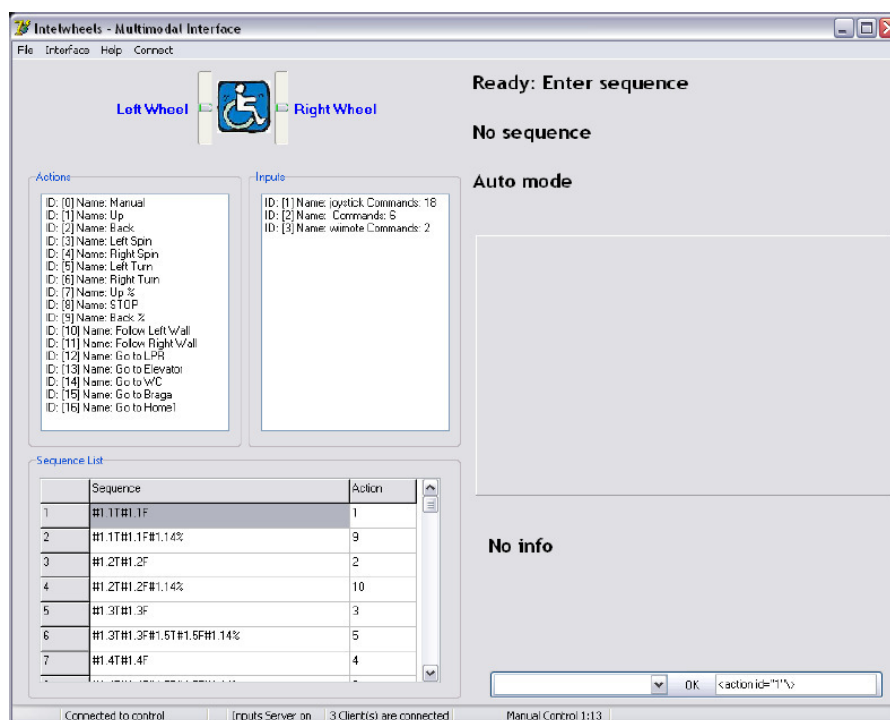


Figura 3.10 – Definição de sequências (adaptado de [21])

Cada dispositivo, por sua vez tem ecrãs de interface específicos onde se podem realizar acções de calibração e controlo específicas.

Como já se referiu anteriormente, os módulos de entrada existentes no momento são em número limitado.

Contudo o sistema é suficientemente aberto para incorporar outros que comuniquem com a interface de controlo de acordo com a sintaxe das mensagens definidas para o efeito.

Na prática, cada módulo pode funcionar como uma aplicação distinta, independente, com o seu interface próprio, que nem é obrigatório que esteja visível num dado momento para poder estabelecer comunicação com o módulo de interface de controlo.

A Figura 3.11 ilustra alguns desses interfaces específicos de módulos distintos de entrada.

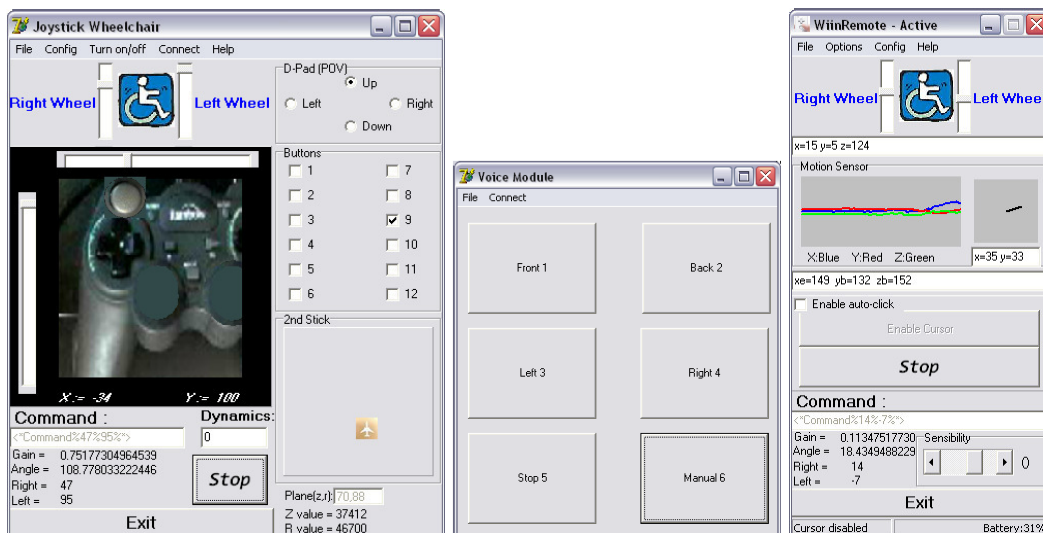


Figura 3.11 – Interfaces específicas (adaptado de [21])

Aplicação principal

O ecrã principal da aplicação resume informação vária como se mostra na Figura 3.12. De forma gráfica, organiza uma série de informações que compreende não só elementos provenientes dos módulos de entrada mas ainda informações úteis de estado sobre a cadeira, bem como ainda sobre o estado do software de controlo no que concerne ao modo de operação do mesmo.

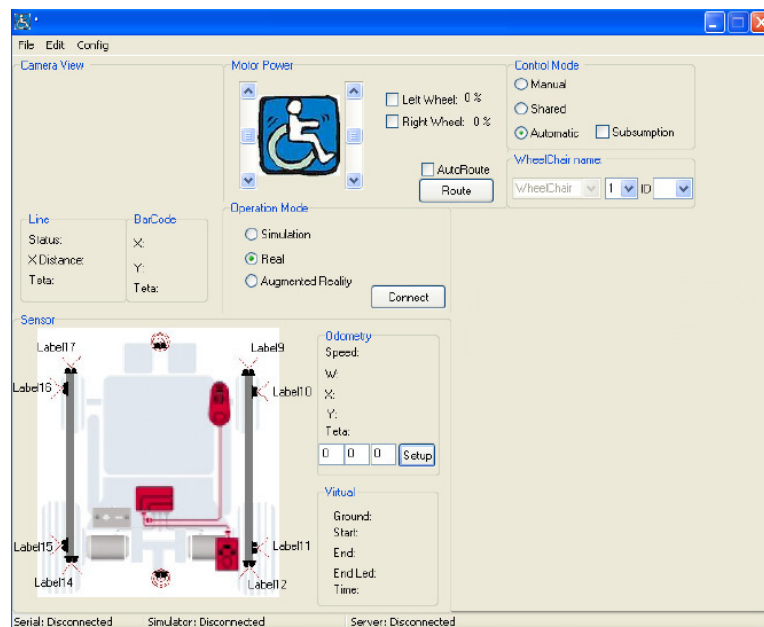


Figura 3.12 – Aplicação principal (adaptado de [21])

Este interface permite ainda configurar o tipo de funcionamento do sistema se real, simulada ou em realidade virtual aumentada.

O modo real, destina-se ao comando directo da cadeira de rodas. Já o modo simulado destina-se a correr testes sobre a cadeira em modo simulado, ou seja através do simulador gráfico da cadeira. O módulo de realidade virtual aumentada, permite comandar a cadeira num ambiente de realidade virtual com a introdução de elementos do mundo real.

Enquadramento do trabalho no projecto *Intellwheels*

Cada um dos módulos de entrada indicados tem as suas vantagens e desvantagens sendo por isso pouco apropriado indicar um como melhor que o outro pois o que conta é o que melhor se adapta ao utilizador que se vai servir da cadeira e não o que potencialmente oferece maior capacidade de controlo da cadeira.

Por outro lado as enfermidades que levam alguém a uma cadeira de rodas têm por vezes fases evolutivas de quadros clínicos que vão degradando a qualidade de vida do utilizador, pelo que nessas situações importa que a mesma cadeira possa adaptar-se a essas modificações sem implicar novos encargos (pelo menos ao nível do hardware/software). Também nesse sentido uma interface multimodal é uma vantagem evidente para este tipo de produto.

O enriquecimento do sistema com mecanismos de condução autónoma permite adicionar comandos de alto nível que aliviam o utilizador da tensão resultante da necessidade de controlar de forma contínua a cadeira. Comandos do tipo "contorna obstáculo", simplesmente podem libertar o utilizador de uma panóplia de manobras com os dispositivos de entrada de dados, que podem para um iniciado ser extremamente frustrantes. Se considerarmos que parte dos utilizadores alvo podem ter idades avançadas, é fácil compreender que a capacidade de aprenderem a controlar um novo dispositivo de forma eficiente não é o resultado mais provável a curto prazo.

O nosso trabalho, não sendo a implementação dum sistema *multimodal*, enquadra-se no mesmo, concretamente no subsistema de visão computacional, aperfeiçoando o reconhecimento de expressões faciais para posteriormente ser adicionado ao sistema *Intellwheels* através de um módulo de entrada.

3.3 Conclusões

Neste capítulo apresentou-se um resumo da evolução das cadeiras de rodas ao longo da história, salientando-se o especial interesse que este tipo de dispositivo representou (e representa) para a melhoria da qualidade de vida dos deficientes motores e a preocupação de desenvolver mais e melhores produtos ao alcance de todos.

Apontou-se ainda a temática do desenvolvimento de uma cadeira de rodas de alta tecnologia como um dos desafios que o sector privado, através de empresas de renome internacional, e o sector académico, encara como suficientemente e economicamente viável para se empenharem em investigação e desenvolvimento de novos produtos e tecnologias.

Apontaram-se exemplos de projectos de âmbito internacional e projectos de âmbito nacional. Nestes últimos expuseram-se detalhes do projecto *Intellwheels* onde se insere o presente trabalho de dissertação no sentido alargar as potencialidades do sistema multimodal através de um subsistema de visão de computador (aspectos que serão tratados no Capítulo 4 - Visão e Aprendizagem Computacional).

Capítulo 4

4. Visão e Aprendizagem Computacional

O principal objectivo dos sistemas de visão computacional (ou artificial) é o de conceber modelos de representação interna do mundo envolvente usando uma ou mais câmaras para captação de imagem.

Se por exemplo para um sistema de visão computacional, mesmo com avançado software de reconhecimento de imagem, é extremamente difícil distinguir entre a captura da imagem duma pessoa existente numa fotografia e da captura da imagem da mesma pessoa no mundo real. Trata-se da mesma dificuldade que se nos afigura, por exemplo se formos confrontados com o mesmo problema ao olharmos para as duas imagens através dum monitor.

Contudo se dispusermos de uma imagem em infravermelhos para a comparar com as capturadas pela câmara normal, de imediato distinguimos qual é a pessoa e qual é a fotografia da pessoa.

Contrariamente à visão humana ou animal em geral, não nos encontramos limitados a apenas um equipamento de recolha e podemos combinar sensores de naturezas distintas para melhor conseguir discernir o mundo que nos envolve. Não estamos limitados ao processamento de imagens na largura de banda do espectro visível pelo olho humano. Podemos variar essa banda de acordo com as nossas necessidades e com os dispositivos de captura de que dispusermos.

Outra das vantagens que estes sistemas dispõem é a de igualmente não estarem limitados ao ângulo de visão do ser humano o que permite abranger cenários mais alargados duma só vez.

No que respeita ao tratamento de profundidade, à semelhança dos seres humanos (e da generalidade dos animais) a utilização de duas câmaras distanciadas entre si o suficiente para necessitarem de diferentes focagens sobre o mesmo objecto, permite calcular com alguma precisão a distância a que o objecto se situa (sistema de visão estereoscópico). Alternativamente a combinação de espelho e câmara permite com algoritmos apropriados alcançar o mesmo efeito.

Usando as possibilidades de confecção de sistemas baseados em câmaras, existem projectos que usam câmaras para recolha de 360 graus. Esta recolha mais envolvente do mundo

nos sistemas de controlo das cadeiras de rodas (e de outros dispositivos), enquadra-se na visão estereoscópica permitindo um cálculo mais rigoroso das distâncias, desde que com software adequado.

No que respeita à condução autónoma de cadeiras, tal sistema tem a vantagem de permitir não só o cálculo de distâncias mais rigoroso, mas também o de permitir ao sistema de controlo tomar decisões informadas sobre todo o mundo envolvente sem ter de rodar a cadeira para o abranger. Um exemplo desses sistemas é o que se ilustra na Figura 4.1 [1].



Figura 4.1 Projecto do Instituto Nacional Japonês de Ciência e Tecnologia Industrial Avançada (adaptado de [1])

A utilização de sistemas mono câmara, apesar de serem limitados no que respeita às potencialidades acima referidas, continuam a ser capazes alimentar de forma qualitativa um sistema de visão computacional.

4.1 Fases da visão computacional

Duma forma genérica podemos dividir os sistemas de visão computacional em etapas ou subsistemas com actuações bem distintas e isoladas.

4.1.1 Captura de imagem

Todo o processamento de imagem tem de partir pela etapa da captura. Neste ponto a preocupação base está na configuração do equipamento que se está a utilizar de forma a que consiga recolher, com a melhor qualidade possível, imagens para esse sucessivo processamento.

A captura (ou aquisição) de imagens tem de ponderar questões como:

- Resolução de imagem – matriz de pontos (pixel) que deve ser tida em conta para apanhar os elementos que se pretendem processar;
- Taxa de captura (taxa de amostragem/*frame rate*) – número de imagens que devem ser recolhidas por segundo de forma a garantir um fluxo de imagens suficientemente constante para

não se perder a noção de movimento e suficientemente esparsa para não sobrecarregar em demasia o processamento dessas imagens. Esta taxa, deve ser ainda ajustada ao movimento do objecto que se pretende capturar. Para a maioria dos casos, uma taxa de recolha de uma webcam convencional, é suficiente para conseguir manter uma noção do movimento que decorre ante a câmara. Mas, por exemplo, para a filmagem de eventos desportivos, normalmente é necessário uma taxa de amostragem muito superior ao normal;

- Profundidade de cor – grandeza que define o grau de rigor com que se representa internamente a cor do mundo real e, por consequência, o resultado que se obtém na reprodução do que foi capturado. Normalmente a este nível define-se se se trata de uma captura a preto e branco ou a cores, e quantos bits são utilizados para representar internamente cada pixel na matriz interna da imagem capturada.

É na captura ainda que se resolvem problemas como focagem e compensação de luz, para de uma forma mais rigorosa conseguir alimentar outros sistemas. Esses podem ser tratados directamente na câmara por hardware ou software incorporado nela, ou à posteriori pelo software do sistema de visão.

4.1.2 Pré processamento

As imagens quando recolhidas são denominadas de imagens cruas. Estas são as que chegam ao sistema de visão após receber a informação da câmara. Aqui, para efeito do sistema, não importa se a câmara possui ou não software de controlo automático para melhoria da qualidade de imagem. A informação é tratada com se proviesse duma câmara simples.

Normalmente estas imagens vêm sobrecarregadas de informação, com detalhes que prejudicam de forma severa a detecção do essencial que se pretende recolher.

O pré processamento e o conjunto de técnicas que se podem utilizar para retirar esse excesso de informação e evidenciar os o que interessa ao processamento subsequente.

Alguns tipos de pré processamentos comuns são a redução de ruído e ajuste de contraste.

A redução de ruído pode ser alcançada com técnicas simples de filtragem (mais ou menos eficazes). No caso do processamento de imagens recolhidas por *Webcams* de baixa qualidade, o ruído pode ser de tal forma presente que chega ao ponto de distorcer por completo as cores na imagem capturada. Se a exposição de luz no momento de captura for insuficiente para a saturação das células de câmara, produz-se um ruído que faz lembrar grãos de areia atirados em frente à imagem. Isso é tanto mais comum quanto maior for a taxa de amostragem e quanto menor for a iluminação.

Ao nível do contraste o que se pretende é evidenciar as diferenças nos elementos que estão presentes na imagem de forma a facilitar a detecção das respectivas fronteiras ajudando assim a interpretação de cada cena.

No caso de as imagens serem recolhidas com excesso de ruído, ainda é possível combinar duas (ou mais) capturas sucessivas e por essa via eliminar significativamente esses ruídos parasitas que, por serem aleatórios, raramente coincidem nas duas imagens.

Contudo as acções de pré processamento mais frequentes são as de transformação de espaço.

4.1.2.1 Transformações de espaço

Qualquer imagem quando é recolhida é representada num espaço normalmente RGB⁸. Esse espaço tem eixos que para além da tradicional ideia do sistema de coordenadas XY (ou XYZ em alguns casos) contém outros eixos igualmente importantes, como por exemplo os da cor.

Na representação digital do espaço, pela própria natureza digital dos dados, existem limitações no que respeita à precisão com que se pode trabalhar a nível da representação numérica interna de dados. Ainda que existam computadores que trabalham com precisões muito para além do que humanamente é significativo, facto é que a partir dum determinado ponto toda a representação redonda em valores discretos.

No caso concreto da representação de imagens utilizamos sempre um espaço limitado, composto por eixos discretos, sobre os quais projectamos os pontos de imagem (pixel). A própria noção de pixel, como um ponto de cor, é um erro de concepção, pois contrariamente ao conceito matemático de ponto, um pixel representa sempre uma área (tipicamente mas não obrigatoriamente quadrada) e não um ponto.

O problema maior que se põe na fase de captura é saber de antemão qual o nível de precisão com que se deve trabalhar para obter bons resultados. Esse problema projecta-se tanto na determinação da resolução⁹ da imagem assim como na profundidade de cor¹⁰.

Análise de componentes principais

Existem técnicas próprias para determinar a melhor forma de representar uma imagem com o menor gasto de memória sem que se sofram perdas de qualidade excessivas. Uma dessas técnicas é a análise de componentes principais que define uma transformação linear do espaço de captura para um outro onde cada ponto se encontra definido num novo espaço com eixos que resultam de combinações lineares das coordenadas dos eixos originais. Esses eixos podem ser ordenados pela informação que representam, ao ponto de se poderem desprezar os eixos finais por conterem apenas informações residuais. Aplicando à imagem transformada (reduzida na sua

⁸ Sistema composto por três cores base vermelho (*R-Red*), verde (*G-Green*) e azul (*B-Blue*).

⁹ Número total de pixels da imagem.

¹⁰ Número de eixos de cor e número de cores diferentes por eixo

dimensão) o processo oposto reconvertendo-a para o espaço inicial, consegue-se eliminar algum do ruído inicial o que simplifica o processamento posterior [29].

4.1.2.2 Filtros

Duma forma simplista os filtros destinam-se a suavizar¹¹ um determinado sinal e podem utilizar-se com resultados bastante interessantes em telecomunicações.

Neste caso pode-se realizar uma adaptação de forma a aplicar a filtragem a um sinal de vídeo. Para um dado pixel seja é possível por análise da sua vizinhança, de acordo com a função de filtragem suavizar as variações excessivas que nele tenham ocorrido.

Como qualquer transformação, a filtragem numa imagem, elimina ruídos prejudiciais ao processamento desejado, mas também elimina alguns aspectos da imagem que pertencem aos elementos a serem tratados e que, como tal, não são ruído. Em todo o caso, o excesso de pormenores, numa forma geral apenas dificulta a capacidade dos sistemas de detectar o essencial no processamento de imagens.

A filtragem numa imagem serve-se dos pontos vizinhos para garantir uma transição mais suave, no sistema de cor utilizado, entre o pixel a suavizar e a sua vizinhança. Logo tal ideia leva a que seja necessário considerar o primeiro problema, do qual depende muito a capacidade de resposta do filtro. Esse problema é a determinação de qual a área da vizinhança de pontos que se deve considerar significativa para garantir uma filtragem adequada.

Neste aspecto, interessa que seja empregue uma área o mais pequena possível uma vez que o processamento aumenta exponencialmente em função do número de *pixels* contidos nessa área (na). Note-se que consoante se introduzam mais elementos no filtro, para um dado ponto, mais se dilui o verdadeiro valor do ponto, o que pode ser indesejado.

O segundo aspecto é o da figura geométrica que se deve utilizar para essa área de suavização. Normalmente, a área é simplesmente um quadrado (como na) o que se deve à relativa facilidade com que se pode construir o algoritmo correspondente. Contudo, em boa verdade, a área mais adequada para a suavização é um círculo. O motivo está no facto de a filtragem dever ser realizada tendo em conta a distância euclidiana para que todos os pontos de igual distância na vizinhança participem de igual forma na filtragem. Todos os que estejam para além duma determinada distância deverão por isso ser ignorados.

Ainda que algoritmicamente possa ser um pouco mais complexo, a selecção desses pontos, se a área for consideravelmente pequena, só traz vantagem ao processamento.

¹¹ Entenda-se por suavização o acto de eliminação de variações excessivas.

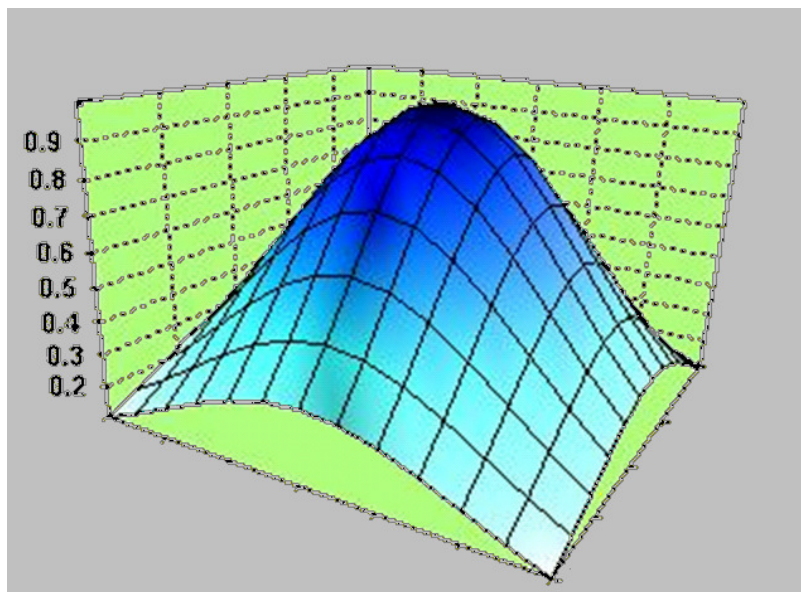


Figura 4.2 – Exemplo de um filtro Gaussiano

Na ilustra-se a diferença para uma janela de vizinhança de 5x5, e nela podemos ver a diferença entre seleccionar os pontos ou considerar a área por inteiro. Os pontos dos cantos, já se encontram para além da distância euclidiana limite, e como tal devem ser ignorados. A diferença para este caso é de 25 pontos para a área quadrada comparados com 21. Ainda que pareça pouco, esta diferença é aplicável a todos os pontos da imagem, e em cada segundo existe uma taxa de amostragem que, embora configurável, é significativa.

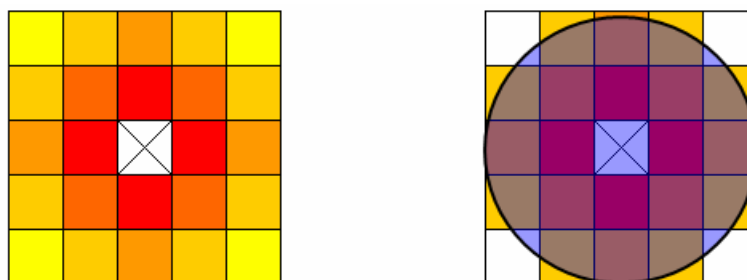


Figura 4.3 – Pontos de vizinhança

Se alargarmos a vizinhança cada vez mais, aproximamo-nos do rácio ideal, dado pelo quociente entre a área do círculo e o seu quadrado envolvente, como se indica na fórmula a seguir:

$$\frac{\pi \times \text{raio}^2}{(2 \times \text{raio})^2} = \frac{\pi}{4} \cong 0,7853... \quad (\text{Exp. 4.1})$$

Esse rácio é equivalente a 21% de ganho em processamento.

A última questão a colocar é o próprio filtro a empregar. No caso concreto do processamento de sinais tipicamente utiliza-se um filtro Gaussiano [31]. Este filtro consiste numa operação de convolução, suavizando a imagem digital com os pontos envolventes, removendo ruídos e detalhes excessivos.

A distribuição *Gaussiana* é isotrópica (circularmente simétrica) com média zero e desvio padrão σ pode ser aplicada a uma região bidimensional pela expressão seguinte:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (\text{Exp. 4.2})$$

Esta distribuição tem a particularidade de a distâncias superiores ao triplo do desvio padrão se aproximar já significativamente do zero, pelo que se podem desprezar os pontos daí em diante.

Desta forma quanto maior for o desvio padrão, maior será o efeito de convolução dos pontos envolventes do central. De igual forma menor será a diferença de impacto desses pontos vizinhos e do próprio ponto.

Opostamente, quanto menor for esse desvio, menor é o resultado do esbatimento, porque a função rapidamente assume valores próximos de zero na vizinhança do ponto a filtrar.

Desta forma é fácil regular o filtro.

4.1.3 Extracção de componentes

Uma vez preparada a imagem para análise o sistema de visão computacional necessita de recolher elementos para processamento.

Essa recolha pode ser feita com maior ou menor complexidade consoante a aplicação que tenhamos em mãos num dado momento.

Dos elementos que se podem extrair da imagem temos, linhas, contornos e sulcos;

A um nível de detalhe maior, conseguem-se evidenciar detalhes como por exemplo cantos, bolhas e pontos.

A um nível ainda superior, conseguem-se extrair ainda componentes da imagem como formas, texturas e (por comparação com imagens anteriores) movimentos.

4.1.4 Detecção, segmentação e processamento de Alto Nível

Uma vez conseguido o processamento para identificar elementos base da imagem, conseguem-se detectar as regiões que são o alvo do processamento final do sistema de visão.

Estas regiões podem conter elementos que embora possam ser reconhecidos por um sistema de visão computacional, mesmo assim não são os que se pretendem tratar.

Por exemplo se estivermos a construir um sistema de reconhecimento facial, interessa-nos essencialmente detectar onde se encontra a face que pretendemos submeter ao reconhecimento. Se a cena estiver saturada com objectos, nenhum deles deve ser assinalado pelo subsistema de visão uma vez que em nada contribuiriam para o cumprimento do objectivo principal, ainda que o sistema fosse perfeitamente capaz de o fazer.

Nesta fase a detecção dos elementos da cena permite a orientar a análise apenas para a zona que contém os elementos de interesse.

No processamento de alto nível, responsável pela identificação do que se está a "ver", que estará numa camada separada, usa-se a informação compilada pelas camadas anteriores e submete-se-lhe apenas a informação interessante ao reconhecimento do nosso objecto de estudo.

Tipicamente nesta fase existe algum método de inteligência artificial de suporte que toma decisões sobre esses dados finais.

Nesta fase tenta verificar-se se o que se está a "ver", corresponde aos modelos que o sistema possui dos objectos que interessam identificar. Ainda mediante estes modelos estimam-se parâmetros desses objectos tais como tamanho, distância e outros.

Esta fase do processamento pode ser tanto mais complicada quanto maior for o nível de integração de inteligência no sistema de visão computacional mas em última análise resume-se sempre a dados como detecção, categorização e localização espacial.

4.2 Aplicação de sistemas de visão computacional no controlo de cadeiras

À medida que evoluímos os sistemas de controlo de cadeiras para a implementação de condução autónoma, cada vez mais os sistemas de visão computacional passam a assumir uma importância crucial no conjunto de subsistemas existentes.

A razão é que os demais tipos de sensores, ainda que possam ser mais precisos na recolha de determinado tipo de informação, como por exemplo distâncias, tem limitações muito maiores que as da visão computacional.

Ou estão limitados pela própria natureza do dispositivo, ou pela posição do mesmo, ou pela velocidade de resposta, etc. ...

Por muito aperfeiçoado que seja um conjunto de sensores (não baseados em câmaras) é praticamente impossível recolher informação variada que permita modelar, tridimensionalmente de forma precisa, o mundo envolvente.

Com um sistema de visão computacional, e com algoritmos de reconhecimento de objectos apurados, consegue-se elevar a capacidade de consciência dos produtos no que respeita ao seu posicionamento contextualizado no mundo.

Informações como a forma dos objectos, cor, distância e até temperatura conseguem ser processadas em computador o que, juntamente a com bases de dados extensas sobre objectos do mundo real, permite criar com maior ou menor facilidade modelos cada vez mais completos do mundo exterior.

4.3 Aprendizagem computacional

Qualquer sistema de visão por computador só consegue ter alguma utilidade se dele se puderem tirar ilações para posteriormente as aplicar a um domínio em particular.

Essas informações têm de ser obtidas a partir de elementos tão abstractos como os pontos de cor que constituem uma imagem, expressos algumas vezes em sequências bem alargadas de bits. Com esse tipo de dados, o sistema de visão tem necessidade de obter informação através de algoritmos de análise da imagem.

É neste campo (entre outros) que a inteligência artificial, através da aprendizagem computacional, dá um contributo inequívoco a este tipo de sistemas.

A razão está na natureza dos problemas que os sistemas de visão têm de resolver. Ou seja trata-se de cenários onde nas entradas temos grandes volumes de dados em formato cru, e nas saídas temos tipicamente pequenos volumes de informação como por exemplo a classificação do que se está a "ver" num dado momento ou por exemplo a trajectória de um objecto conhecido.

Há ainda a questão de que o modelo que representa da melhor forma um determinado objecto ser uma tarefa que se pode entregar aos modelos criados pelas várias técnicas de aprendizagem computacional, sem que para isso tenhamos de investir tempo em modelações manuais.

Outro aspecto que é frequente ao nível dos algoritmos de processamento de imagem é o avultado peso computacional que está por detrás dos mesmos. O emprego de técnicas de aprendizagem computacional permite muitas vezes simplificar a complexidade computacional da aplicação prática em causa, na medida em que recorre a modelos matemáticos de resposta tipicamente linear em que das entradas para as saídas do sistema raramente existe a necessidade de encontrar soluções em espaços de procura de dimensão infinita. Esses modelos oferecem tempos de resposta, duma maneira geral, suficientemente pequenos para aplicações em tempo real [30].

Estas técnicas encontram-se divididas em categorias das quais as mais comuns são a *classificação* e a *regressão* [15].

Nos métodos de classificação o objectivo é a determinação duma classe para a rotulagem do estado do mundo num dado momento. Essas classes representam elementos tão simples como a presença de um determinado objecto ou propriedade. Num cenário de classificação mais complexo, poderá surgir a necessidade de determinar uma classe ante um leque de várias disponíveis mas sempre em número finito (e normalmente reduzido).

No caso da regressão é normal tentar-se detectar a posição de um objecto ou alguma outra propriedade cujo domínio seja contínuo (ainda que possa ser delimitado). Aqui a complexidade dos algoritmos geralmente é superior à da classificação.

No respeitante ainda às entradas que se submetem a estes dois tipos de métodos, a complexidade também varia sendo a menor encontrada quando as variáveis de entrada (também denominadas de variáveis independentes) se apresentarem com domínios discretos e de enumeração finita.

A complexidade maior, apresenta-se quando as variáveis independentes são contínuas.

4.3.1 Ferramentas de exploração

Longe vai o tempo em que para a análise de um conjunto de dados no sentido de os aplicar a um determinado método de aprendizagem computacional (ou outro qualquer), se recorria ao processamento manual dos dados com cálculos morosos e muitas vezes falhos.

De igual forma, também vai longe o tempo em que era necessário recorrer às capacidades de programação de um investigador, para que fosse possível por em prática num computador a exploração de um determinado algoritmo. Mesmo quando se implementa um algoritmo em que o controlo total sobre o mesmo é essencial, é fácil encontrar bibliotecas de rotinas que podem ser empregues nesse sentido com um mínimo de esforço de programação.

Correntemente são já muitos os exemplos de *software* que se encontram quer no mercado, quer sob a forma de *freeware* que permitem pegar num conjunto de dados e os processam ante as indicações do explorador através duma interface gráfica amigável.

A questão evoluiu de tal forma que, com a disseminação dessas ferramentas, nasceu o conceito de *data mining* (e por consequência *data mining tools*).

No domínio do *freeware*, encontramos dois exemplos bastante funcionais: o *WEKA* [35], o *R-Project* [28] com a sua mais recente interface gráfica *RATTLE* [36].

De entre as várias ferramentas de carácter *freeware* que se encontram online, a nossa escolha recaiu sobre o *RATTLE*, na medida em que é uma interface que se serve do vasto acervo de pacotes estatísticos e de *data mining*, que é o *R-Project*, e permite ao utilizador aproveitar os comandos que a *RATTLE* executa em forma de código-r para posterior aplicação e afinação.

4.3.1.1 RATTLE

A ferramenta *RATTLE* (*R Analytical Tool To Learn Easily*) é composta por um conjunto de ferramentas utilizado para tratamento de grandes quantidades de dados [36]. A ferramenta apresenta informações estatísticas e gráficas sobre os dados, realiza transformações sobre os dados em fase de pré-processamento para que possam ser modelados, constrói modelos tanto supervisionados como não supervisionados sobre os dados, apresenta o desempenho de forma gráfica e permite ainda a geração de previsões e novos conjuntos de dados.

Através da sua interface gráfica, o *Rattle* pode ser utilizado como ferramenta completa em projectos de data mining ou como ponto de partida para uma análise mais personalizada através da linguagem R.

Para além das potencialidades já referidas o *Rattle* é compatível com a maioria dos sistemas operativos standard.

Data (Definição de dados)

A primeira fase de qualquer projecto que se sirva da ferramenta é a definição dos dados.

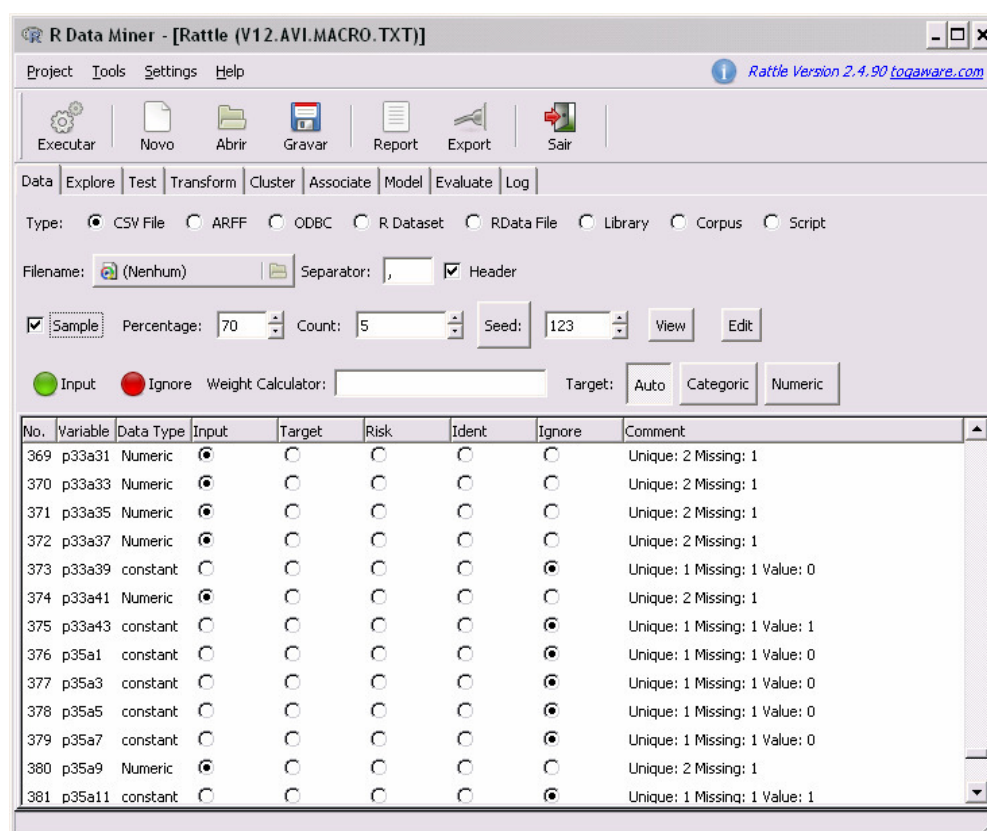


Figura 4.4 - Definição de dados (Data)

Nesta fase o objectivo é o de indicar à ferramenta o conjunto de dados a ser tratado bem como as propriedades desses dados. De entre essas propriedades a definição das variáveis independentes e dependentes é uma das tarefas mais importantes.

Para além dessa tarefa a definição do tipo de dados, quando esse não é implícito, é igualmente um ponto fulcral.

A ferramenta *Rattle*, permite neste ponto definir também a divisão desejada para o conjunto de dados de treino e de teste, que por defeito é de 70% e 30% respectivamente.

Caso se pretenda ainda faculta a possibilidade de utilizar um conjunto de dados para teste completamente distinto (mas tal realiza-se na fase de teste dos modelos).

Explore (Exploração)

Na pasta de exploração, é-nos possível aceder a um conjunto de estatísticas descritivas sobre os nossos dados (opção: *summary*), estudar as distribuições associadas a cada uma das variáveis (*distributions*), contemplar gráficos de distribuição (*visualize*), estudar a matriz de correlações entre variáveis (*correlation*) e realizar uma análise de componentes principais (*principal components*).

Test (teste de hipóteses)

Este menu tem por objectivo a realização de testes estatísticos sobre os dados para validação de hipóteses.

Nos testes de distribuições podemos realizar o teste não paramétrico como o de kolomogorov-smirnov bem como alguns testes de correlação como o de *pearson*.

Transform (transformação de espaço)

Esta opção tem por objectivo realizar transformações nos dados para os converter em dados normalizados. Permite a centragem dos dados, escalamento, ordenação, etc.

Cluster (agrupamento de dados)

Nesta opção é possível realizar agrupamentos (*clusters*) de dados através ou do algoritmo de *k-means* ou através de um agrupamento hierárquico. Possui ainda opção de desenho de dendogramas.

Associate (regras de associação)

A análise de associação identifica relacionamentos ou afinidades entre entidade e ou entre variáveis. Estes relacionamentos podem ser expressos por um conjunto de regras – regras de associação. Este tipo de análise foi particularmente bem sucedido na análise de regras de associação em compras de supermercado (tendo mesmo sido criado um formato de ficheiro próprio para o efeito – basket).

Model (criação de modelos)

É nesta opção que incide o nosso interesse. Aqui podem realizar-se a criação de modelos de classificação ou de regressão.

De entre os primeiros temos:

- *Tree* – árvores de classificação [26];
- *Boost* – ajuste de pesos conforme o acerto de vários algoritmos [37];
- *Forest* – árvores de classificação de acordo com o algoritmo *random forest* descrito por *Quinlan*.
- *SVM* – máquinas de vectores de suporte [8];
- *Linear Model* – discriminantes lineares;
- *Neural Net* - redes neuronais artificiais.

No segundo tipo temos:

- *Rpart* – (recursive partition) árvores de regressão;
- *Forest* – o mesmo que em classificação, mas aplicado à regressão;
- *Linear Model* – regressão linear;
- *Neural Net* - redes neuronais artificiais.

Infelizmente a ferramenta poderia ser um pouco mais abrangente no que respeita tanto à classificação como à regressão. No primeiro caso, não seria de todo excessivo que figurassem métodos como por exemplo o NaiveBayes (que tem excelente comportamento quando a variável objectivo é binária) ou por exemplo o K-Vizinhos mais Próximos [9]. No caso da regressão seria interessante por exemplo poder confrontar resultado com a regressão de Kernel (variação das máquinas de vectores de suporte aplicada à regressão).

4.3.2 Modelos de classificação

Como já foi referido anteriormente o objectivo dos modelos de classificação é o de determinar a classificação correcta mediante a apresentação de um conjunto de dados. Essa classificação obtém-se pelo processamento das variáveis de entradas que contém o referido conjunto de dados. O processamento gera um valor de saída que representa a classificação. As classificações podem ser de diversas naturezas sendo corrente encontrar classificações binárias. Também é frequente encontrarmos classificações nominais.

Consoante o tipo de classificação que se pretende realizar assim se escolhem os métodos de classificação adequados e, de entre eles, o que melhor desempenho apresentar. De seguida abordaremos em termos introdutórios os diferentes métodos oferecidos pela ferramenta *Rattle*.

4.3.2.1 Árvores de classificação

A ideia de qualquer modelo em árvore é a de hierarquização das decisões que levam ao objectivo final. Em termos das árvores de classificação, obviamente, o objectivo é o de tomar uma decisão sobre a classe a atribuir ao caso que se está a classificar.

Para cada conjunto de dados existem múltiplas configurações possíveis de árvores que se podem criar para a respectiva classificação.

A dificuldade consiste em determinar qual a árvore que melhor classifica os dados quer em termos de exactidão, quer em termos de redução de complexidade do processo de classificação. Tanto num caso como noutro, a diferença assenta nos critérios que se utilizam para a selecção do conjunto de variáveis que deverá ser testada em cada um dos nós da árvore para surtir o melhor efeito.

Um dos algoritmos de árvores mais utilizado é baseado no ID3 [26] e utiliza o índice da entropia¹³ para a selecção das variáveis de cada nó.

O RPART¹⁴, utilizado pela ferramenta *Rattle*, permite a parametrização do critério a ser utilizado para selecção do nó. Por defeito, é analisada a variável objectivo e em função dela é assumido o método mais provável pretendido. No caso de a variável ser categórica (caso em que o seu domínio em R é de factores), então é escolhida automaticamente a classificação como o objectivo do modelo.

No campo das variáveis independentes, é caso não sejam categóricas, é realizada uma discretização automática criando classes artificiais por intervalos de valores contínuos [13]. Quando os valores são inteiros, então esses intervalos são determinados em função no número total de valores que ocorrem para uma dada variável (*bucket size*) nos casos indicados para construção do modelo, de forma a equilibrar esses intervalos.

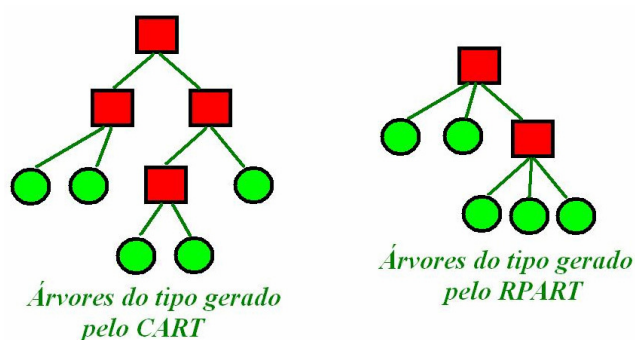


Figura 4.5 - Árvores binárias v.s. Árvores não binárias

¹³ Na verdade utilizam o critério do ganho de informação, baseado na diferença de entropias em cada nó entre a variável dependente e as diferentes variáveis elegíveis no nó, dentro dos remanescentes casos que incidem nesse nó.

¹⁴ Pacote do R – *Recursive Partitioning And Regression Trees*

Contrariamente a algoritmos como o CART (Figura 4.5), no RPART as árvores não são binárias, pelo que atingem níveis de profundidade dimensões menores salvo se a própria natureza das variáveis já for binária.

Para além dessa diferença, também não padecem tanto do problema do sobre-ajustamento aos dados.

4.3.2.2 Discriminantes lineares

No caso dos discriminantes lineares o que se pretende é traçar uma divisão no domínio da variável dependente de forma a discriminar entre duas classes existentes.

Como o processo de discriminação é baseado num critério separador, baseado numa função linear, existe a necessidade de discretizar o resultado final para poder estabelecer a correspondência entre a função linear e o valor da classe.

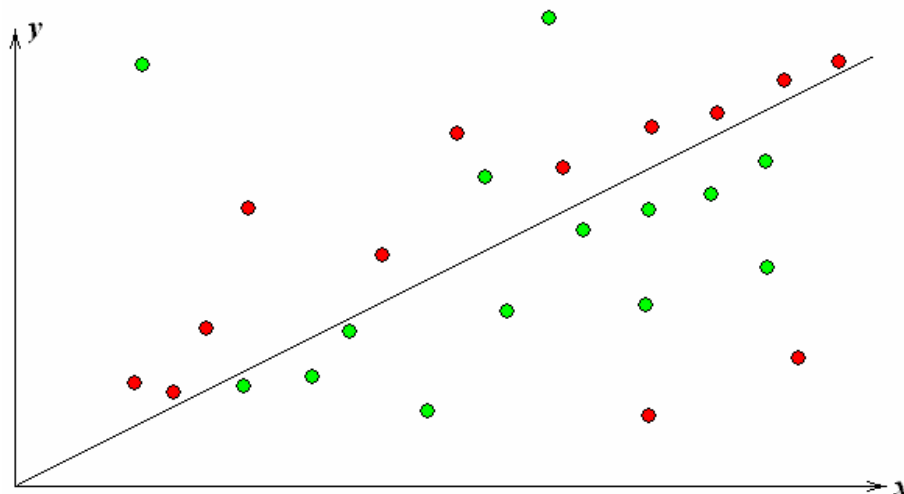


Figura 4.6 - Discriminante aplicado a duas classes

A vantagem dum discriminante linear (Figura 4.6) é a sua aparente simplicidade em termos de aplicação. Contudo, para dados não linearmente separáveis o erro de classificação pode ser bastante significativo.

4.3.2.3 Random forest

O algoritmo de *random forest* [5] baseia-se em baterias de árvores geradas com excertos aleatórios dos dados base. Cada árvore é gerada com um conjunto independente de dados. É definido um subconjunto aleatório de variáveis das quais se usa uma, aleatoriamente para o nó de raiz aplicando-se daí em diante o algoritmo convencional de construção da árvore.

Contrariamente a um modelo criado baseado numa única árvore, as árvores da floresta são criadas até ao limite de crescimento, sem poda, uma vez que pretendemos criar o máximo de árvores diferentes. A poda tem por objectivo caminhar rumo à generalização pelo que, se aplicada, a probabilidade de criar árvores com a mesma estrutura aumentaria [27].

No final, para efeito de classificação, cada uma das árvores da floresta gera uma classificação que é submetida a um processo de contagem. A maioria determina o valor da classe.

O erro generalizado da floresta converge para um limite à medida que o número de árvores na floresta aumenta. Depende também da robustez de cada uma das árvores individuais e respectivo grau de correlação entre elas.

4.3.2.4 Redes neuronais

A ideia das redes neuronais assenta na biologia recorrendo-se à imagem dum neurónio para criar uma modelação matemática [McCulloch and Pitts 1943].

Cada neurónio, é uma função linear que combina as diferentes variáveis de entrada para gerar um valor. Esse valor é submetido a uma função de saída que, consoante o objectivo da rede, pode tornar o valor discreto (entre zero ou um, para saídas binárias - função escada) ou um valor que varia num intervalo restrito (entre 0 e 1 : função sigmóide, ou entre -1 e 1 : função tangente hiperbólica).

$$\begin{bmatrix} v_1 \\ v_2 \\ \dots \\ v_k \end{bmatrix} \rightarrow f\left(\sum_{i=1}^k (w_i \times v_i)\right) \rightarrow y$$

Figura 4.7 - Neurónio artificial de McCulloch e Walter Pitts

O valor é calculado por combinação linear das variáveis de entrada v_i com os pesos w_i , sendo o resultado final submetido à função de saída.

Por uma questão de melhor ajustamento ao contradomínio, podem-se utilizar aproximações onde para além das variáveis de entrada se introduz uma fictícia (de valor sempre igual à unidade). Isto permite enviesar o valor de acordo com a natureza dos dados submetidos à rede.

As redes neuronais (artificiais), utilizam vários neurónios formando camadas. As saídas dos neurónios são passados às camadas seguintes como se fossem valores de variáveis de entrada.

Tipicamente as redes neuronais artificiais, apresentam uma camada intermédia de neurónios e uma camada de saída. Na saída normalmente apenas figura um único neurónio que com a função de saída final gera o valor pretendido. Contudo também se encontram aplicações onde se pretendem diferentes valores de saída em paralelo (mais de uma função) correspondendo cada valor a um neurónio de saída diferente.

Na apresenta-se um exemplo duma rede multicamada com camadas intermédias.

A vantagem das redes neuronais multicamada é a de poderem retalhar o espaço com múltiplo hiperplanos, de forma a poderem ajustar-se às múltiplas variâncias dos dados.

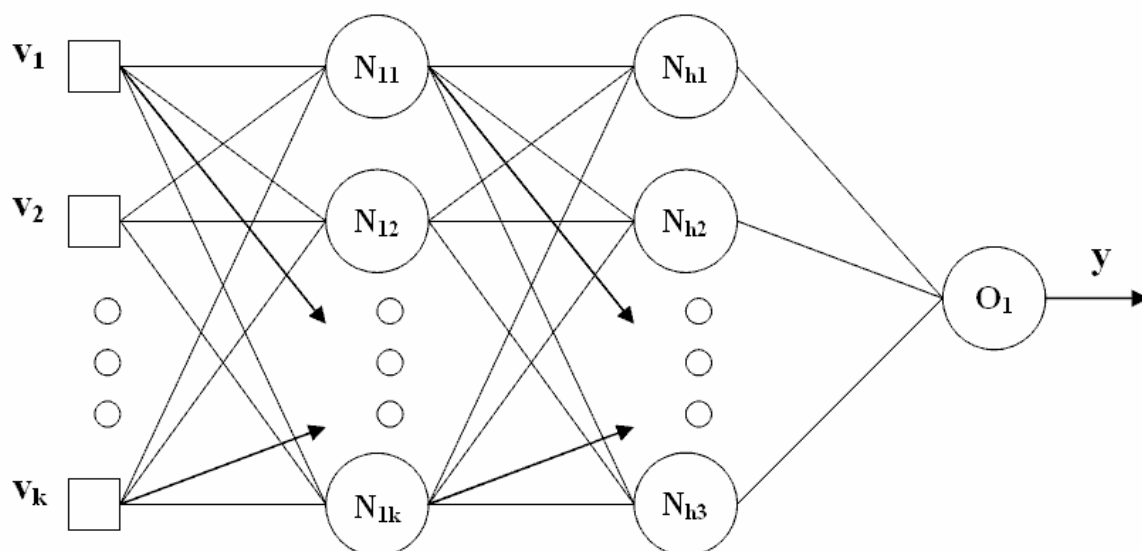


Figura 4.8 - Rede neuronal multicamada (adaptado de [30])

A grande desvantagem é que, para além terem de ser submetidas a um processo moroso de treino, as redes podem facilmente sobreajustar-se aos dados.

4.3.2.5 Máquinas de vectores de suporte

As máquinas de vectores de suporte servem-se da ideia dum discriminante linear aplicado. Realizam de igual forma uma separação do espaço em áreas onde seja possível identificar cada uma das classes, mas apresentando algumas diferenças.

A primeira é que a separação se realiza por maximização de uma fronteira separadora entre os conjuntos de pontos de cada classe - fronteira negativa () [8].

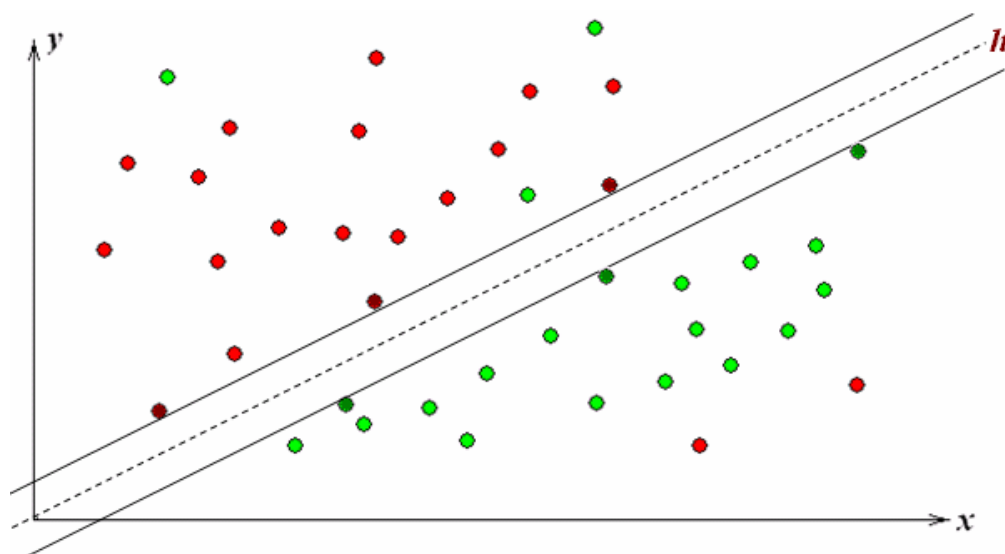


Figura 4.9 - Separação por maximização da fronteira

Para isso localizam-se os pontos (vectores de suporte) que permitem alargar essa fronteira.

A segunda diferença, é a de para casos onde os dados não são linearmente separáveis, utilizar uma transformação de espaço (através duma função kernel) para um outro onde os dados se encontram projectados de forma linearmente separável. É nesse espaço que se aplica a separação.

Como se ilustra () o hiperplano h representa o discriminante real dos diferentes pontos. Os vectores de suporte são os pontos assinalados de cor mais escura e definem onde passa a fronteira.

4.3.3 Avaliação de modelos de classificação

No que respeita à avaliação dos modelos de classificação é óbvio que o que interessa é aferir o grau de acerto nas diferentes classes.

Para cada caso, do conjunto de treino, temos um e só um valor para a classe¹⁵. O mesmo sucede para o conjunto de dados de teste¹⁶. No processo de teste do modelo, cria-se uma tabela com tantas colunas e linhas quanto as diferentes classes existentes nesses conjuntos de dados.

Corre-se o teste e confronta-se para cada valor da classe existente por cada caso do conjunto de teste o valor correspondente de classificação gerado pelo modelo. Procura-se nessa tabela a linha da primeira classe e nessa linha incrementa-se a coluna correspondente ao resultado obtido.

Esta tabela recebe a designação de matriz de confusão ($t_{i,j}$ - Tabela 4.1). Na diagonal principal, assinalada a sombreado, encontram-se as contagens de elementos correspondentes a uma classificação correcta. As restantes células, que neste caso correspondem à diagonal oposta, apresenta os erros de classificação.

¹⁵ Poderão ocorrer casos em que apareçam no conjunto de treino dois casos com os mesmos valores para as variáveis independentes mas com valores diferentes para a variável dependente (classes diferentes). Quando isso acontece, estamos perante ruído nos dados e importa eliminá-los uma vez que apenas lançam confusão para os modelos de classificação. Se o conjunto de dados tiver demasiadas ocorrências desta natureza, o que se detecta no pré-processamento dos dados, então o conjunto de dados deve ser recriado.

¹⁶ Dependendo do volume de dados, poderemos ter um conjunto de treino e de teste separado ou não. Duma forma geral, quando os dados são poucos, usa-se um ciclo de validações – validação cruzada. Neste caso, repetem-se as fases de treino e testes sobre fracções distintas dos dados em cada iteração e colectam-se os resultados dos vários testes.

Tabela 4.1 - Matriz de confusão: exemplo para duas classes

		Classificações obtidas	
		Classe A	Classe B
Valores reais	Classe A	$t_{1,1}$	$t_{1,2}$
	Classe B	$t_{2,1}$	$t_{2,2}$

Note-se que a matriz de confusão é sempre uma matriz quadrada e terá sempre tantas linha e colunas quantas as diferentes classes que existem nos dados.

A taxa de acerto é determinada pelo quociente do somatório da diagonal principal pelo somatório de todos os elementos da tabela.

$$tx_{acerto} = \frac{\sum_{i=1}^n t_{i,i}}{\sum_{i=1}^n \sum_{j=1}^n t_{i,j}} \quad (\text{Exp. 4.3})$$

(n – representa o número de classes diferentes, $t_{i,j}$ representa um elemento da matriz de confusão da linha i e coluna j)

4.3.4 Modelos de regressão

Nos modelos de regressão o objectivo é aproximar com o maior grau de exactidão possível o valor da variável objectivo.

No respeitante a este tipo de problemas o pré-processamento poderá assumir importância crucial face aos resultados que se obtêm na medida em que na maioria dos casos se utilizam as próprias variáveis de entrada no cálculo do valor estimativa. Nesse sentido, os métodos são mais sensíveis aos dados que para o caso da classificação. Relembremos que neste último caso a noção de distância entre classes se cinge ao um critério binário de identidade. Ou a classe é igual ou é diferente, não sobrando espaço para graduações intermédias. No caso da regressão, por oposição, o grau de acerto pode ser medido por funções de distância o que potencia os efeitos nos resultados face a variações pequenas nos dados de entrada.

Não é de todo invulgar encontrarmos na fase de pré-processamento um passo de normalização estatística de dados antes da aplicação dos métodos de regressão. Noutros casos, esse passo já é parte integrante dos próprios métodos.

Por último convém ainda lembrar que mediante a natureza do método empregue poderá ser ou não necessário redimensionar o contradomínio de forma a adaptá-lo à sua natureza original, correspondendo esse passo ao processo inverso da normalização estatística.

4.3.4.1 – Regressão linear

Nos modelos de regressão linear tenta modelar-se a variável dependente procurando uma relação linear entre as variáveis independentes [16].

$$Y' = \alpha + \beta X \quad (\text{Exp. 4.4})$$

Aqui, Y' representa os valores estimados, α representa a constante de desfasamento da recta face à origem, β o vector de parâmetros um por cada uma das variáveis dependentes e X a tabela de valores em que cada coluna corresponde aos valores das diferentes variáveis dependentes e cada linha correspondente a um caso do conjunto de dados.

$$Y = Y' + \varepsilon \quad (\text{Exp. 4.5})$$

Sendo que Y representa o vector de valores objectivo (variável dependente), ε representa o vector de erro.

O problema principal reside no cálculo do vector de parâmetros β de forma a minimizar o erro ε . Neste sentido existem diferentes abordagens em aspectos como a complexidade algorítmica ou a robustez face aos dados. Não sendo nosso objectivo abordar estas variações, para discutir virtudes e fraquezas, referimos apenas que dois dos métodos mais frequentes são o OLS¹⁷ e o GLS¹⁸.

4.3.4.2 - Árvores de regressão

Neste caso o valor da função objectivo é dividido intervalos e cada nó corresponde a um valor (normalmente o valor do meio de cada intervalo).

O processo de criação da árvore segue as mesmas linhas que para a classificação.

Dado que os valores são determinados em função dos pontos correspondentes a cada segmento são geradas escadas de valores em vez de valores contínuos. Contudo, se o numero de intervalos for suficientemente grande, o erro de aproximação pode tornar-se suficientemente pequeno para não ser significativo.

4.3.4.3 - Redes neuronais de regressão

Segue o mesmo procedimento que para o caso das redes neuronais de classificação, mas não usa funções de saída do tipo escada. No final, é necessário realizar uma desnormalização do valor de saída para o reajustar ao valor original dos dados.

¹⁷ Ordinary Least Square

¹⁸ Generalized Least Square (adaptação do OLS)

4.3.5 Avaliação de modelos de regressão

Como já foi anteriormente referido, a avaliação de um modelo requer uma medida de distância a ser computada para todo o conjunto de dados testados. Utilizando essas medidas de distância calculam-se as métricas que servem para posteriormente avaliar o desempenho de cada método e as métricas mais comuns são *mae* - *mean absolute error*, *mse* - *mean square errors*, *nmae* - *normalized mean absolute error*, *nmse* - *normalized mean square error* e a correlação.

Para efeito de exposição dessas métricas consideremos dois vectores Y e Y' que representam as sequências de valores observados e estimados (respectivamente) para os n casos que temos em estudo.

MAE – Erro médio absoluto

$$mae = \frac{\sum_{i=1}^n |Y_i - Y'_i|}{n} \quad (\text{Exp. 4.6})$$

Esta métrica não tem em linha de conta o sinal da diferença e acumula de forma igual todas as distâncias. Trata-se de uma métrica que não tem em linha de conta qualquer aspecto relacionado com a própria distribuição dos elementos.

MSE – Erro quadrado médio

$$mse = \frac{\sum_{i=1}^n (Y_i - Y'_i)^2}{n} \quad (\text{Exp. 4.7})$$

A única diferença face ao *mae* é que o efeito dos erros é potenciado pela sua gravidade. Assim erros superiores à unidade tem um contributo maior para o erro médio final e os inferiores à unidade tornam-se menos expressivos.

NMAE – Erro absoluto médio normalizado

$$nmae = \frac{\sum_{i=1}^n |Y_i - Y'_i|}{\sum_{i=1}^n |Y_i - \bar{Y}|} \quad (\text{Exp. 4.8})$$

Esta métrica tem em linha de conta a própria distribuição e é ajustada de acordo com o afastamento que a série de dados tem face à respectiva média. Isto permite que o valor resultante seja expresso na forma de um rácio que já é comparável com outras sequências de dados. No caso do *mae* tal já não é possível.

NMSE – Erro quadrado médio normalizado

$$nmse = \frac{\sum_{i=1}^n (Y_i - Y'_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} \quad (\text{Exp. 4.9})$$

À semelhança do *nmae*, esta métrica já se apresenta em forma comparável com outras sequências de dados, na medida em que tem em linha de conta o factor de normalização que é o somatório do erro quadrado da diferença dos próprios valores com a média.

Correlação

$$corr = \frac{\text{cov}(Y, Y')}{\sigma_Y \times \sigma_{Y'}} \quad (\text{Exp. 4.10})$$

Uma forma simples de exprimir a correlação poderá ser a indicada na expressão anterior, através do quociente entre a covariância entre os dois vectores e o produto do desvio padrão de cada um deles.

A correlação afere o grau de similaridade das duas sequências, sendo positiva quando o comportamento das duas sequências é similar e negativo quando são tem um comportamento simétrico. Será ainda próximo do zero quando não houver qualquer correlação e próximo da unidade (em valor absoluto) quando houver uma correlação total (positiva ou inversa se negativa).

NOTA: Esta métrica é interessante desde que as duas sequências tenham uma média equiparável. Mas pode ser bastante enganosa se os dois vectores se encontrarem em todos os seus valores desfasados em valores constantes. A correlação de uma sequência de valores Y com outra Y' em que Y'=Y+k (sendo K uma constante) é de 100%. No entanto o erro de aproximação entre Y e Y' é igual a k.

Métricas adoptadas

Em termos das métricas de erro adoptadas neste trabalho elas são o *nmse* e a *correlação*.

Com a primeira aferimos o erro total, em termos de valor, encontrado entre os dois vectores.

Com a segunda métrica, aferimos o comportamento da sequência de dados dos dois vectores (independentemente do valor absoluto).

Nem sempre ao longo do trabalho se faz necessário recorrer às duas métricas. Nesses casos onde a sensibilidade ao erro em termos de valor é menor, face ao problema em foco, utilizamos apenas a *correlação*. Quando o desvio em termos de valor é mais importante utilizamos apenas o *nmse*.

4.4 Conclusões

Neste capítulo abordaram-se aspectos relativos à visão computacional apontando-se as vantagens dos sistemas apoiados em múltiplas câmaras, dando-se mesmo um exemplo de uma câmara omnidireccional, e referindo-se a aplicabilidade de sistemas monocâmara.

Apresentaram-se diferentes fases dum sistema de visão computacional expôndo nelas os detalhes necessários ao nível da implementação dum sistema deste tipo e articulou-se esta temática com a necessidade de utilização de técnicas de apoio baseadas em inteligência artificial, em concreto a aprendizagem computacional.

Indicaram-se ferramentas que podem ser utilizadas para efeito de tratamento de dados na criação de modelos de suporte ao reconhecimento dos elementos recolhidos e previamente tratados pela visão computacional e apontou-se a ferramenta eleita para efeito de desenvolvimento desses modelos descrevendo nela as principais funcionalidades.

Expuseram-se os diferentes tipos de modelos que podem ser trabalhados com a ferramenta escolhida (*Rattle*) e indicou-se um breve resumo sobre as principais características deles.

Abordaram-se métricas de avaliação da qualidade dos modelos gerados pela ferramenta e definiram-se as convenções a adoptar no Capítulo 5 (O Sistema proposto) nessa avaliação, onde se realiza a exploração de dados de sustentação ao sistema de visão.

Capítulo 5

5. O Sistema proposto

Neste capítulo vamos abordar a proposta para construção do sistema de visão em termos dos diferentes componentes necessários e da respectiva interligação e enquadramento na arquitectura global do sistema.

Inicia-se a temática com a apresentação de pressupostos base sobre os quais se edifica o conceito de máscara. Apresenta-se a forma como se trabalha a máscara, através da sua normalização, e como a partir dela se obtém as diferentes áreas de interesse na imagem recolhida. Introduce-se o conceito de macro-visão e esclarece-se como se recolhem os conjuntos de dados retirados dessas áreas na fase de treino com recurso a marcadores.

Ilustra-se a forma como se utilizam os dados para criar modelos de regressão que estimam a posição dos marcadores de forma a sobre eles se ajustar a máscara do utilizador. Apresenta-se a forma como se calculam os ângulos de rotação da máscara em função dos marcadores e como em função dos mesmo se consegue aceder às áreas de interesse normalizadas.

Indica-se de que forma se pode utilizar o rasto de movimento do utilizador para implementar *reality checks* capazes de avaliar os valores obtidos pelas estimativas.

Apresenta-se o assistente de classificação que aplicado aos dados recolhidos permite criar os conjuntos de dados independentes para cada elemento expressivo e para esses conjuntos aplicam-se os diferentes métodos de classificação da ferramenta *Rattle* com a discussão dos resultados obtidos.

Utilizando os resultados das classificações, faz-se ainda a exploração duma mini-linguagem (ver ponto 2.4) no sentido de aferir o nível de confiança atribuível à mesma.

5.1 Pressupostos

O sistema de visão é específico para aplicação a utilizadores de mobilidade reduzida.

Este objectivo base, leva a que se possam (e devam) ter em consideração alguns pressupostos simples que permitem orientar o processamento às especificidades do utilizador,

libertando esse processamento do ónus computacional de determinar essas especificidades e de agir sobre elas.

Note-se que, como já foi referido anteriormente, a necessidade de uma resposta base do sistema é um dos requisitos do projecto.

Relativa imutabilidade

Por muito rica que seja a variância da disposição dos elementos expressivos do rosto humano, uma vez apresentados ao sistema, eles encontram-se localizados sempre aproximadamente no mesmo local da face. Podem apresentar-se ligeiramente deslocadas da sua posição de repouso consoante a expressão ou apresentar-se em perspectivas diferentes, mas o modelo é bastante imutável:

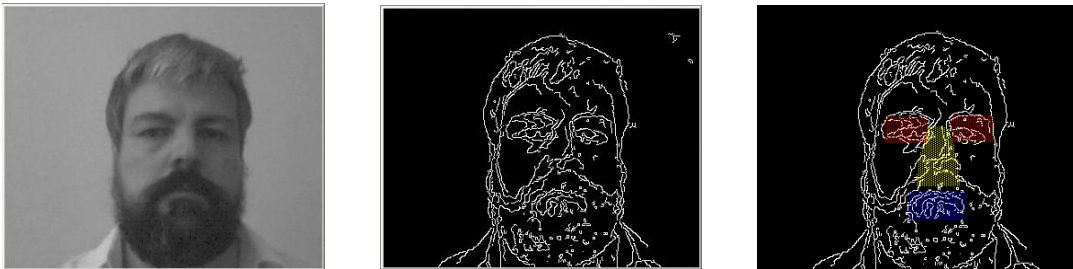


Figura 5.1 – Imutabilidade

Duma forma geral podemos desconhecer os pormenores associados ao utilizador tais como sexo, idade e mais importante, pigmentação da pele. Não importam essas variações, é de se esperar que posicionalmente a disposição dos elementos expressivos seja similar a todos os outros utilizadores, como se ilustra na Figura 5.1¹⁹.

Podem existir variações ligeiras nesse posicionamento (de cada elemento expressivo) consoante o utilizador, mas, é sempre de se esperar que os elementos se encontrem entre si em posicionamentos relativos similares²⁰.

Mobilidade relativa

¹⁹ Na figura foram demarcadas algumas áreas manualmente apenas a título ilustrativo do posicionamento dos olhos - a vermelho, do nariz - a amarelo e da boca - a azul;

²⁰ Apenas fazendo o pino seria de se esperar, por exemplo, que os olhos se encontrassem acima das sobrancelhas e estas abaixo da boca, e isso aplica-se a qualquer utilizador que, morfológicamente, não seja uma aberração (para o qual se teria de proceder a um treino específico do sistema);

Uma vez identificada a posição dos olhos, sobrancelhas, entre outros elementos, dado um movimento da cabeça, conseguimos, por projecção desses elementos, de forma proporcional, detectar aproximadamente a posição onde se encontram (dependendo do lapso de tempo que separa o reconhecimento e essa projecção).

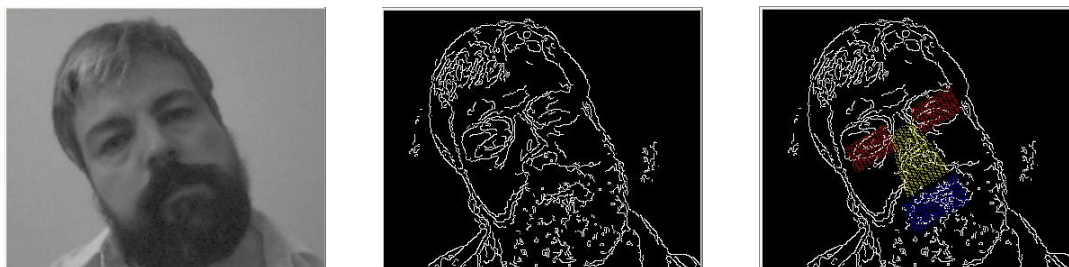


Figura 5.2 – Posicionamento relativo dos elementos a analisar

Este pressuposto permite que, através de transformações lineares relativamente simples, dada uma máscara de localização e dados dos elementos de movimento do utilizador e da última posição conhecida, seja possível determinar onde se deve esperar o posicionamento desses elementos na nova imagem para análise. Essa transformação ilustra-se na Figura 5.2 por comparação com a Figura 5.1.

Continuidade do movimento

Num mundo perfeito, a recolha de imagens seria contínua, uma emanção representativa de movimentos fluidos e constantes. A realidade contudo é bem diversa. As imagens recolhidas apresentam-nos movimentos discretos do utilizador, como se saltasse no tempo de uma posição para a seguinte.

Esses saltos, dependem sobretudo da taxa de amostragem da câmara que se está a usar num dado momento. Dependem ainda da capacidade de processamento do sistema que está a analisar essas imagens. Duma forma resumida dependem qualidade do hardware e do software utilizado.

Sucedem que duma forma geral essa descontinuidade não chega a ser demasiado alargada ao ponto de se deixar de perceber o movimento descrito pelo utilizador.

A máscara

Outro pressuposto é o da existência de uma máscara qualquer que seja o utilizador, o que para a extracção dos elementos expressivos é um pressuposto interessante.

Ao longo dos tempos foram diversos os estudiosos que se dedicaram à observação do ser humano no que respeita às suas proporções. Na verdade tal estudo, assumiu em alguns casos a característica de obsessão transformando-se na procura das dimensões e relações divinas no corpo humano. Noutros tratou-se de uma procura do ideal de beleza do homem. Tanto num caso

como noutro tais esforços, pelos objectivos que se propunham alcançar, foram coroados de fracassos inquestionáveis. Nem no domínio do divino foram aceites pela comunidade eclesiástica nem no domínio da beleza foram aceites para comunidade correspondente, havendo sempre opiniões acutilantes que contrariaram a validade dos resultados.

Contudo, nem por isso deixaram de ser trabalhos que recolheram métricas importantes sobre o corpo humano que ainda hoje são tidas por base em estudos e tratados de arte sobre o tema. De entre os vários nomes associados a esses esforços um dos mais conhecidos foi *Marcus Vitruvius Pollio*, arquitecto que criou textos com a descrição dessas proporções bem como ilustrações demonstrativas das mesmas. Infelizmente na época a divulgação das obras efectuava-se por cópia manuscrita, cuja qualidade estava na directa proporção da arte dos copistas, pelo que com o passar do tempo as ilustrações perderam-se.

Leonardo Da Vinci, nas suas deambulações sobre o tema, recriou essas ilustrações através do texto original concebendo a obra tão conhecida por Homem Vitruviano, Figura 5.3.

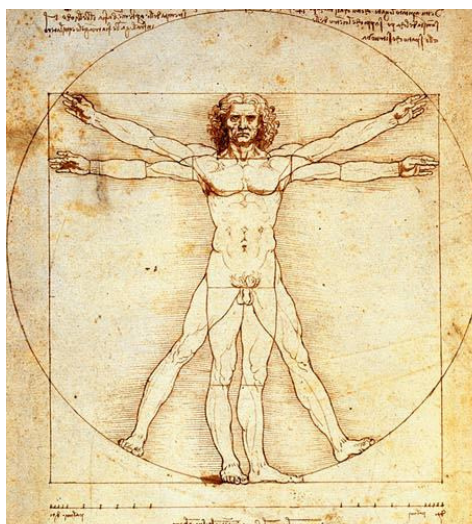


Figura 5.3 – Homem Vitruviano (ilustração de Leonardo da Vinci)

Nesta obra, Leonardo qual conduz os esforços de Pollio mais além, levando a bom porto as tentativas anteriores de Marco, de enquadrar o homem num círculo e num quadrado, tendo determinado algumas curiosidades sobre a proporção humana:

- Um palmo é a largura de quatro dedos;
- Um pé é a largura de quatro palmos;
- Um antebraço ou cúbito é a largura de seis palmos;
- A altura de um homem é quatro antebraços (24 palmos);
- Um passo é quatro antebraços;
- A longitude dos braços estendidos de um homem é igual à altura dele;
- A distância entre o nascimento do cabelo e o queixo é um décimo da altura de um homem ($d1$);
- A distância do topo da cabeça para o fundo do queixo é um oitavo da altura de um homem ($d2$);
- A distância do nascimento do cabelo para o topo do peito é um sétimo da altura de um homem;
- A distância do topo da cabeça para os mamilos é um quarto da altura de um homem;
- A largura máxima dos ombros é um quarto da altura de um homem;

- A distância do cotovelo para o fim da mão é um quinto da altura de um homem;
- A distância do cotovelo para a axila é um oitavo da altura de um homem;
- A longitude da mão é um décimo da altura de um homem;
- A distância do fundo do queixo para o nariz é um terço da longitude da face ($d3$);
- A distância do nascimento do cabelo para as sobrancelhas é um terço da longitude da face ($d4$);
- A altura da orelha é um terço da longitude da face.

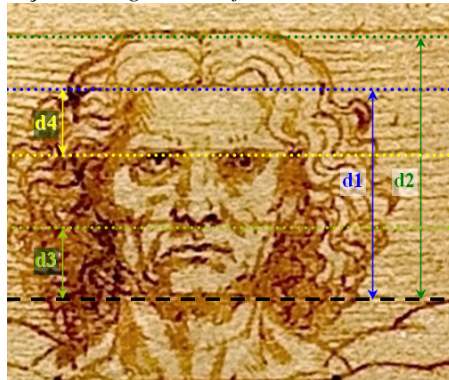


Figura 5.4 – Homem Vitruviano – proporções do rosto

No nosso caso apenas nos interessam as referências às proporções da face humana como se indica na Figura 5.4.

Sobre estes elementos basta definir na horizontal a área onde medianamente se enquadram os olhos do utilizador. Note-se que a própria forma da cabeça, mais ou menos larga em relação a altura da face permite inferir o posicionamento horizontal dos olhos e da boca. Relativamente às sobrancelhas estas acompanham (horizontalmente) os olhos dispondo-se para além dos limites dos mesmos.

Estas métricas dado que são relativas ao próprio ser, são ajustadas pelo sistema em função do modelo utilizado num dado momento.

5.2 Normalização da máscara

A análise constante de cada *frame*, partindo sempre da estaca zero, decorrente duma captação constante, é um processo pesado para o sistema, como se referiu anteriormente.

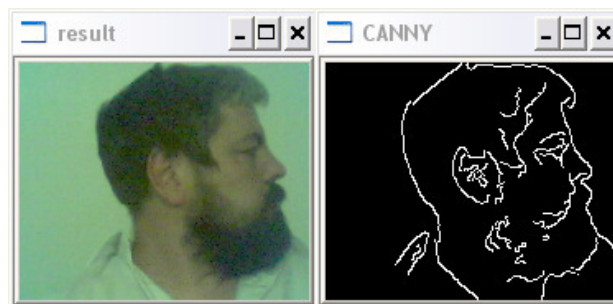


Figura 5.5 – Ocultação de pormenores

Uma das grandes dificuldades é a da mobilidade do utilizador em frente à câmara que leva a deslocação dos elementos expressivos na imagem que se está a analisar, mesmo quando não há variação na expressão que o utilizador está a realizar.

Por outro lado, não só existe uma deslocação na vertical e horizontal, mas ainda em profundidade, podendo determinados pormenores ficarem escondidos do ângulo de captura da câmara num dado momento, como se ilustra na Figura 5.5.

Ao se observar a imagem, imediatamente sabemos que não é possível ver o olho esquerdo do utilizador porque a cabeça foi rodada para a esquerda. Pelo que de nada adianta encetar mecanismos para reconhecimento daquilo que o utilizador está a fazer com esse olho (por exemplo para ver se o está a piscar).

Mais grave ainda é que o desencadear desses mecanismos, por falta de dados de suporte para que possa funcionar correctamente poderá levar a tirar conclusões erradas que apenas serviriam para lançar o caos no sistema de visão.

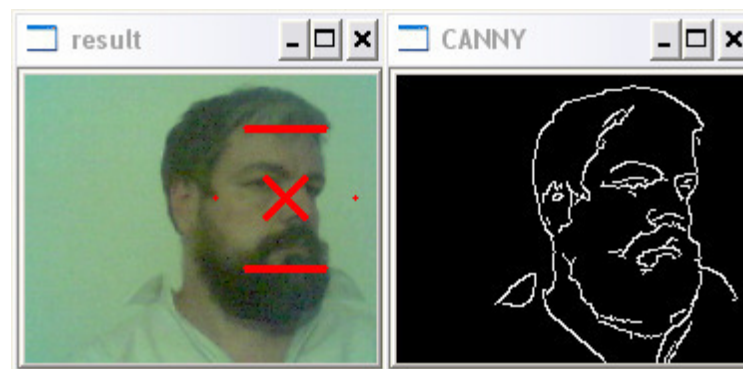


Figura 5.6 - Perspectiva

Já na Figura 5.6, ainda que os elementos expressivos não estejam totalmente visíveis, já é possível aplica-los a um classificador. Do ponto de vista prático, para um classificador é indiferente se a perspectiva é a melhor ou não, desde que se consigam extrair dados reais sobre o elemento que se está a tentar classificar.

Contudo, desde que a distorção resultante da perspectiva não seja extrema (como por exemplo o caso de ocultar por completo um elemento), poderá realizar-se um esforço para normalizar a área do elemento expressivo para sobre essa área lançar os classificadores, obtendo assim uma menor variância dos dados originais e tirando maior partido dos dados independentemente da perspectiva.

Considerando um sistema de três eixos convencional para descrever o espaço filmado pela câmara, podemos imaginar que o centróide da cabeça se encontra em cada momento posicionado nesse sistema à frente da câmara. Se o tomarmos como o ponto de origem relativamente aos

elementos expressivos que pretendemos observar no utilizador, podemos imaginar nessa origem os três eixos (Figura 5.7) que se encontram rodados de acordo com as três possibilidades face ao posicionamento da câmara.

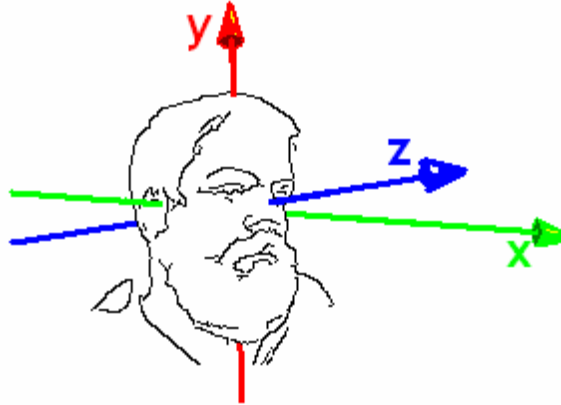
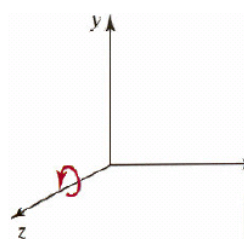


Figura 5.7 – Eixos relativos

O eixo dos Z, é perpendicular ao plano de captura da imagem. O eixo X é a projecção horizontal e o Y é a projecção vertical.

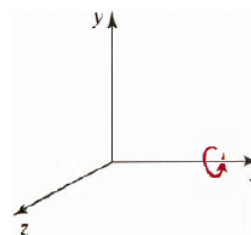
O rodar da cabeça para a esquerda ou para a direita, equivale a uma rotação no plano definido pelos eixos de X e Z. O inclinar da cabeça para a direita e para a esquerda, equivale a uma rotação no plano definido pelos eixos X e Y. Finalmente o inclinar para baixo ou para cima da cabeça, equivale a uma rotação da cabeça de acordo com os eixos Y e Z.

Conhecendo-se esses três ângulos, conseguimos pelo uso de matrizes de transformação facilmente localizar, na imagem recolhida da câmara, a área onde se pode esperar que estaria aplicada a máscara [17]:



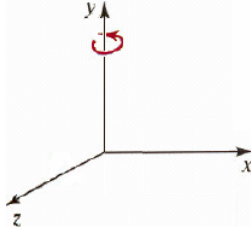
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Figura 5.8 - Rotação individual do eixo ZZ



$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Figura 5.9 - Rotação individual do eixo XX



$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Figura 5.10 - Rotação individual do eixo YY

Para localizar correctamente a máscara a aplicar ao utilizador, necessitamos:

- de conhecer $\hat{\alpha}, \hat{\beta}, \hat{\chi}$, ângulos de rotação dos eixos xx, yy e zz respectivamente;
- de determinar o centróide da cabeça, através da estimativa dos marcadores (ver em 5.4);
- de criar a matriz de transformação T que quando aplicada às coordenadas x, y, z dum ponto da máscara as projecta, de acordo com os ângulos nas coordenadas relativas x', y', z' da imagem capturada.

Como a imagem capturada é um simples mapa de pixels, é ainda necessário converter as coordenadas tridimensionais através de uma projecção em coordenadas do próprio mapa de bits onde finalmente se extraem os dados pretendidos (Figura 5.11).

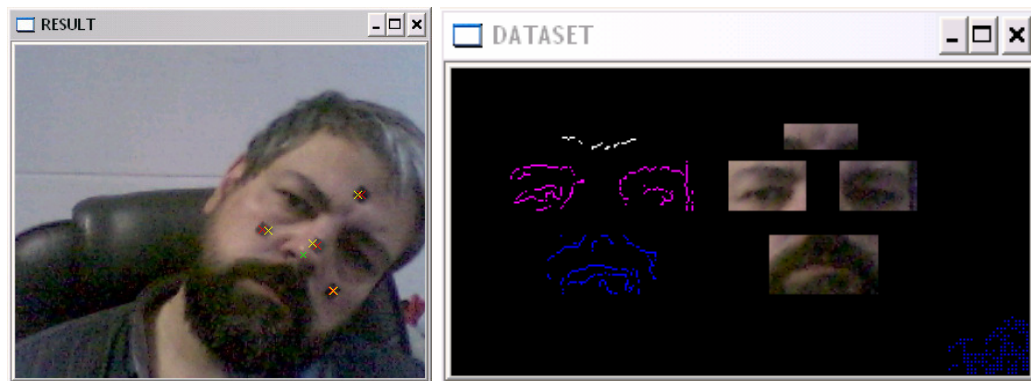


Figura 5.11 – Normalização das áreas de interesse

É nesse processo (desde a captura até à projecção na imagem capturada dos elementos pretendidos) que consiste a normalização da máscara.

Essa normalização²¹, permite que os classificadores sejam aplicados a dados com uma variância menor, tornando-os relativamente imunes aos ângulos de posicionamento da cabeça.

²¹ As áreas de interesse são processadas de acordo com a máscara normalizada do utilizador, pelo que se varrem áreas rectangulares de pontos nas coordenadas virtuais da máscara. Essas coordenadas são projectadas através das matrizes de transformação dos ângulos na matriz de pontos constituída pela imagem capturada. O resultado é que independentemente do

5.3 *Macro-visão*

Como é evidente a imagem captada pela câmara é rica em pormenores que numa forma geral pouco ajudam o propósito de reconhecimento da expressão do utilizador. Se a isso acrescentarmos o ruído das imagens, cada vez inserimos mais elementos que escondem o que verdadeiramente nos interessa analisar na imagem.

Na resolução deste tipo de problema, um dos factores a ter em linha de conta é a qualidade da câmara. As tradicionais *WebCam*, como as utilizadas neste trabalho, são extremamente sensíveis à falta de luz, produzindo imagens absolutamente ruidosas se o nível de iluminação for médio ou fraco.

No extremo oposto da iluminação, também se verifica que são extremamente sensíveis ao excesso de luz, pelo que a aplicação dum cenário com níveis de iluminação fortes produz resultados em que as cores praticamente desaparecem criando-se uma nuvem branca na imagem capturada.

Tanto num caso, como no outro, o problema base está na variação dos pormenores que são recolhidos para a imagem.

Caso a iluminação seja fraca, surge um granulado parasita na imagem que pode mesmo assumir proporções impeditivas do reconhecimento se não for filtrado.

Se a iluminação for demasiado forte, esbatem-se as linhas expressivas do utilizador podendo chegar-se ao extremo de apenas se reconhecer o contorno global do utilizador (e pouco mais).

Através de uma filtragem podem-se eliminar aspectos de ruído desinteressantes para a análise da imagem. É neste sentido que aplicamos um filtro *gaussiano* à imagem originalmente recolhida.

Depois, para evitar as variações quer por excesso quer por defeito de luz, lançamos um operador Canny sobre a imagem para obter os contornos gerais da imagem²².

Nesta fase, a imagem já se apresenta relativamente robusta às variações de luminosidade, uma vez que apenas temos as linhas de contorno. Contudo, mesmo assim existem linhas parasitas

posicionamento do utilizador, para efeito de classificação dos elementos expressivos são apresentados sempre da mesma forma (Figura 5.11) (ainda que no limite de rotação dos diferentes eixos essas áreas possam já ser falhas de informação útil).

²² Utilizamos o operador *canny* porque já faz parte das bibliotecas base do *OpenCV* que é o software de suporte à programação realizada em Visual C++.

resultantes do excesso de ruído ou do excesso de zelo do algoritmo no respeitante ao tratamento dos pormenores do utilizador.

Esses excessos, apenas aumentam a complexidade dos dados levando a que seja difícil treinar classificadores de forma fiável.

Se estudarmos um observador humano rapidamente concluímos que não há a necessidade de conseguir discernir as manchas do rosto de um indivíduo para conseguir localizar os olhos, nariz, boca, etc. ...

Duma forma geral é mais o facto de fazermos um reconhecimento informado que nos leva a localizar rapidamente esses elementos numa pessoa. Isso é muito mais importante que propriamente a nossa acuidade óptica. Para o reconhecimento informado, requeremos uma quantidade impressionantemente pequena de dados visuais, conseguindo por isso distinguir onde estão os elementos do rosto de uma pessoa, mesmo a distâncias para além da nossa capacidade óptica. Esse tipo de dados cinge-se apenas à análise dos contornos vagos e das variações mais importantes de cor. Na prática imaginamos muito mais do que o que vemos.

É justamente aqui que os sistemas de visão computacional apresentam problemas. Seria necessário criar mecanismos de informação dos algoritmos para que pegando em pequenas pistas visuais se consiga discernir com correcção os elementos pretendidos.

Em vez disso temos imagens originais que vêm saturadas de pormenores inúteis para o reconhecimento global, o que nos leva a encontrar um processo simples de resolução do problema.

Através da redução da resolução da imagem e da profundidade de cor consegue-se chegar a uma silhueta de *macro-visão*²³ () capaz de ser introduzida num sistema de reconhecimento rápido.



Figura 5.12 – *macro-visão*

²³ Chamamos "Macro-visão" por que se trata de obter o mesmo efeito que se afastássemos a câmara para inicialmente abranger uma área muito mais alargada e depois, sem mudar a resolução da imagem, recortássemos os pontos à área inicial. Não nos referimos por isso ao efeito de usar lentes Macro das máquinas fotográficas que produzem exactamente o oposto (aumento de pormenor).

Uma vez obtida a redução de resolução desejada cada pixel corresponde (por redução da profundidade de cor) a um bit em que o domínio, obviamente, se cinge a zero ou um.²⁴

É possível sobre a imagem transformada, obter uma matriz com relativamente poucos pontos.

$$M_{x,y} = \begin{bmatrix} m_{1,1} & m_{1,2} & \dots & m_{1,c} \\ m_{2,1} & m_{2,2} & \dots & m_{2,c} \\ \dots & \dots & \dots & \dots \\ m_{l,1} & m_{l,2} & \dots & m_{l,c} \end{bmatrix} \quad (\text{Exp. 5.1})$$

Nesta matriz x, y são as coordenadas do ponto dentro da matriz imagem e vão desde 1 até ao limite de colunas e de linhas, c e l respectivamente.

Esta matriz, já pode ser submetida aos processamentos posteriores desejados para localização dos elementos. Normalmente estes algoritmos servem-se dos dados dessa matriz em forma de variáveis simples, como se de uma matriz linha ou coluna se tratasse com $c \times l$ variáveis de entrada.

$$V = [v_1 \quad v_2 \quad \dots \quad v_k] = [m_{1,1} \quad m_{1,2} \quad \dots \quad m_{1,c} \quad \dots \quad m_{l,1} \quad m_{l,2} \quad \dots \quad m_k] \quad (\text{Exp. 5.2})$$

(Sendo $k = l \times c$, apenas assim indicado para referência mais à frente)

Apesar das inúmeras diferenças se que podem encontrar nos utilizadores, mesmo para um observador destreinado, o estudo atento da imagem de macro-visão deixa antever o tipo de posicionamento do utilizador ().

No respeitante à resolução para efeito da matriz de macro-visão inicial são utilizados 18 por 22 pontos num total de 396 variáveis de entrada.

Quanto mais pontos se acrescentarem maior é o tempo necessário para o processamento de cada *frame* por parte da estimativa de localização dos marcadores. Os valores apresentados foram os primeiros, de entre os experimentados, que permitiam uma manipulação fácil através da ferramenta *Rattle*, com o máximo de informação disponível.

²⁴ Neste caso, como estamos a realizar uma detecção tendo por base uma imagem de contornos já não se põe o problema da redução da profundidade de cor.

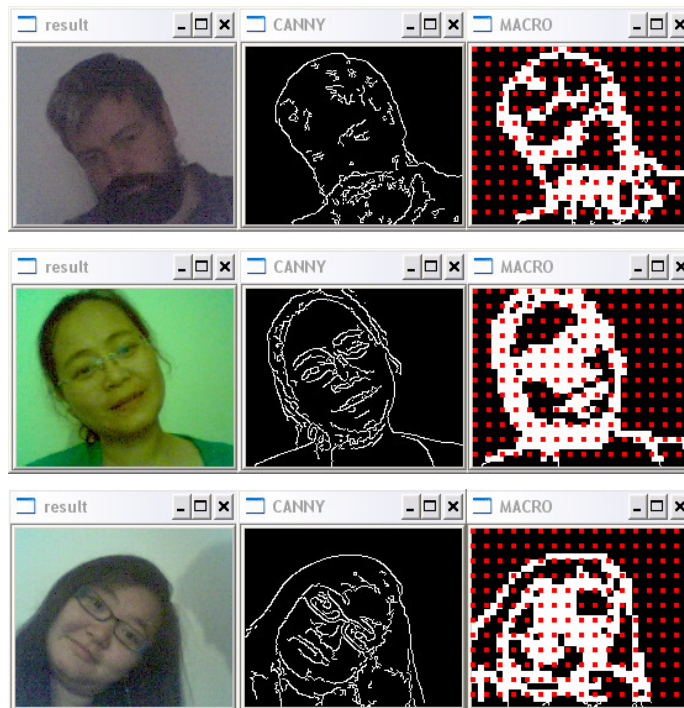


Figura 5.13 – Pontos de recolha na macro-visão

Os detalhes que se perdem, poderiam ser essenciais para a identificação do utilizador, mas para a determinação de ângulos de inclinação podem ser dispensáveis.

5.4 Treino usando Marcadores

Este sistema, à semelhança de muitos outros baseados em *machine learning*, passa por uma fase de treino e afinação que constitui a base para a aplicação em tempo real.

Neste caso concreto, toda a essência do treino consiste, numa primeira fase, na detecção dos pontos onde se encontram os elementos desejáveis para análise da expressão.

Isto implica numa primeira fase obter *clips de video* com diferentes posicionamentos do utilizador que possam ser aplicados para extracção de medições tais como posição e ângulos da cabeça, entre outros.



Figura 5.14 - Exemplos de marcadores

Por este motivo, decidimos aplicar marcadores (Figura 5.14) de forma a dotar o sistema de treino da capacidade de localização automática dos mesmos e com essa informação calcular os demais elementos.

De entre os diferentes marcadores utilizados os que produziram melhor efeito que foram os feitos de material não reflector (pretos).

Estes foram colocados em zonas do rosto que permitem o reconhecimento das métricas pretendidas sem prejuízo da leitura dos pontos de interesse das mesmas.

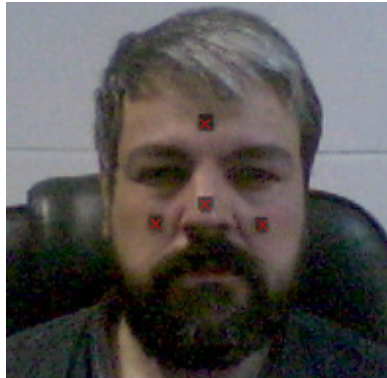


Figura 5.15 - Marcadores identificados

Concretamente, como se mostra na Figura 5.15, os marcadores estão colocados um a meio da testa, acima do cenho, outro ligeiramente acima da ponta do nariz (para evitar interferir com as linhas de contorno) e os laterais estão colocados abaixo do marcador do nariz, em ambas as faces, alinhados horizontalmente pela linha das narinas e verticalmente pela pupila dos olhos.

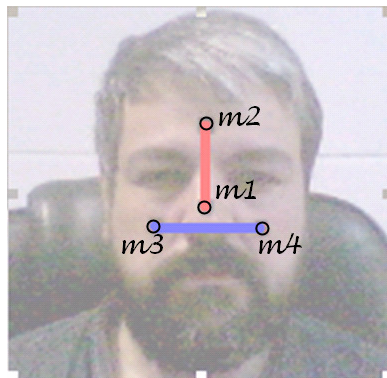


Figura 5.16 - Marcadores numerados

O reconhecimento automático produz as coordenadas (bidimensionais) dos quatro marcadores: $m1 = (x1, y1)$, $m2 = (x2, y2)$, $m3 = (x3, y3)$ e $m4 = (x4, y4)$, de acordo com a Figura 5.16.

A disposição indicada, não interfere com as diversas áreas de interesse e como tal permite extrair da face as características que se encontrariam mesmo sem marcadores.

A remoção dos marcadores para fim de detecção do centróide e para cálculo dos ângulos é um processo algoritmicamente pouco exigente pelo que pode ser realizado em tempo real durante o próprio treino.

O efeito sobre a imagem de "Macro-visão" usada para recolha de pontos de alinhamento não sofre interferências dos marcadores, como se ilustra na Figura 5.17 e na Figura 5.18.

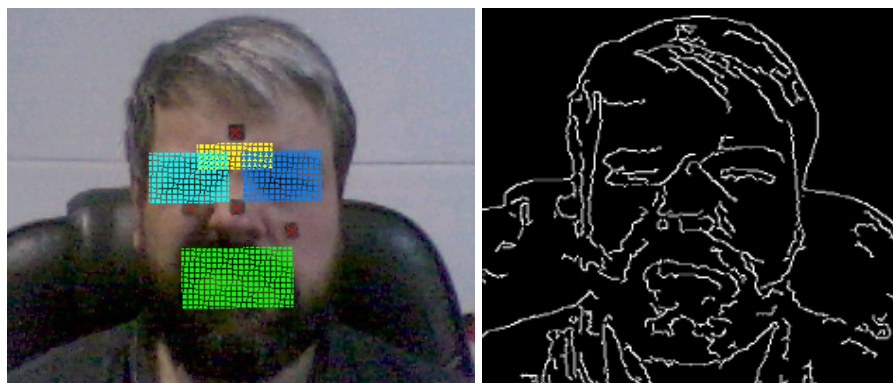


Figura 5.17 - Imagem removidos os marcadores



Figura 5.18 - Macro-visão, removidos os marcadores

5.4.1 Factores de normalização

O centróide do rosto é de importância fulcral para que se possam treinar classificadores aplicados a áreas específicas da imagem. Essa orientação específica tem por vantagem principal a eliminação do ruído geral que rodeia o utilizador.

Do ponto de vista prático das variações anatómicas de cada utilizador, é evidente que não é possível aplicar os marcadores exactamente no mesmo local da face de utilizador para utilizador.

Basta reflectirmos nas variações existentes ao nível do nariz (pontiagudo, achatado, etc. ...) que rapidamente concluímos que os marcadores são aplicados em coordenadas reais diversas consoante o utilizador.

Contudo, quando se tem o utilizador de frente para a câmara (no início de cada treino) temos a possibilidade de extrair a distância entre marcadores e com ela poder redimensionar a imagem de forma a esticá-la à máscara normalizada.



Figura 5.19 - Altura e largura para efeito de normalização

Para esse efeito basta-nos o cálculo da distância euclidiana entre os marcadores $m1$ a $m2$ e $m3$ a $m4$:

$$a = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \quad (\text{Exp. 5.3})$$

$$l = \sqrt{(x4 - x3)^2 + (y4 - y3)^2} \quad (\text{Exp. 5.4})$$

Para efeito de treino dos nossos modelos tomamos por base a altura e a largura do modelo base, sendo os factores de alargamento horizontal (fh) e vertical (fv) unitários. Se o modelo fosse o de uma pessoa com uma largura ou altura de face diferente, o factor de alargamento seria dado directamente pelos dois rácios seguintes, onde a' e l' seriam os valores medidos no início para esse modelo.²⁵

$$fv = \frac{a'}{a} \text{ e } fh = \frac{l'}{l} \quad (\text{Exp. 5.5})$$

No caso concreto do nosso modelo de teste, os rácios serão iguais a um.

5.4.2 Regressão na estimativa das coordenadas dos marcadores

O objectivo principal da utilização dos marcadores é o de criar um conjunto de dados em que estejam representados pontos da imagem de *Macro-visão* (onde já não figuram os marcadores) e as coordenadas dos quatro marcadores.

²⁵ Esses dois factores seriam aplicados imediatamente antes da aplicação do centro no acto de transformação das coordenadas $3d$ para $2d$, sempre que pretendemos obter um ponto de cor concreto.

Esses conjuntos de dados servem para ser submetidos ao treino de modelos de regressão que identificam a posição dos marcadores através da imagem de *Macro-visão* de um utilizador qualquer em frente à câmara (desde que corresponda ao modelo que está em uso).

Para além deste modelo de regressão consegue-se ainda criar o conjunto de dados correspondente ao rasto, onde apenas figuram as coordenadas dos marcadores de um pequeno conjunto de *frames* anteriores consecutivas e a posição dos marcadores da *frame* actual. Esse conjunto de dados, serve para treinar o modelo de regressão para implementação do subsistema de *Reality Check* (ver em 5.5).

Para efeito de estimativa, foi utilizada a ferramenta *rattle* que corre em R, e utilizando as opções nela disponíveis testámos o acerto dos métodos *Rpart*, *lm* e *Nnet*. Nesses testes fomos confrontados com vários problemas associados à configuração dos métodos.

No caso do *Rpart*, a selecção do parâmetro "*Bucket Size*" controla o processo de discretização dos dados para poder correr o modelo que é baseado em árvores de classificação e por isso influencia os resultados do método. O valor por defeito (7) para o conjunto dos nossos dados produz resultados satisfatórios, pelo que não lhe mexemos.

No caso do *LM*, não há parametrizações aplicáveis.

No caso das redes neuronais, a ferramenta oferece a possibilidade de configuração do número de neurónios da camada intermédia sendo o valor por defeito de 10 neurónios. Contudo, o limite de pesos que se podem afinar, neste pacote de redes neuronais, está fixado em 1000. As nossas variáveis correspondendo ao mapa de bits, ainda que na forma processada já descrita, tem uma dimensão de 18 por 22 pontos, ou seja um total de 396 variáveis o que para 10 neurónios excede largamente o limite de pesos das redes. Isto obriga à configuração de apenas um neurónio.

Na Tabela 5.1, mostram-se os resultados da aplicação dos três métodos aplicados avaliados segundo a perspectiva da correlação. Como se verifica pela tabela, apenas o método *Rpart* apresenta resultados satisfatórios, ainda que para a estimativa da coordenada Y3 a correlação seja bastante inferior.

Tabela 5.1 - Correlação para a estimativa das coordenadas dos marcadores

Marcador	Coordenada	Correlação			
		<i>rpart</i>	<i>glm</i>	<i>nnet</i>	média
1	X	94,17%	75,79%	89,84%	0,00%
	Y	92,29%	51,96%	46,22%	0,00%
2	X	96,70%	73,73%	99,29%	0,00%
	Y	91,40%	20,76%	40,29%	0,00%
3	X	94,62%	62,61%	83,78%	0,00%
	Y	71,34%	-16,08%	40,40%	0,00%
4	X	95,93%	49,95%	76,49%	0,00%
	Y	90,65%	8,60%	58,12%	0,00%

Na respeitante ao GLM, o pacote base não consegue processar convenientemente os dados trocando os valores reais pelos factores que guarda em memória à medida que carrega os valores dos dados, forma como o R faz em vez de utilizar os dados reais ao carregar o *dataset* para memória. Por esse motivo os resultados deste método são suspeitos.

A maior desilusão sucede com as redes neuronais, onde se esperaria um pouco mais de acerto.

No respeitante à correlação, que estuda as variações da sequência de dados estimados em comparação com os dados a estimar, não é a métrica mais interessante para avaliar a eficácia na estimativa dos marcadores, uma vez que aqui nos interessa determinar a posição dos mesmos e é nos desvios que se verificam erros.

A mostra o erro normalizado que é muito mais significativo neste contexto. Aqui mostra-se a comparação do acerto de cada média se confrontado com a média das coordenadas de cada marcador. Também aqui se demonstra que o *Rpart*, neste pacote tem melhor acerto que os demais métodos. A excepção está para a coordenada X2, onde as redes neuronais têm um acerto mais interessante com NMSE de 1,51%. O pior resultado verifica-se para a coordenada Y3 com NMSE de 49,91% o que é francamente fraco, ainda que os outros dois métodos sejam claramente piores.

Tabela 5.2 - Erro normalizado na estimativa das coordenadas dos marcadores

Marcador	Coordenada	Taxa de erro: NMSE			
		<i>rpart</i>	<i>glm</i>	<i>nnet</i>	média
1	X	11,56%	76,31%	24,96%	100,00%
	Y	15,27%	237,86%	181,07%	100,00%
2	X	6,56%	70,82%	1,51%	100,00%
	Y	16,65%	1683,64%	276,51%	100,00%
3	X	10,51%	149,65%	45,47%	100,00%
	Y	49,91%	3951,31%	282,47%	100,00%
4	X	8,33%	288,88%	85,52%	100,00%
	Y	18,14%	5678,04%	191,49%	100,00%

Ainda nos fica a ideia de que a ferramenta poderia ter opção para utilizar outros métodos como por exemplo os *Kernel SVM* aplicados à regressão.

5.4.3 Estimativa do centróide inicial

Não dispomos de um teorema que nos defina onde se situa o centróide (tridimensional) de um rosto dada uma imagem bidimensional.

No entanto e porque necessitamos de o estimar para efeito de uma melhor definição das áreas de interesse do rosto na classificação dos elementos expressivos, tivemos a de criar uma regra simples que pudesse ser aplicada ao nosso contexto.

Para tal aplicou-se o facto de o interesse na determinação do centróide ser o de localizar elementos do rosto e que por isso estamos a tentar estimar um ponto dentro de uma área relativamente alargada. Ou seja o erro de estimativa, esbate-se dada a largura da imagem ao ponto de se tornar quase insignificante quanto o utilizador olha de frente e tornando-se ligeiramente representativo apenas quando o utilizador vira francamente a cabeça para os lados ou a inclina francamente para cima ou para baixo.

A resolução deste problema passa pelo uso duma regra geral que seja aplicável independentemente do modelo em questão. Isso implica que essa regra tenha de aproveitar as próprias métricas recolhidas na posição inicial do modelo e em função delas ajustar as coordenadas.

A regra escolhida tem por base a altura do rosto e é a seguinte: o centróide do rosto situa-se nas coordenadas do primeiro marcador, em x e y , e à distância desse marcador, em profundidade igual à altura base calculada a (Exp. 5.3). O que equivale a dizer que as coordenadas do marcador 2 serão aproximadamente: $M2 = (0,0,a)$.

Contudo as diferenças entre as formas do rosto levam a que valha a pena reflectir sobre as proporções do rosto sem entrarmos em preciosismos excessivos.

É duma forma geral aceite que um rosto largo tem normalmente associado um nariz achatado e um rosto mais alto e estreito, o oposto.

Ou seja o rácio entre a altura e a largura do rosto (medidas expressas por a' e l' para um dado modelo de utilizador) são uma primeira indicação sobre o alongamento do nariz.

Esse rácio pode ser expresso simplesmente como $fn = \frac{a'}{l'}$.

Por outro lado é igualmente de se esperar que quanto maior for o rosto, também maior será o nariz (e a prova disso está na observação do nariz de um bebé). Quanto maior for o nariz mais este se projecta para além da testa. Nesse sentido tanto se pode utilizar a métrica da altura como da largura. No nosso caso optamos por utilizar a altura sendo o cálculo aproximado da saliência do nariz, determinado por um quinto²⁶ desse valor após aplicado o rácio fn .

Assim as coordenadas dos marcadores serão as seguintes:

$$M1 = (0,0,a' + \frac{a' \times fn}{5}) \quad (\text{Exp. 5.6})$$

$$M2 = (x2 - x1, y2 - y1, a') \quad (\text{Exp. 5.7})$$

$$M3 = (x3 - x1, y3 - y1, a') \quad (\text{Exp. 5.8})$$

²⁶ Valor tirado do modelo estudado

$$M4 = (x4 - x1, y4 - y1, a') \quad (\text{Exp. 5.9})$$

A diferença de profundidade em termos dos marcadores $m2$, $m3$ e $m4$ é praticamente irrelevante, pelo que lhe atribuímos a mesma coordenada z .

O centróide da cabeça, obviamente assume as coordenadas $(0,0,0)$, passando a ser o ponto de origem para efeito de rotação da máscara normalizada.

5.4.4 Determinação dos ângulos

Durante a fase de treino, não estamos atados a restrições de tempo no respeitante aos algoritmos empregues. Assim, é possível activar processamentos temporalmente mais complexos sem qualquer prejuízo para o sistema final.

No respeitante à determinação dos ângulos, em vez de encetar esforços na descoberta de modelos matemáticos complexos, baseados na distribuição dos marcadores, reconhecidos num dado momento, para efeito de determinação dos ângulos de rotação, passamos a um processo alternativo, o da procura pela melhor afinação.

Partimos da posição inicial, em frente à câmara em que os ângulos são nulos, e actualizamos-os *frame* a *frame* rodando os marcadores da posição inicial até alcançarmos uma posição cuja distância, aos marcadores estimados, é desprezível.

Essa actualização é realizada através de ciclos de variação dos três ângulos da *frame* anterior. Se definíssemos um espaço vectorial de três eixos onde se projectasse um ponto, de coordenadas iguais aos três ângulos (espaço esse cujos pontos válidos, definiriam um cubo com lado igual a $2 \times \pi$ variando em cada eixo de $[-\pi, \pi]$) a estimativa dos ângulos equivaleria a procurar na vizinhança do ponto da *frame* anterior (definido pelos ângulos nessa *frame*) o mais próximo que aplicado à rotação dos marcadores da posição original, se aproxima mais aos estimados nesta *frame*.

Esses ângulos são posteriormente introduzidos num conjunto de dados onde figuram as coordenadas de cada um dos quatro marcadores como variáveis independentes e os ângulos como variáveis dependentes.

Esse conjunto de dados, por treino de um modelo de regressão, permite obter num dado momento uma estimativa rápida do estado de rotação da cabeça do utilizador.

Como se observa pela Tabela 5.3, os valores mínimos de correlação obtida são 0,79 o que corresponde a um valor não brilhante, mas aceitável.

No respeitante ao erro máximo normalizado, ele situa-se no valor dos 0,49 o que já é um valor bastante significativo.

A razão para estes resultados pobres na rotação segundo o eixo Y, está relacionada com o facto de a determinação dos ângulos, durante o processo de construção do ficheiro de dados,

sofrer de algum ruído quando é suposto não haver rotação no eixo dos y (o que equivale à rotação resultante de rodar o pescoço para a esquerda ou para a direita).

Tabela 5.3 - Resultados de regressão – rotação segundo os eixos

Eixo	corr(rpart)	corr(glm)		NMSE(rpart)	NMSE(glm)
x	0,926836451	0,85127931		0,1409742	0,2753235
y	0,791433973	0,71066758		0,3736323	0,4949516
z	0,969739889	0,99871752		0,0596045	0,0025633

O melhor resultado foi o obtido na estimativa de rotação do eixo Z (uma vez que corresponde a inclinar a cabeça para os lados (esquerda e direita). Aqui obtiveram-se valores para a correlação na ordem dos 0,998 o que é muito bom. A prova final encontra-se no valor para o erro, NMSE, de 0,0025, o que para este efeito é excelente.

No respeitante às redes neuronais, devido ao funcionamento duvidoso da ferramenta *rattle*, estas apresentam exactamente os mesmos resultados que o discriminante linear. Como não conseguimos descortinar o motivo para esses resultados, optamos por retirá-los desta análise (até porque neste caso seriam redundantes).

5.5 Reality Checks

Como é evidente, todo o processamento é por natureza passível de errar.

Se estivermos a operar com velocidades de processamento, no reconhecimento de expressões, abaixo dos 0.066 segundos (para taxas de amostragem na ordem das 15 frames por segundo), podemos utilizar com alguma confiança a informação de reconhecimento proveniente duma iteração anterior para validar as conclusões da actual.

Para além disso podemos ainda aproveitar essa informação para com maior certeza confirmar a localização dos elementos a serem submetidos à validação.

A título de exemplo: se num instante t reconhecemos a posição de todos os componentes expressivos (olhos, boca, etc. ...) no instante $t + \frac{1}{15}s$ é de se esperar uma variação mínima no centróide de cada um deles. Pelo que se pode orientar de imediato a análise de imagem para a informação que tínhamos no momento anterior e ajustá-la a partir daí.

Por comparação da posição de dois centróides sucessivos, é possível prever ainda com mais exactidão o local onde é de se esperar que se encontre o do momento seguinte.

O grau de confiança que se atribui à localização dos centróides por esta via pode ser ajustado de forma decrescente para deixar transparecer a existência de estimativas cada vez mais

distantes do último momento confirmado e por isso menos exactas da posição real de cada elemento.

Se num momento o reconhecimento apontar para um centróide excessivamente deslocado do determinado no momento anterior (desde que esse não tenha sido estimado por análise sucessiva de centróides) podemos despoletar o mecanismo de correcção por afastamento da realidade provável, mecanismo a que chamamos de *Reality Check*.

Note-se que este mecanismo resulta, na prática, do factor humano que é necessário ter em linha de conta. Esse factor é o das limitações de movimento do próprio ser humano. Essas limitações são de ordem muscular, de elasticidade dos tecidos e das articulações ósseas que constituem os pontos móveis do esqueleto humano.

Esse conjunto de factores leva a considerar que todo o movimento (dado que o utilizador se encontra sentado na cadeira) tem limites bem definidos. Existem limites à velocidade de resposta dos músculos. Existem limites à elasticidade da pele e outros tecidos que delimitam por sua vez até onde se conseguem mover determinados elementos expressivos.

Logo, quando há um reconhecimento facial em que os elementos estão a ser localizados em pontos de tal forma diversos do momento anterior que humanamente não são possíveis de serem realizados, o melhor é utilizar a estimativa nesse instante e esperar que haja de seguida pelo menos dois ou três reconhecimentos que confirmem esse novo posicionamento [3].

Como se ilustra na figura anterior (), na estimativa da posição no instante seguinte ao último onde se tem a confirmação efectiva da posição do utilizador, após dois reconhecimentos sucessivos, define um movimento aproximadamente linear.

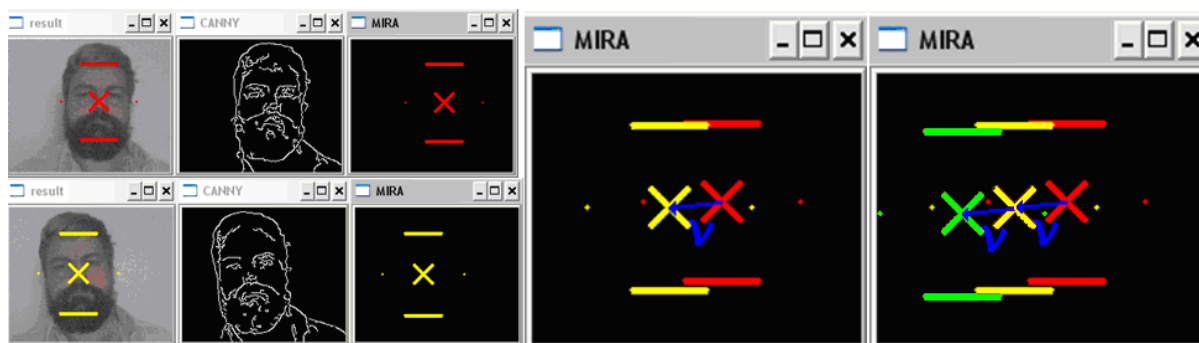


Figura 5.20 – Estimativa de posicionamento

Essa linearidade não é o esperado num utilizador humano. A cabeça, não é uma parte anatómica solta. Como todas as outras partes do corpo humano está presa à estrutura do corpo, pelo que essa estrutura delimita e condiciona os movimentos. Dessa forma o ponto seguinte dum movimento da cabeça, dada a posição do utilizador frente à câmara dificilmente teria um quase deslocamento inteiramente linear. Este seria mais acentuado se estivermos no acto de início de

um movimento, ou o oposto do final. É por isso difícil criar uma expressão que traduza matematicamente uma estimativa aceitável do centróide seguinte.

Em todo o caso, podemos recorrer a um modelo de previsão baseado em métodos de regressão. Aqui basta criar uma série de registos consecutivos sobre a posição do centróide, referentes aos movimentos anteriores, e por efeito de regressão estimar a posição seguinte.

Essa sequência de registos constitui o *rasto* do utilizador e, quanto maior for (até um dado limite de abrangência de tempo) melhor poderá levar à estimativa da posição seguinte.

Num outro tipo de estudo, este rasto poderá ser uma forma interessante de abordar o problema dos utilizadores que realizam movimentos involuntários devidos a espasmos musculares. A descoberta de padrões nesse tipo de movimentos poderá contribuir para ajudar um sistema a distinguir melhor os movimentos involuntários dos demais e por essa via reconhecer os comandos que o utilizador tenta veicular separando-os dos involuntários.

No respeitante a esta técnica, o uso do rasto do utilizador, a aplicabilidade vai ainda mais longe. No reconhecimento dos ângulos de inclinação da cabeça, pode-se repetir o mesmo processo. De igual forma, uma vez determinadas as métricas sobre o rosto do utilizador, pode-se ainda aplicar o conceito de rasto sobre os próprios elementos expressivos, como por exemplo sobrancelhas, pupila do olho, etc. ... para se conseguir de forma mais fiável detectar o posicionamento dos mesmos.

Treinando-se um modelo de regressão para cada um destes efeitos, desde que apresentem uma boa taxa de acerto, passamos a ter meios bastante eficazes de implementar mecanismos de *Reality Check*. Basta-nos para isso configurar limites de tolerância que por comparação das posições estimadas pelos modelos de regressão por rasto, e pelos modelos de regressão por análise da imagem nos dizem rapidamente se o que estamos a reconhecer é ou não aceitável dentro da realidade esperada, bastando para isso a simples observação da distância entre os dois tipos de estimativas.

5.5.1 Regressão para estimativa de rasto

Utilizando a ferramenta *Rattle*, após recolhidos os rastos de uma série de animações e criados dois ficheiros (dessa recolha), um com 70% dos dados para treino e outro com 30% para teste, partimos para a exploração dos métodos de regressão, aplicados ao rasto, que a ferramenta oferece.

Infelizmente, como já vimos, não são muitos. Contamos apenas com *árvores de regressão*, *modelo linear* e *redes neuronais*.

Como se pode observar pela Tabela 5.4, a correlação mais baixa é de 95,97% o que em qualquer situação seria uma correlação muito boa entre a estimativa e os valores originais.

Não é de todo surpreendente uma vez que para qualquer dos marcadores estamos a procurar uma sequência baseando-nos para isso em colunas que já por si apresentam uma forte correlação (o rasto).

Tabela 5.4 - Regressão aplicada ao rasto dos marcadores

	rpart	glm	nnet
xx1	98,486%	99,962%	99,962%
yy1	97,087%	99,612%	99,611%
xx2	98,667%	99,943%	99,953%
yy2	96,442%	99,466%	99,466%
xx3	98,215%	99,912%	99,912%
yy3	95,974%	99,022%	99,034%
xx4	98,090%	99,914%	99,914%
yy4	97,120%	99,498%	99,495%
	97,510%	99,666%	99,668%

A métrica de avaliação dos modelos utilizada foi uma simples correlação estatística entre dois vectores uma vez que nos é mais importante a tendência da previsão do que o cálculo do erro (que nunca será excessivo para uma previsão consecutiva).

Em todo caso, dos três métodos que a ferramenta nos oferece, o que melhor acerto médio produz no teste, baseado na correlação, é o das redes neuronais aplicado com uma arquitectura de 10 neurónios na camada intermédia.

O modelo linear apresenta igualmente uma excelente correlação pelo que em comparação com as redes neuronais se vê claramente que pouco se ganha em usar uma estrutura com 10 neurónios.

O que fica provado, é que os *reality checks*, baseados na análise do rasto, são uma excelente forma de avaliar uma *frame* na medida em que as taxas de acerto conseguem ter uma correlação de mais de 99%.

Se considerarmos a correlação como sendo o nível de confiança atribuível à estimativa dada por um modelo, podemos ainda observar o nível de degradação esperada da confiança se aplicarmos recorrentemente a correlação a *frames* onde não se consiga determinar o posicionamento real dos marcadores (utilizando estimativas consecutivas).

Tabela 5.5 - Degradação do nível de confiança utilizando redes neuronais

	X1	Y1	X2	Y2	X3	Y3	X4	Y4	Média
1	99,962%	99,611%	99,953%	99,466%	99,912%	99,034%	99,914%	99,495%	99,668%
2	99,631%	99,281%	99,621%	99,136%	99,580%	98,705%	99,582%	99,165%	99,338%
3	98,971%	98,623%	98,962%	98,479%	98,921%	98,052%	98,923%	98,509%	98,680%
4	97,664%	97,322%	97,655%	97,179%	97,615%	96,757%	97,617%	97,208%	97,377%
5	95,103%	94,769%	95,094%	94,631%	95,055%	94,219%	95,057%	94,659%	94,823%
6	90,179%	89,863%	90,171%	89,732%	90,134%	89,342%	90,136%	89,758%	89,914%
7	81,084%	80,800%	81,077%	80,682%	81,043%	80,331%	81,045%	80,705%	80,846%
8	65,553%	65,323%	65,547%	65,228%	65,520%	64,944%	65,521%	65,247%	65,360%
9	42,846%	42,695%	42,842%	42,633%	42,824%	42,448%	42,825%	42,646%	42,720%
10	18,304%	18,239%	18,302%	18,213%	18,294%	18,134%	18,295%	18,218%	18,250%
11	3,340%	3,329%	3,340%	3,324%	3,339%	3,309%	3,339%	3,325%	3,331%
12	0,111%	0,111%	0,111%	0,111%	0,111%	0,110%	0,111%	0,111%	0,111%

Tabela 5.6 - Degradação do nível de confiança utilizando o modelo linear generalizado

	X1	Y1	X2	Y2	X3	Y3	X4	Y4	Média
1	99,962%	99,612%	99,943%	99,466%	99,912%	99,022%	99,914%	99,498%	99,666%
2	99,629%	99,280%	99,609%	99,134%	99,578%	98,691%	99,580%	99,165%	99,333%
3	98,964%	98,618%	98,945%	98,473%	98,914%	98,033%	98,917%	98,504%	98,671%
4	97,649%	97,307%	97,630%	97,164%	97,600%	96,730%	97,602%	97,195%	97,360%
5	95,071%	94,738%	95,053%	94,599%	95,023%	94,177%	95,025%	94,629%	94,789%
6	90,117%	89,802%	90,100%	89,670%	90,072%	89,269%	90,074%	89,698%	89,850%
7	80,971%	80,687%	80,955%	80,568%	80,929%	80,209%	80,931%	80,594%	80,731%
8	65,368%	65,139%	65,355%	65,043%	65,335%	64,753%	65,336%	65,064%	65,174%
9	42,603%	42,454%	42,595%	42,392%	42,582%	42,202%	42,583%	42,405%	42,477%
10	18,097%	18,033%	18,093%	18,007%	18,087%	17,926%	18,088%	18,012%	18,043%
11	3,265%	3,254%	3,265%	3,249%	3,263%	3,234%	3,264%	3,250%	3,255%
12	0,106%	0,106%	0,106%	0,106%	0,106%	0,105%	0,106%	0,106%	0,106%

Tabela 5.7 - Degradação do nível de confiança utilizando as árvores de regressão

	X1	Y1	X2	Y2	X3	Y3	X4	Y4	Média
1	98,486%	97,087%	98,667%	96,442%	98,215%	95,974%	98,090%	97,120%	97,510%
2	96,034%	94,670%	96,211%	94,040%	95,770%	93,584%	95,648%	94,702%	95,082%
3	91,312%	90,014%	91,479%	89,416%	91,060%	88,982%	90,944%	90,045%	90,406%
4	82,552%	81,379%	82,703%	80,838%	82,324%	80,446%	82,219%	81,406%	81,733%
5	67,472%	66,514%	67,596%	66,071%	67,286%	65,751%	67,201%	66,536%	66,803%
6	45,074%	44,433%	45,157%	44,138%	44,949%	43,924%	44,892%	44,448%	44,627%
7	20,115%	19,829%	20,152%	19,697%	20,060%	19,602%	20,034%	19,836%	19,916%
8	4,006%	3,949%	4,013%	3,923%	3,995%	3,904%	3,990%	3,950%	3,966%
9	0,159%	0,157%	0,159%	0,156%	0,158%	0,155%	0,158%	0,157%	0,157%

Pela observação das tabelas anteriores (Tabela 5.5 e Tabela 5.6) verifica-se que tanto com as *redes neuronais* como com o *modelo linear* poderíamos num dado momento perder sete *frames* consecutivas tendo mesmo assim um nível de confiança nas estimativas próximo dos 80%. Com as árvores de regressão não poderíamos contar com mais de quatro *frames* para que a confiança não baixasse desse patamar (como se observa na Tabela 5.7).

5.6 Classificação individual dos componentes expressivos

Uma vez identificadas as áreas alvo temos o problema de reconhecimento dividido em problemas mais simples, perfeitamente localizados, pelo que se pode prosseguir com a análise corrente da imagem.

Nesta fase estamos aptos a poder classificar cada um dos elementos expressivos para numa análise posterior identificar a expressão que se está a produzir num dado momento. Estamos pois perante um típico problema de classificação.



Figura 5.21 - Exemplo de pormenor recolhido para classificação

Das áreas de interesse anteriormente ilustradas (), por efeito de aplicação da matriz de transformação conseguimos extrair os pontos direccionados ao componente em análise com a vantagem de os aplicar com os componentes minimamente normalizados.



Figura 5.22 - Áreas de interesse

Em cada *frame* dos *clips* de vídeo, há quatro componentes expressivos em análise. Para estes, durante a fase de treino, é necessário realizar a classificação manual do que cada um deles está a realizar.

5.6.1 Assistente de classificação

Durante o processo de criação dos conjuntos de dados para treino, o sistema ainda não dispõe de informação suficiente para poder reconhecer cada um dos elementos. Por este motivo, justifica-se um assistente onde seja possível guiar o sistema numa aprendizagem supervisionada.

O assistente tem por base duas imagens. A primeira é uma simples janela na qual se mostra o fotograma original capturado num determinado momento. Este, não tem ainda qualquer tipo de processamento e serve apenas para o classificador humano, esclarecer qualquer dúvida sobre o que está a ser exibido num determinado momento pelo modelo.

A segunda imagem (Figura 5.23) é já a processada da qual inicialmente se removeu o ruído e onde posteriormente se aplicou o operador *Canny*.

Este assistente por defeito mostra a última classificação como já activa. Sempre partindo-se do princípio de que o estado do modelo num determinado instante t não difere em demasia do estado no instante $t-1$.

A Figura 5.23, ilustra a disposição das paletas de classificação. Uma para o olho direito (exibida à esquerda), outra para o olho esquerdo. Ambas as paletas tem 12 classificações possíveis que correspondem ao olho em repouso, semi-cerrado, fechado, esbugalhado, virado para cima, virado para baixo, para a esquerda, para a direita, para a direita e cima, para a esquerda e cima, para a direita e baixo e para a esquerda e baixo.



Figura 5.23 - Assistente de classificação

No respeitante ao que denominamos de cenho, só temos uma classificação possível que é o franzir do mesmo.

Posteriormente temos ainda uma para a boca, a posição de repouso, os lábios comprimidos, a boca aberta e os lábios em funil.

Em qualquer momento, se nenhum dos elementos expostos nas paletes for satisfeito, então o elemento expressivo assume o valor de zero.

No respeitante ao âmbito deste trabalho, não é objectivo realizar o treino de todas as *unidades de acção* tal como se define no sistema FACS. Antes pelo contrário, interessa-nos definir um subconjunto mais limitado, mas cujo contraste expressivo seja mais evidente.

A classificação produz um conjunto de variáveis dependentes de natureza binária que são utilizadas para caracterizar cada um dos elementos expressivos (Tabela 5.8, Tabela 5.9, Tabela 5.10 e Tabela 5.11).

Tabela 5.8 - Caracterização do cenho

variável	descrição
<i>cf</i>	cenho franzido

Tabela 5.9 - Caracterização da boca

variável	descrição
<i>bn</i>	boca em repouso
<i>bd</i>	lábios comprimidos
<i>ba</i>	boca aberta
<i>bb</i>	lábios em funil

Tabela 5.10 - Caracterização do olho esquerdo

variável	descrição
<i>oda</i>	olho direito aberto
<i>odsa</i>	olho direito semicerrado
<i>odf</i>	olho direito fechado
<i>ode</i>	olho direito esbugalhado
<i>odoc</i>	olho direito olha para cima
<i>odob</i>	olho direito olha para baixo
<i>odoe</i>	olho direito olha para a esquerda
<i>odod</i>	olho direito olha para a direita
<i>odocd</i>	olho direito olha para cima e para a direita
<i>odoce</i>	olho direito olha para cima e para a esquerda
<i>odobd</i>	olho direito olha para baixo e para a direita
<i>odobe</i>	olho direito olha para baixo e para a esquerda

Tabela 5.11 - Caracterização do olho esquerdo

variável	descrição
<i>oea</i>	olho esquerdo aberto
<i>oesa</i>	olho esquerdo semicerrado
<i>oef</i>	olho esquerdo fechado
<i>oee</i>	olho esquerdo esbugalhado
<i>oeoc</i>	olho esquerdo olha para cima
<i>oeob</i>	olho esquerdo olha para baixo
<i>oeoe</i>	olho esquerdo olha para a esquerda
<i>oeod</i>	olho esquerdo olha para a direita
<i>oeocd</i>	olho esquerdo olha para cima e para a direita
<i>oeoce</i>	olho esquerdo olha para cima e para a esquerda
<i>oeobd</i>	olho esquerdo olha para baixo e para a direita

<i>oeobe</i>	olho esquerdo olha para baixo e para a esquerda
--------------	---

5.6.2 Uso de classificadores com o Rattle

Cada uma das variáveis anteriormente descritas é binária. Contudo a utilização de classificadores independente para distinguir entre elementos alternativos poderia gerar conflitos indesejáveis no sistema de reconhecimento. A título de exemplo podemos apontar as diferentes variáveis referentes às pupilas dos olhos, por exemplo para o olho esquerdo: *oeoc*, *oeob*, *oeoe*, *oeod*, *oeocd*, *oeoce*, *oeobd* e *oeobe*. Como é evidente, se o olho estiver com a pupila a apontar para cima *oeoc=1* não estará certamente a apontar para baixo *oeob=1*. Se tivermos classificadores binários independentes para cada uma dessas variáveis poderíamos cair num conflito de classificação.

Para evitar essas situações essas variáveis são aglutinadas por uma só que tem valores discretos correspondentes às classes {*nada*, *oeoc*, *oeob*, *oeoe*, *oeod*, *oeocd*, *oeoce*, *oeobd*, *oeobe*} (ou seja {0,1,2,3,4,5,6,7} para este caso).

Por este motivo foram criadas as seguintes variáveis combinadas como se descreve na Tabela 5.12.

Tabela 5.12 – Variáveis combinadas para classificação

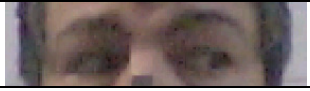
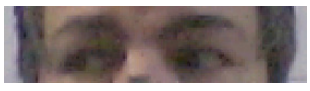
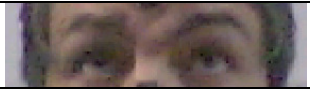
Variável combinada	Descrição	Domínio
<i>od</i>	olho direito	<i>nada</i> , <i>oda</i> , <i>odsa</i> , <i>odf</i> , <i>ode</i>
<i>odo</i>	olho direito olha para...	<i>nada</i> , <i>odoc</i> , <i>odob</i> , <i>odoe</i> , <i>odod</i> , <i>odocd</i> , <i>odoce</i> , <i>odobd</i> , <i>odobe</i>
<i>oe</i>	olho esquerdo	<i>nada</i> , <i>oea</i> , <i>oesa</i> , <i>oef</i> , <i>oe</i>
<i>oeo</i>	olho esquerdo olha para...	<i>nada</i> , <i>oeoc</i> , <i>oeob</i> , <i>oeoe</i> , <i>oeod</i> , <i>oeocd</i> , <i>oeoce</i> , <i>oeobd</i> , <i>oeobe</i>
<i>b</i>	boca	<i>nada</i> , <i>bn</i> , <i>bd</i> , <i>ba</i> , <i>bb</i>
<i>c</i>	"cenho"	<i>nada</i> , <i>cf</i>

5.6.2.1 Classificações para os olhos

Cada um dos olhos é analisado em separado, sendo igualmente classificado em separado.

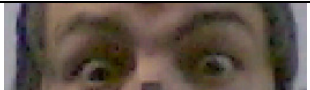
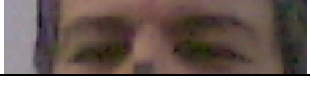
As unidades de acção apresentadas em baixo (61, 62, 63, 43, 5 e 6) só poderão ser identificadas se ambos os olhos forem classificados da mesma forma.

Tabela 5.13 - Utilizador modelo - au 61, 62, 63 e 43

61	Olhos para a esquerda <i>Eyes turn left;</i>	
62	Olhos para a direita <i>Eyes turn right;</i>	
63	Olhos para cima <i>Eyes up;</i>	
43	Olhos fechados <i>Eyes Closed; Relaxation of Levator palpebrae superioris Orbicularis oculi, pars palpebralis</i>	

No respeitante a acções combinadas com as sobrancelhas temos ainda:

Tabela 5.14 - Utilizador modelo - au 5 e 6

5	Elevar das palpebras <i>Upper Lid Raiser; Levator palpebrae superioris</i>	
6	Elevar das bochechas <i>Cheek Raiser; Orbicularis oculi, pars orbitalis</i>	

Contudo se para o olho esquerdo reconhecermos a unidade de acção 6 e para o direito reconhecermos a 46, então poderemos estar perante o piscar do olho esquerdo. Aparentemente o sistema FACS não incorpora esse tipo de unidade de acção. Contudo, para efeito de comando poderá ser um elemento interessante na linguagem de interface da cadeira.

Do conjunto de classificadores que a ferramenta oferece, as *redes neuronais artificiais* e o *boost* não se encontram disponíveis na medida em que o pacote de rotinas do R-Project que a ferramenta *rattle* utiliza não suporta variáveis dependentes com mais de duas classes para estes métodos.

Aplicámos por isso os demais classificadores experimentando algumas parametrizações distintas.

No caso dos olhos temos as variáveis que representam a classificação dos olhos direito e esquerdo.

No respeitante à classificação do olho direito (Tabela 5.15), o classificador que melhor resultado produziu no teste dum conjunto separado de dados, foi o *Random Forest* com uma taxa de acerto da ordem dos 81,9%.

Como se observa pela tabela, aplicando o teste ao conjunto de dados utilizados para o treino, verifica-se que tanto o *Random Forest*, *SVM* de *Kernel Linear* e *SVM* de *Kernel Polinomial* (2º grau) conseguem classificar totalmente os dados. Em, quando submetidos a um treino com conjunto independente de dados do conjunto de teste, verifica-se que a taxa decresce significativamente sendo no pior dos casos de 76,8%. Este facto denota que os modelos se sobre ajustam aos dados de forma significativa.

Tabela 5.15 - Classificação do olho direito

Classificador	Treino e Teste 100% dos dados		Treino 70%/Teste 30%		
	Casos	Acerto	Treino	Teste	Acerto
Linear Model (multinormal)	903	--	632	271	--
Random Forest (500)	903	100,0%	632	271	81,9%
Rpart	903	79,6%	632	271	76,4%
Svm - kernel:Anova RBF	903	99,0%	632	271	77,5%
Svm - kernel:bessel	903	60,6%	632	271	42,8%
Svm - kernel:laplace	903	67,1%	632	271	58,7%
Svm - kernel:linear	903	100,0%	632	271	76,8%
Svm - Kernel:Polinomial (2)	903	100,0%	632	271	80,1%
Svm - kernel:tangente hiperbolica	903	51,6%	632	271	57,6%
Svm Kernel: Radial Basis	903	89,0%	632	271	80,8%

Em termos de consistência o melhor método é o de *kernel* de *base radial* que para além de não se afastar excessivamente do melhor resultado, ficando apenas a 1,1%, mantém uma descida entre as duas taxas de acerto menos acentuada.

Os resultados obtidos para a classificação do olho esquerdo, apresentam algumas diferenças comparativamente aos do olho direito (como se apresenta na Tabela 5.16).

Tabela 5.16 - Classificação do olho esquerdo

Classificador	Treino e Teste 100% dos dados		Treino 70%/Teste 30%		
	Casos	Acerto	Treino	Teste	Acerto
Linear Model (multinormal)	903	--	632	271	--
Random Forest (500)	903	100,0%	632	271	82,3%
Rpart	903	80,4%	632	271	70,7%
Svm - kernel:Anova RBF	903	98,3%	632	271	76,0%
Svm - kernel:bessel	903	59,7%	632	271	38,7%

(Continuação)

Classificador	Treino e Teste 100% dos dados		Treino 70%/Teste 30%		
	Casos	Acerto	Treino	Teste	Acerto
Svm - kernel:laplace	903	65,4%	632	271	62,0%
Svm - kernel:linear	903	99,8%	632	271	72,7%
Svm - Kernel:Polinomial (2)	903	100,0%	632	271	80,4%
Svm - kernel:tangente hiperbolica	903	52,7%	632	271	58,3%
Svm Kernel: Radial Basis	903	89,5%	632	271	78,6%

O melhor acerto, continua a verificar-se para o algoritmo classificador *Random Forest* que no teste chega um pouco mais além do acerto para o olho direito (82,3%).

Ao nível do sobre ajustamento, conseguem-se identificar os classificadores mais afectados, que correspondem aos mesmos que para o olho direito (com a excepção do *kernel linear* que desce uma percentagem mínima de 0,2%).

Dado que na maioria dos casos quando se avalia a expressão baseada nos olhos, ambos os olhos realizam uma determinada acção. Se os métodos de classificação não apresentassem valores minimamente correlacionados, no acto de reconhecimento da unidade de acção poderíamos ser confrontados com problemas sérios de conflito.

No caso concreto, como se pode avaliar pela Tabela 5.17 e pela Tabela 5.18 os resultados são bastante equiparáveis.

Tabela 5.17 - Resumo comparativo (100%/100%)

	Treino e teste na globalidade dos dados								
Olho Direito	100,0%	79,6%	99,0%	60,6%	67,1%	100,0%	100,0%	51,6%	89,0%
Olho Esquerdo	100,0%	80,4%	98,3%	59,7%	65,4%	99,8%	100,0%	52,7%	89,5%

Tabela 5.18 - Resumo comparativo (70%/30%)

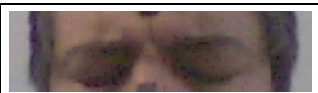
	Treino com 70% e teste 30% dos dados								
Olho Direito	81,9%	76,4%	77,5%	42,8%	58,7%	76,8%	80,1%	57,6%	80,8%
Olho Esquerdo	82,3%	70,7%	76,0%	38,7%	62,0%	72,7%	80,4%	58,3%	78,6%

A correlação estatística entre os resultados obtidos é de 99,25%, o que demonstra muito bom para este efeito.

5.6.2.2 Classificações do cenho

No respeitante ao cenho²⁷ apenas nos parece interessante identificar a AU 4.

Tabela 5.19 - Utilizador modelo, au 4

4	Baixar das sobrancelhas Brow Lowerer; <i>Corrugator supercilii</i> , <i>Depressor supercilii</i>	
---	---	---

Ainda que seja uma unidade de acção combinada, onde entram os olhos e as sobrancelhas, podemos indicar tal unidade apenas pela observação da zona entre as sobrancelhas uma vez que não estamos a tentar distinguir a AU4 da AU9.

Tabela 5.20 - Classificação do Cenho

Classificador	Treino e Teste 100% dos dados		Treino 70%/Teste 30%		
	Casos	Acerto	Treino	Teste	Acerto
Boost	451	97,8%	316	135	93,3%
Linear Model (multinormal)	451	100,0%	316	135	79,3%
Random Forest (500)	451	100,0%	316	135	92,6%
Redes neuronais	451	72,1%	316	135	66,7%
Rpart	451	90,9%	316	135	88,9%
Svm - Kernel: Radial Basis	451	97,1%	316	135	94,1%
Svm - kernel:Anova RBF	451	100,0%	316	135	71,9%
Svm - kernel:bessel	451	88,2%	316	135	8,1%
Svm - kernel:laplace	451	93,1%	316	135	92,6%
Svm - kernel:linear	451	100,0%	316	135	88,9%
Svm - Kernel:Polinomial (2)	451	100,0%	316	135	94,1%
Svm - kernel:tangente hiperbolica	451	81,4%	316	135	48,9%

Logicamente estes tipos de unidade de acção dependem bastante do modelo. Basta reflectir sobre a questão do avanço da idade, uma vez que o ser humano tem tendência a desenvolver rugas que podem causar erros de interpretação da expressão.

Como se observa dos resultados de classificação (Tabela 5.20) o resultado mais sólido verifica-se com 94,1% no acerto sobre o teste dum conjunto independente de dados. Aqui os *SVM* de *Kernel* de *Base Radial* e o polinomial de 2º grau, obtêm o mesmo resultado.

²⁷ Apenas a título de curiosidade na pesquisa pelo significado do termo encontramos tantas opiniões díspares que entendemos por bem referir que para efeito deste trabalho se refere ao espaço entre as sobrancelhas acima do nariz fazendo jus à frase "de cenho franzido".

Já no teste sobre o conjunto de treino (usando o conjunto de dados por inteiro), verifica-se que os *SVM* de *Kernel* de *Base Radial* apresentam maior descida face ao teste para um conjunto separado de dados. Isso leva a crer que seja mais sensível devido a um possível sobre-ajustamento que o de *kernel Polinomial*.

Para este caso, como a variável dependente apresenta apenas dois valores distintos, já é possível recorrer às redes neuronais e ao *Boost* usando a interface *Rattle*.

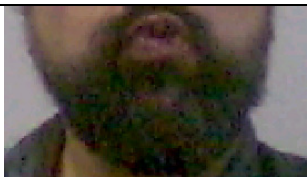
Neste caso as redes neuronais foram uma desilusão na medida em que obtiveram o segundo pior valor de entre todos os classificadores utilizados (72,1% e 66,7%). Ainda sobre as redes, é possível que testando mais modelos se possam melhorar os resultados do classificador, uma vez que inicialmente se parte de um ponto com pesos gerados aleatoriamente.

No caso do *Boost* os resultados obtidos (97,8% e 93,3%) são muito bons, ainda que tenham sido superados pelas máquinas de vector de suporte.

5.6.2.3 Classificações da boca

As unidades de acção escolhidas para efeito de treino foram as seguintes:

Tabela 5.21 - Utilizador modelo, au 18,27 e 28

18	Franzidor de lábios <i>Lip Puckerer; Incisivii labii superioris and Incisivii labii inferioris</i>	
27	Boca aberta <i>Mouth Stretch; Pterygoids, Digastric</i>	
28	Sucção dos lábios <i>Lip Suck; Orbicularis oris</i>	

Uma vez mais estamos perante um caso em que o reconhecimento depende bastante do modelo em causa. Como se ilustra, o nosso modelo é ornado com barba, pelo que muitos dos traços dos lábios ficam ocultos.

Por outro lado, o traçado da própria barba dá indicações preciosas aos classificadores do que o utilizador está a realizar, pois define linhas facilmente reconhecíveis.

Como se observa (Tabela 5.22) de entre os diferentes classificadores o que apresenta melhor desempenho ao nível do teste num conjunto de dados independente dos de treino é o classificador baseado em máquina de vectores de suporte com uma taxa de acerto da ordem dos

96,3%. Este classificador utiliza um *kernel* polinomial de 2º grau (o de primeiro grau é idêntico ao de *kernel*/linear).

Tabela 5.22 – Resultados de classificação da boca

Classificador	Treino e Teste 100% dos dados		Treino 70%/Teste 30%		
	Casos	Acerto	Treino	Teste	Acerto
Linear Model (multinormal)	900	100,0%	630	270	76,3%
Random Forest (500)	900	94,3%	630	270	94,1%
Rpart	900	89,3%	630	270	80,0%
Svm - kernel:Anova RBF	900	100,0%	630	270	91,9%
Svm - kernel:bessel	900	35,8%	630	270	20,4%
Svm - kernel:laplace	900	91,2%	630	270	85,9%
Svm - kernel:linear	900	100,0%	630	270	93,3%
Svm - Kernel:Polinomial (2)	900	100,0%	630	270	96,3%
Svm - kernel:tangente hiperbolica	900	60,8%	630	270	66,3%
Svm Kernel: Radial Basis	900	97,3%	630	270	94,1%

Em termos de melhor consistência temos o *Random Forest* que pouca variação introduz nos resultados mantendo-se em ambos os casos muito próximo dos 94%.

O modelo linear, como se demonstra sofre de algum sobre-ajustamento obtendo por isso uma taxa de acerto de 100% quando aplicado aos dados de treino, mas aplicado a um conjunto de dados distinto do treino decai para 76,3%.

O pior resultado verifica-se com as máquinas de vectores de suporte quando aplicadas com um *kernel* de *Bessel*.

5.6.2.4 Classificações pelos ângulos

Ainda para enriquecimento numa possível linguagem de comando, utilizamos as unidades de acção correspondentes ao deslocamento da cabeça. Essas pela sua natureza são extraíveis directamente dos ângulos que a cabeça apresenta num dado momento.

Nestes casos, o que se está a reconhecer em vez de serem expressões faciais, são antes posicionamentos da cabeça.

Estes são igualmente elementos bastante eloquentes se tivermos em linha de conta que mesmo um utente no seu discurso normal os acaba por utilizar consciente ou inconscientemente.

Tabela 5.23 - Utilizador modelo - au 55, 56, 51 e 52

55	Inclinação à esquerda <i>Head tilt left;</i>	
56	Inclinação à direita <i>Head tilt right;</i>	
51	Cabeça voltada à esquerda <i>Head turn left;</i>	
52	Cabeça voltada à direita <i>Head turn right;</i>	

Por exemplo o segmento de frase "... aí à esquerda ...", para dar mais ênfase, é muitas vezes acompanhada pela AU 55, 51 ou ainda 61.

O erro de cálculo dos ângulos da cabeça (Tabela 5.3) determina directamente o erro de avaliação de cada uma destas AUs.

5.6.3 Classificação da expressão

Tendo por base a classificação individual dos diferentes elementos expressivos, chegamos ao ponto onde se torna necessário retirar conclusões sobre a expressão do utilizador. Esta tanto poderá ser um vaticínio sobre as Unidades de Acção²⁸, que por sua vez poderão ser submetidas a um sistema seguinte, quer directamente para identificar termos duma linguagem rudimentar de comando.

²⁸ Sistema FACS

No caso da nossa linguagem exemplo de comando (ver em 2.4), pretendíamos identificar cinco comandos simples: virar à esquerda, virar à direita, deslocar para a frente, parar, iniciar ou terminar aceitação de comandos.

No caso desta linguagem é bastante simples de construir uma árvore que implemente o diagrama de decisões necessário para distinguir cada um dos comandos.

À semelhança de todas as linguagens há combinações de termos que não fazem sentido e que, como tal, não tem acção associada (assinalada no nosso esquema pelo ponto de interrogação encapsulado no circulo vermelho).

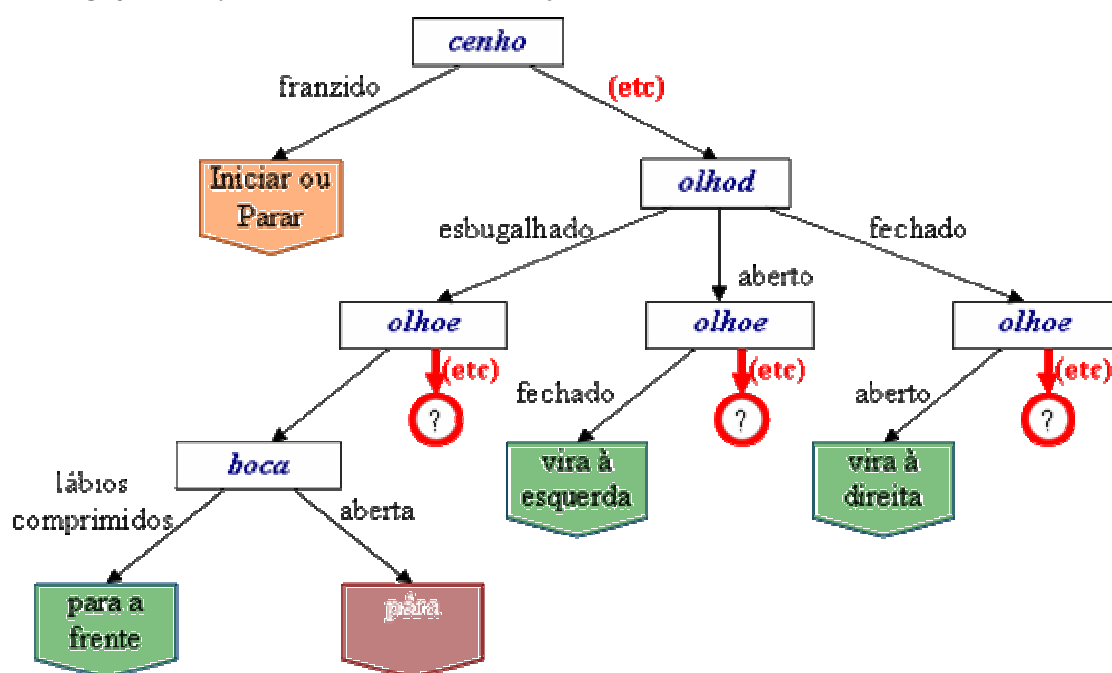


Figura 5.24 - Exemplo de árvore de decisão

No exemplo apresentado desenhámos uma árvore para tomar decisões sobre as acções a realizar (Figura 5.24) e não propriamente sobre a expressão (que neste caso está implícita).

Para este caso concreto, confiança depositada em cada um dos comandos, baseada nas taxas de acerto de cada componente (apresentadas anteriormente) seria a que se pode observar na . A confiança numa correcta identificação de um comando decresce de forma inversamente proporcional ao tamanho do percurso da árvore para o identificar.

Poderíamos por isso pensar em desenhar outras árvores com a mesma capacidade final de discernimento sobre os comandos mas com um nível de confiança maior.

Para efeito de alargamento do *vocabulário* para construção de frases de comando seria possível criar uma lista exaustiva de classificação de *ALUs* de acordo com o FACS, e depois, entregar a tarefa do reconhecimento frásico final à interface *Intelwheels* que já incorpora um configurador para criar uma linguagem adaptável ao utilizador.

Tabela 5.24 - Confiança esperada num comando

Acção	Confiança
Iniciar/Parar de aceitar comandos	94,1%
Virar à esquerda	$94,1\% \times 81,9\% \times 82,3\% = 63,4\%$
Virar à direita	$94,1\% \times 81,9\% \times 82,3\% = 63,4\%$
Para a frente	$94,1\% \times 81,9\% \times 82,3\% \times 96,3\% = 60,5\%$
Pára	$94,1\% \times 81,9\% \times 82,3\% \times 96,3\% = 60,5\%$

Desta forma a fase final de conversão das classificações em comandos e respectivas sequências fica a cargo dessa própria interface e não como parte integrante do subsistema de visão.

5.7 Perspectiva global do sistema de visão

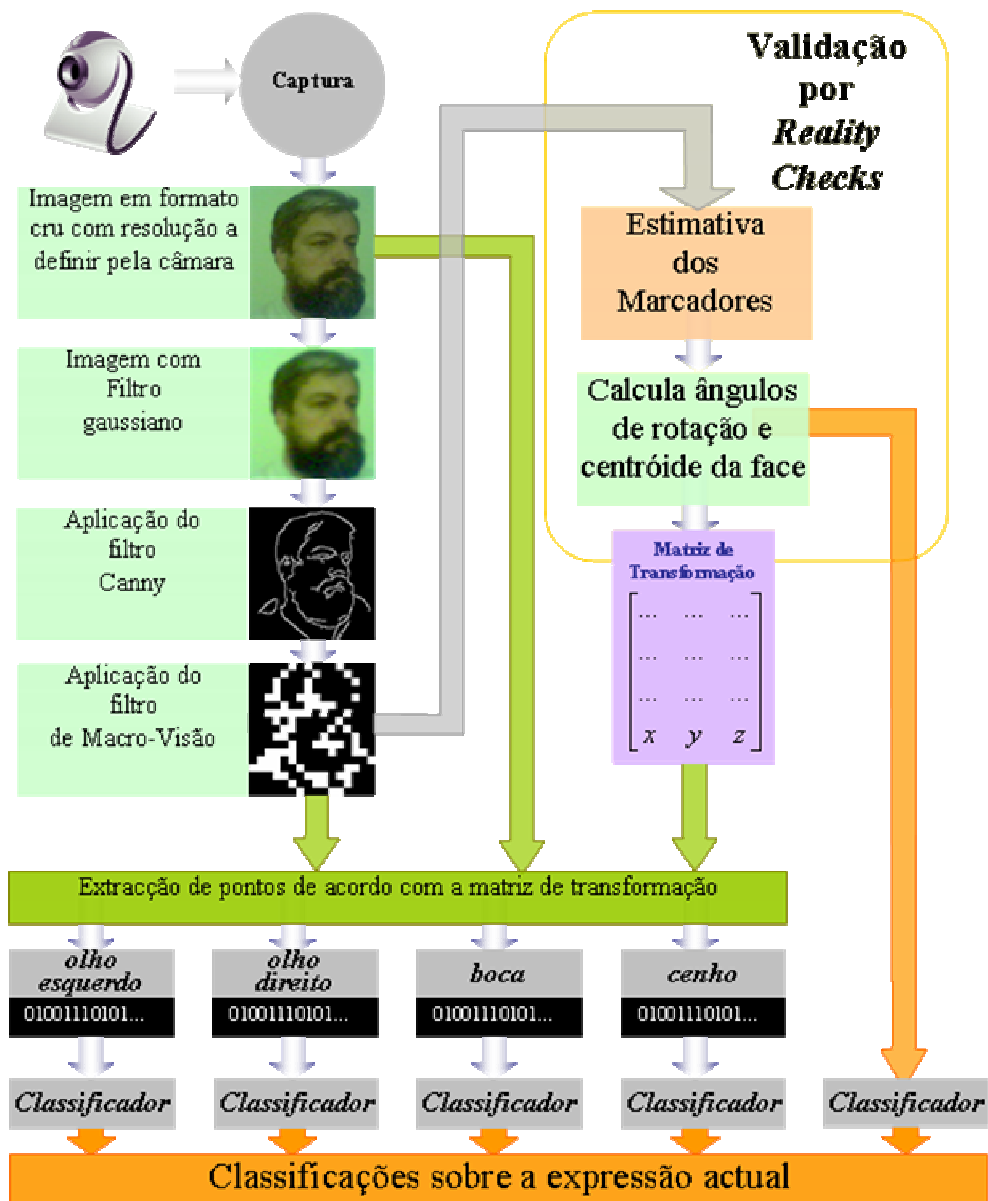


Figura 5.25 - Arquitetura global do sistema de visão

O sistema de visão, como já foi referido, destina-se à integração com o projecto *Intellwheels*. Numa primeira linha de preocupação é objectivo do sistema reconhecer expressões faciais, o que foi descrito durante este capítulo.

Numa segunda linha de preocupação, o sistema tem por objectivo ser de fácil implementação e de resposta rápida. Para corresponder melhor a esta directiva, tal como se ilustra na Figura 5.25, a estrutura do sistema define um fluxo principal sequencial que em

determinados pontos se ramifica para permitir fios de execução paralelos (permitindo tirar partido por exemplo dos sistemas actuais *multi-core*).

5.8 Conclusões

Expuseram-se neste capítulo alguns pressupostos base que serviram de linha orientadora sobre o mundo envolvente no respeitante à aquisição de imagem para efeitos de processamento.

Com esses pressupostos e aproveitando parte de estudos medievais o homem, criou-se uma máscara aplicável ao modelo de utilizador que serve para a recolha de elementos a tratar posteriormente para aprendizagem computacional.

Expôs-se a forma como se aplica a máscara em termos de seguimento das poses do utilizador, em tempo real, e como se identificam por essa via as área da imagem recolhida para estudo.

Apresentou-se o conceito a que se chamou de "Macrovisão" tendo por objectivo a simplificação dos dados que são alvo tratamento, tanto para a fase de aprendizagem como posteriormente para a fase de reconhecimento em tempo real.

Explicou-se a forma como se procedeu para auxiliar o sistema de recolha de dados de treino através do uso de marcadores. Neste tópico explicou-se como através da identificação dos marcadores na fase de treino, utilizando algumas métricas simples por análise da distância entre eles, se consegue ajustar a máscara (descrita no capítulo anterior) a diferentes formas de rosto através de factores de normalização.

Aplicaram-se metodologias de regressão sobre os dados recolhidos de macrovisão (da imagem como um todo) a fim de verificar as taxas de acerto na estimativa da localização desses marcadores, quando eles não estão presentes e reflectiu-se sobre os resultados apresentados, para as coordenadas deles na imagem base, face aos métodos de regressão que a ferramenta de exploração oferece.

De seguida, apresenta-se uma forma de em função das coordenadas bidimensionais dos marcadores, num dado momento, e em função dos factores de normalização, como se podem modelar estimar tridimensionalmente esses marcadores.

Expôs-se de seguida como se aplicou regressão sobre as coordenadas dos marcadores para estimar os ângulos de rotação dos três eixos cujo origem imaginária seria o centróide da cabeça, e apresentam-se os resultado dessa regressão por confrontação com os ângulos calculados por força bruta no acto de criação dos conjuntos de dados base.

Abordou-se a necessidade de implementação da implementação de "Reality Checks" e definiu-se a forma como aumentar a robustez do sistema, dadas as estimativas, por introdução do conceito de rasto. Nesse aspecto apresentou-se a eficácia do rasto pela avaliação da qualidade da

estimativa da posição subsequente durante o seguimento da posição dos marcadores no rosto do utilizador. Indicaram-se as tabelas que demonstram a degradação da confiança de estimativas sucessivas pelo rasto em caso de falha em tempo real do processamento do sistema de visão na estimativa do posicionamento dos marcadores por interpretação da imagem.

Explicou-se a fase de classificação dos dados por meio do uso de um assistente de classificação que por activação de ícones define o valor de cada uma das variáveis objectivo para os diferentes conjuntos de dados (*datasets*).

Apresentaram-se as diferentes aplicações dos métodos de classificação disponíveis na ferramenta base sobre os diferentes elementos expressivos em análise e estudaram-se os resultados obtidos, apontando-se os métodos que para cada um dos casos melhor lidam com os conjuntos de dados.

Analisou-se o grau de confiança atribuível à nossa proposta exemplo de linguagem de controlo (capítulo 2), em função dos resultados obtidos nas tabelas anteriores.

Explicou-se a forma como na implementação do sistema se pode delegar a fase de construção da linguagem de comando de forma adaptada ao utilizador tirando partido da interface já existente no projecto intellwheels.

Resumiu-se a arquitectura do sistema através de um diagrama de blocos que ilustra os principais fluxos de tratamento de dados, dando assim uma ideia clara sobre o sistema proposto.

Todos estes elementos estudados são utilizados para o tecer das conclusões finais apresentada no capítulo seguinte.

Capítulo 6

6. Conclusão

6.1 Resultados principais

O módulo do sistema de visão, encontra-se preparado para poder dar resposta aos diferentes desafios a que se propôs.

Numa primeira vertente, permite através das metodologias anteriormente explicadas realizar um processamento inteligente da imagem para enquadramento das respectivas percepções no mundo real, aumentando a robustez do processamento.

Numa segunda linha, concebeu-se um sistema que permite a aplicação a diferentes condições de luz e disposições do utilizador, de forma a imunizá-lo às mudanças decorrentes do meio ambiente. Este sistema permite ainda imunizar o reconhecimento de características do rosto dos elementos adicionais do fundo, na medida em que orienta de acordo com uma máscara.

Relativamente ao utilizador, a criação de uma estrutura baseada em modelos preconcebidos e activáveis de acordo com o que melhor se adapta ao utilizador, permite libertar o sistema de visão de restrições que advém de sexo, idade, raça bem como hábitos e costumes que alterem o aspecto/apresentação do rosto do utilizador.

Com a introdução do modelo de máscaras adaptadas ao modelo de utilizador que corresponde melhor às necessidades do mesmo, conseguem-se modelar características dimorfas do próprio utilizador, resultantes de acidentes traumáticos, sem alteração do próprio sistema mas sim, e apenas, por configuração do modelo de utilizador.

Em termos de processamento o recurso a métodos computacionais eficientes, eleva a capacidade de resposta do sistema de forma a tornar o processamento de imagem mais fluído aumentado dessa forma a qualidade do reconhecimento.

6.2 Trabalho futuro

Esta fase de desenvolvimento do sistema de visão tem por objectivo último provar a sua utilidade como contributo positivo para um sistema da natureza do projecto *Intellwheels*. Por este motivo não se trata de um esforço no sentido de treinar múltiplos modelos de utilizador com bases de dados alargadas para tornar a capacidade de reconhecimento universal. Antes pelo contrário o objectivo é apenas o de provar, que com um utilizador específico o sistema funciona correctamente.

No que toca aos modelos de utilizador, apenas se utiliza de momento o do próprio autor. Como nos parece óbvio, a preponderância de utilizadores masculinos, de barba farta, de penteado indefinível na casa dos 40, não nos parece ser suficiente para que o modelo utilizado sirva de base prática para uma aplicação alargada como produto final. Por este motivo é necessário alargar a base de modelos acrescento a maior quantidade possível de modelos base. Isto permitirá ao sistema uma resposta mais acertada independente do utilizador uma vez que passaria a dispor de redes treinadas para características mais compatíveis do que as do modelo único.

A introdução de modelos com idades diferentes, de diferentes sexos, de etnias distintas e mesmo em estados de apresentação diferentes, é o meio de se transformar o sistema de visão num sistema universal. Se dotarmos o sistema da capacidade de procurar na sua base de modelos o que mais se aproxima do utilizador, então para além de o tornarmos universal tornamo-lo também automático, dispensando a fase de configuração do utilizador.

Numa fase posterior, podem-se alargar os conjuntos de dados de forma a aumentar o grau de confiança dos modelos gerados, tanto para a regressão como para a classificação.

A melhoria das interfaces do próprio módulo de visão é um dos pontos de aperfeiçoamento do trabalho. Em particular a fase de configuração inicial, requer bastantes acréscimos sob a forma de interfaces de uso amigável.

O refinamento dos algoritmos de filtragem de forma a optimizá-los para um tempo de execução menor poderá sempre ser um dos pontos de trabalho futuro.

A aplicação de uma câmara adicional poderia ainda ser um dos pontos de exploração para expansão futura pela introdução de visão estéreo no sistema, com todas as transformações que exige ao sistema de visão mono-câmara.

6.3 Comentários finais

Embora existam diversos pontos possíveis de melhoria deste trabalho, como se referiu previamente, julgamos ter conseguido, na essência, os propósitos a que nos propusemos com a realização do mesmo.

Contudo, ao longo de qualquer projecto onde o investigador se empenhe, nascem constantemente novas linhas de orientação que permitem o alargamento do trabalho em sentidos não inicialmente previstos.

Não é de todo raro, que no momento de rematar um trabalho, pelas limitações de tempo ou de cumprimento de objectivos, tenhamos a amarga sensação de que seria óptimo poder recomeçar o projecto. Não devido a não se terem alcançado, ou até superado os resultados esperados à partida mas simplesmente porque se chega ao fim com uma percepção muito diferente da que inicialmente se tinha sobre o projecto. É nesses momentos que se vê com clareza os caminhos que não foram trilhados e por isso fica sempre a sensação de se poder ir mais além.

Contudo, o pragmatismo tem de se impor e no que respeita a este projecto, ainda que nunca pudéssemos dar por concluído o módulo de sistema de visão computacional ideal, damos por concluído este nosso trabalho.

Bibliografia

Referências bibliográficas

- [1] Intelligent Wheelchair with 360° camera sensor, Japan's National Institute of Advanced Industrial Science and Technology [online] Available at http://www.aist.go.jp/aist_j/press_release/pr2006/pr20060920/pr20060920.html, [Consulted on Jul, 2008]
- [2]. Araújo, Paulo Ferreira, 1997, *"Desporto Adaptado no Brasil: Origem, Institucionalização e Actualidade"*, PHD Dissertation, Faculdade de Educação Física, Universidade Estadual de Campinas, pp.9-10.
- [3] Ballard, Dana H. & Brown, Christopher M., 1982, Computer Vision, Prentice Hall Inc, pp. 207-211.
- [4] R. A. M. Braga, M. Petry, A. P. Moreira, and L. P. Reis, "INTELLWHEELS - A Development Platform for Intelligent Wheelchairs for Disabled People," in *5th International Conference on Informatics in Control, Automation and Robotics*, vol. I, Funchal, Madeira, Portugal, 2008, pp. 115-121.
- [5] Breiman, Leo, 2001. "Random Forests". Machine Learning 45 (1): pp. 5–32.
- [6] História da cadeira de rodas. [Online]. Available at: http://www.cadeirarodas.com/historia_da_cadeira_de_rodas.html [Consulted on Jun 2008]
- [7] Chairdex, History of Wheelchairs [Online] Available at <http://www.chairdex.com/history.htm> [Consulted on Jul 2008].
- [8] Cortes, C. and Vapnik, V., 1995, Support Vector Machines. Machine Learning 20(3): pp 273-297.
- [9] Cover, T. M. and Hart, P.E., Nearest Neighbor Pattern Classification, IEEE Transaction on Information Theory, Vol. IT-13, n.1, Jan.1967, pp. 21-27.
- [10] Cyberlearning, *Franklin Delano Roosevelt Memorial* [online] Available at <http://www.cyberlearning-world.com/nhhs/compapps/workshop/dc/fdrmem.htm>, [Consulted on Jul, 2008]
- [11] Carnegie Mellon -. School of computer science, *FACS - Facial Action Coding System* [online] Available at <http://www.cs.cmu.edu/~face/facs.htm> [Consulted on Feb 2008],
- [12] Ekman, P. & Friesen, W. V. (1977). *Manual for the Facial Action Coding System*. Palo Alto, Calif.: Consulting Psychologists Press.

- [13] Fayyad, U.M. and Irani, K.B., 1993, Multi-interval discretization of continuous valued attributes for classification learning, in *Proc. Thirteenth International Joint Conference on Artificial Intelligence*, Chambéry, France, San Francisco. Morgan Kaufmann, pp. 1022-1027.
- [14] Hamagami, T.; Hirata, H. Development of intelligent wheelchair acquiring autonomous, cooperative, and collaborative behavior. *IEEE International Conference on Systems, Man and Cybernetics*. 10-13 de Oct. de 2004, Vol. 4, pp. 3525 - 3530 .
- [15] Han, Jiawei and Kamber, Micheline, 2001, Data Mining. Morgan Kaufmann Publishers, pp. 279/281.
- [16] Han, Jiawei and Kamber, Micheline, 2001, Data Mining. Morgan Kaufmann Publishers, pp. 319/322.
- [17] Hearn, Donald & Baker, M. Pauline, 1997, Computer Graphics. Prentice-Hall International Inc, pp. 408-422.
- [18] Kanagawa Institute of Technology – Department of Welfare System Engineering, *Wearable Power Assist Suit - 1st Stand Alone Suit* (2002) [online], available at: http://www.we.kanagawa-it.ac.jp/~yamamoto_lab/pas/index.htm, [Consulted on Jul, 2008]
- [19] Kenji Hashimoto, Yusuke Sugahara, Hun-Ok Lim, 2008, *Biped Landing Pattern Modification Method and Walking Experiments in Outdoor Environment*, in *Journal of Robotics and Mechatronics* Vol.20 No.5, pp.775.
- [20] McCulloch, W.S, and Pitts, W., 1943, *A logical calculus of the ideas immanent in nervous activity*. *Bulletin of Mathematical Biophysics* (5), pp. 115-137.
- [21] M. R. Petry, "Desenvolvimento de um Protótipo e de Metodologias de Controlo de uma Cadeira de Rodas Inteligente," Dissertation for Masters Degree, Departamento de Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, Porto, 2008.
- [22] Pires, Gabriel & Nunes, Urbano, 2002, *A Wheelchair Steered through Voice Commands and Assisted by a Reactive Fuzzy-Logic Controller*, Kluwer Academic Publishers, *Journal of Intelligent and Robotic Systems* 34: pp. 301–314,
- [23] Pride Mobility Products Corp, Jazzy Power Chairs, [online] Available at: <http://www.pridemobility.com/jazzy/models.asp> [Consulted on Jul, 2008]
- [24] Consultor médico do Hospital Policlín, *Paralisia Cerebral* , [online] Available at: <http://www.policlin.com.br/drpoli/130/> [Consulted on Jul, 2008]
- [25] Medgadged – internet journal of emerging medical technologies, [online] Available at: http://medgadged.com/archives/2007/06/pegasus_wheelchair_1.html [Consulted on Jul, 2008]
- [26] Quinlan, J.R., 1986, Induction of Decision Trees. *Machine Learning* 1(1): pp 81-106.

- [27] Quinlan, J. R., 1987, Simplifying decision trees. *International Journal of Man-Machine Studies*, 27: pp. 221-234.
- [28] R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org> [Consulted on Aug, 2008].
- [29] Sá, Joaquim P. Marques, 2001, *Pattern Recognition*, Springer Verlag, pp. 39-41.
- [30] Schalkoff, Robert, 1992, *Pattern Recognition, Statistical, Structural and Neural Approaches*, John Wiley & Sons, Inc, pp 27-28.
- [31] Schalkoff, Robert, 1992, *Pattern Recognition, Statistical, Structural and Neural Approaches*, John Wiley & Sons, Inc, pp 34-35.
- [32] Surgical Artistry Academy Lessons (2008)- Aesthetician Muscles of the Face [online], available at: <http://www.surgerytoday.com/facemuscles/> [consulted on Jul 2008].
- [33] Toyota Co, Toyota Group Pavilion at EXPO 2005 AICHI, Japan [online] available at: http://www.toyota.co.jp/en/news/04/1203_1d.html [Consulted on Jul, 2008]
- [34] Unites Spinal Association, iBOT@4000 Mobility System, (2006) [online], available at: http://www.unitedspinal.org/publications/action/wp-content/Dean_IBot054.jpg [Consulted on Jul, 2008]
- [35] The University of Waikato, Weka 3 - Data Mining with Open Source Machine Learning Software in Java, (2008) [online], available at: <http://www.cs.waikato.ac.nz/ml/weka/> [Consulted on Jul, 2008]
- [36] Williams, Graham (2008). Rattle: A graphical user interface for data mining in R using GTK. R package version 2.4.10, [Online] Available at: <http://rattle.togaware.com/> [Consulted on Aug, 2008]
- [37] Witten, Ian H. and Frank, Eibe, 2000, *Data mining*, Academic Press, pp. 254-258.
- [38] Pires G., Araújo R., Nunes U., Coimbra J., 2001, "Development of the RobChair Assistive Navigation System", *Revista Robótica: Automação, Controlo e Instrumentação*, n.º 43, 2º Trimestre 2001, pp. 25-28.

Outra bibliografia

Azuna, R.; Baillot, Yohan; Behringer, Reinhold; Feiner, Steven; Julier, Simon; MacIntyre, Blair, "Recent Advances in Augmented Reality," *IEEE Computer Graphics and Applications*, vol. 21, no. 6, pp. 34-37, Nov. 2001.

- Bourhis, G. and Agostini, Y., "The Vahm Robotized Wheelchair: System Architecture and Human-Machine Interaction," *Journal of Intelligent and Robotic Systems*, vol. 22, no. 1, pp. 39-50, May 1998.
- Braga, R. A. M.; Petry, M. R.; Moreira, A. P.; and Reis, L. P., "Platform for intelligent wheelchairs using multi-level control and probabilistic motion model," in *8th Portuguese Conference on Automatic Control, Controlo 2008*, Vila Real, Portugal, 2008.
- Braga, R. A. M.; Petry, M. R.; Reis, L. P., and Oliveira, E., "Multi-Level Control of an Intelligent Wheelchair in a Hospital Environment using a Cyber-Mouse Simulation System," in *5th International Conference on Informatics in Control, Automation and Robotics*, Funchal, Madeira, Portugal, 2008
- Carpin, S.; Lewis, M.; Wang, J., Balakirsky, S.; Scrapper, C., "USARSim: a robot simulator for research and education," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 1400-1405.
- Cohen, Philip R.; Johnston, Michael; McGee, David; Oviatt, Sharon; Pittman, Jay; Smith, Ira; Liang Chen and Clow, Josh, "QuickSet: multimodal interaction for distributed applications," in *Proceedings of the Fifth ACM international Conference on Multimedia*, Seattle, Washington, United State, 1997, pp. 31-40.
- Faria, P. M., Braga, R. A. M., Valgôde, E., and Reis, L. P., "Interface framework to drive an intelligent wheelchair using facial expressions," in *IEEE International Symposium on Industrial Electronics*, 2007.
- Faria, P. M., Braga, R. A. M., Valgôde, E., and Reis, L. P., "Platform to Drive an Intelligent Wheelchair using Facial Expressions," in *Proceedings 9th International Conference on Enterprise Information Systems - Human-Computer Interaction (ICEIS 2007)*, Funchal, Madeira, 2007, pp. 164-169.
- Gerkey, B., Vaughan, R., and Howard, A., "The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems," in *International Conference on Advanced Robotics*, Coimbra, Portugal, 2003, pp. 317-323.
- Johnston, M., Bangalore, S., and Vasireddy, G., "MATCH: multimodal access to city help," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2001
- Martins, B., Valgôde, E., Faria, P., and Reis, L. P., "Multimedia Interface with an Intelligent Wheelchair," in *Proc. of ComplImage 2006 -Computational Modelling of Objects Represented in Images: Fundamentals Methods and Applications*, Coimbra, Portugal, 2006, pp. 267-274.
- Niniss, ,H. and Nadif, A., "Simulation of the behaviour of a powered wheelchair using virtual reality," in *Proc. 3rd Intl Conf. Disability, Virtual Reality & Assoc. Tech*, Alghero, Italy, 2000.

Röfer, T., "Strategies for Using a Simulation in the Development of the Bremen Autonomous Wheelchair," in *Proceedings of the 12th European Simulation Multiconference on Simulation - Past, Present and Future*, Manchester, United Kingdom, Junho, 1998

Zandifar, A. and Chahine, A., "A Video Based Interface to Textual Information for the Visually Impaired" in *Proceedings of the 4th IEEE international Conference on Multimodal interfaces*, 2002.

Zudilova E. V., P. M. A. Slood, and R. G. Belleman, "A multi-modal interface for an interactive simulated vascular reconstruction system," in *Proceedings of Fourth IEEE International Conference on Multimodal Interfaces*, 14-16 October 2002, Pittsburgh, PA, USA