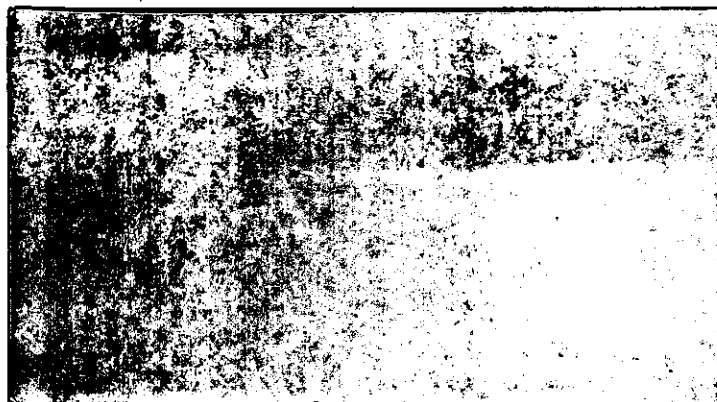




DEPARTAMENTO DE
ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES



FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Rua dos Bragas, 4099 Porto Codex – PORTUGAL

10.285
R.J.

E.M.V.

PAULA MARIA MARQUES DE MOURA GOMES VIANA

Equipamento Terminal para
Acesso Remoto a
Informação Fotográfica

UNIVERSIDADE DO PORTO
Faculdade de Engenharia
BIBLIOTECA M
N.º 20621-08
CDU 681.3(043)
Data 2/6/95

N.º 34812

Dissertação submetida para satisfação parcial dos requisitos do programa do curso de
Mestrado em Engenharia Electrotécnica e de Computadores
(Área de especialização em Sistemas)

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Abril de 1994

043 M

V6682

2X.2

Dissertação realizada sob a supervisão de
Prof. Doutor Eurico Manuel Elias de Moraes Carrapatoso,
Professor Auxiliar do Departamento de Engenharia Electrotécnica e de Computadores
da Faculdade de Engenharia da Universidade do Porto

Dissertação realizada sob a supervisão de
Prof. Doutor Eurico Manuel Elias de Morais Carrapatoso,
Professor Auxiliar do Departamento de Engenharia Electrotécnica e de Computadores
da Faculdade de Engenharia da Universidade do Porto

Aos meus pais,
ao Vasco e à Ana

Resumo

Pode dizer-se que a necessidade de partilha e acesso remoto a informação multimédia e a evolução das infra-estruturas de comunicação ditam a introdução de inovações nos serviços públicos de telecomunicações. Ambientes integrados de vídeo, audio, imagem e trabalho cooperativo são exemplos de aplicações a disponibilizar e para as quais existe já procura. Neste contexto, apresenta-se um sistema que, aproveitando as potencialidades da Rede Digital com Integração de Serviços, que começou recentemente a ser disponibilizada em Portugal, pretende fornecer um novo serviço que permite o acesso a servidores nacionais e internacionais com informação fotográfica, mas mantendo compatibilidade com o videotex alfamosaico existente.

Este sistema usa como equipamento terminal um computador pessoal equipado com uma interface de rede e módulos de *software* independentes para tratamento de dados nos formatos gráfico e fotográfico. A opção por esta arquitectura modular permitirá, de uma forma simples, a futura integração no terminal de novas facilidades como o audio ou a transferência transparentes de dados.

A nível europeu, a compatibilidade do serviço de videotex alfafotográfico é assegurada pela utilização de protocolos de comunicação e de tratamento de dados, normalizados. De forma a tornar aceitáveis os tempos de transmissão e a quantidade de memória requisitada para armazenar uma imagem no servidor, recorre-se a algoritmos de compressão que, juntamente com uma sintaxe própria, definem a forma como a imagem será descodificada e apresentada ao utilizador.

Especifica-se uma aplicação que, embora implicando uma perda de compatibilidade com serviços fornecidos por outros países, permite aumentar a eficiência e a qualidade no caso de terminais com capacidade gráfica reduzida ao introduzir no algoritmo de compressão técnicas de quantificação de cor.

Palavras chave: videotex, RDIS, compressão de imagem, JPEG, sintaxe fotográfica, quantificação de cor.

Abstract

In this thesis a multimedia service is presented together with the terminal requested to access it. This service is one of the first to offer remote access to various kinds of information through the Integrated Service Digital Network. The terminal, specified and developed with the support of the Portuguese telecommunications operators, is based on a personal computer, enhanced by a network interface, and a modular software package that deals with image decompression and presentation. This modular architecture will render the introduction of new media easy, allowing the upgrade of the application and making it even more attractive from a commercial perspective. From the outset, it was decided that the new system, although offering a new service, should maintain compatibility with the traditional alphamosaic videotex.

Interworking between different European photovideotex services was also considered to be paramount and this compatibility is guaranteed by the use of standards that define the communication protocols and the data syntaxes. To make it acceptable, from the point of view of transmission delay and server capacity, compression algorithms were implemented and optimized keeping in mind the quality of service that should be provided.

A new photographic profile based on colour quantisation was specified. Although not conforming to standards, it increases the performance on low cost and reduced graphic capabilities terminals.

Keywords: videotex, ISDN, image compression, JPEG, photographic syntax, colour quantisation.

Índice

1	Introdução	1
2	Enquadramento	5
2.1	Infra-estruturas de Comunicações	5
2.2	Serviço Público Multimédia	7
2.3	O Novo Serviço de Videotex	8
3	Especificação do Equipamento Terminal	11
3.1	API — <i>Application Program Interface</i>	12
3.1.1	Esquema de Comunicação com a API	13
3.2	Protocolos de Comunicação	15
3.2.1	NAF — <i>Network Access Facility</i>	15
3.2.2	Aplicação de Comunicação	16
3.3	Aplicação Local	21
3.3.1	TFI — <i>Terminal Facility Identifier</i>	22
3.3.2	Interface com o Utilizador	22
4	Fundamentos de Compressão de Imagem	25
4.1	Codificação por Predição	28
4.2	Codificação por Transformadas	29

4.2.1	Transformada de Karhunen-Loeve	31
4.2.2	Transformada de Cosenos	32
4.3	Quantificação	34
4.4	Códigos de Comprimento Variável	36
4.4.1	Codificação de Huffman	39
4.4.2	Codificação Aritmética	40
4.5	Quantificação Vectorial	42
4.6	Codificação por Sub-Bandas	46
5	Aplicação Fotográfica	49
5.1	Algoritmo JPEG	49
5.1.1	Método baseado na DCT	50
5.1.2	Método de Codificação por Predição	53
5.1.3	Método DCT Progressivo	54
5.1.4	Método Hierárquico	57
5.1.5	Tratamento de Imagens com Múltiplas Componentes	58
5.2	Sintaxe Fotográfica	59
5.2.1	Adaptação de Resoluções entre as Várias Classes	60
5.3	Técnicas de Implementação	61
5.4	Especificação de uma Aplicação Privada	68
5.4.1	Quantificação de Cor	70
5.4.2	Ordenação da Tabela de Cores	74
5.4.3	Algoritmo de Compressão	77
5.5	Análise Subjectiva de Qualidade	77
6	Conclusões	79

Lista de Figuras

2.1	Configuração actual do serviço de videotex	7
2.2	Sintaxes de videotex a)mosaico b)geométrico c)fotográfico	8
2.3	Configuração do futuro serviço de videotex	9
3.1	Modelo do Equipamento Terminal de Videotex	11
3.2	Problema actual de incompatibilidade entre aplicações e interfaces RDIS . . .	12
3.3	Estrutura de filas de espera entre a API e a Aplicação	14
3.4	Pilha protocolar do serviço de videotex	15
3.5	Procedimentos de sinalização entre um terminal e um servidor de videotex RDIS	19
3.6	Fluxograma dos serviços de nível aplicação Sbv_Tfi e Sbv_Dfk	20
3.7	Máquina de estados de um terminal	21
3.8	Esquema da interface de utilizador do terminal de videotex alfafotográfico . .	23
3.9	Interface de utilizador do terminal de videotex alfafotográfico	23
4.1	Informação redundante e irrelevante	26
4.2	Desempenho de um algoritmo	27
4.3	Sistema DPCM	29
4.4	Exemplo da decorrelação efectuada por uma transformada	30
4.5	Vectores Base para uma DCT de 8 elementos	33
4.6	Quantificador a) uniforme b) não uniforme	35

4.7	Exemplo de um codificador aritmético	42
4.8	Diagrama de blocos de um quantificador vectorial	44
4.9	Decomposição de uma imagem em bandas de frequências, sub-amostragem e reconstrução	47
5.1	Método de compressão baseado na DCT	50
5.2	Ordenação dos coeficientes em <i>zig-zag</i>	52
5.3	Método de compressão por predição	53
5.4	Vizinhança de uma amostra para o cálculo da estimativa no método de com- pressão por predição	54
5.5	Método de compressão progressivo por a) selecção espectral e b) aproximações sucessivas	55
5.6	Etapas intermédias de apresentação de uma imagem construída no modo pro- gressivo por selecção espectral com 4 bandas	56
5.7	Versão final da imagem construída nos modos progressivo por selecção espectral e sequencial	57
5.8	Exemplo de multiplexagem e ordenação das componentes de uma imagem . .	58
5.9	Definição de uma área segundo coordenadas normalizadas	61
5.10	Fluxograma da transformada rápida de cosenos	63
5.11	Simplificação do fluxograma da transformada rápida de cosenos quando os coeficientes 5, 6 e 7 são nulos	63
5.12	Sintaxe fotográfica: a) exemplo de colocação parcial de uma imagem na janela definida pela sintaxe e b) selecção dos blocos a descomprimir	64
5.13	Tamanhos relativos de uma imagem e da imagem DC correspondente	65
5.14	Imagem descomprimida para a resolução original de 640x480 pixels	66
5.15	Imagem DC correspondente	66
5.16	Exemplo da criação de um menu auxiliar utilizando imagens DC	67
5.17	Imagem comprimida no formato 2:1:1 e apresentada num terminal 4:2:2 . . .	69

Lista de Tabelas

3.1	Comandos de Videotex	18
3.2	Exemplo de codificação de um comando telemático	18
4.1	Códigos de comprimento variável	38
4.2	Probabilidades e intervalos atribuídos no processo de codificação aritmética .	42
5.1	Preditores para o método de compressão sem perdas	54
5.2	Filtros associados à propagação do erro às amostras vizinhas para a operação de <i>dithering</i>	74

Lista de Abreviaturas

AC	Aplicação de Comunicação
AL	Aplicação Local
API	<i>Application Program Interface</i>
CAPTAIN	<i>Character and Pattern Telephone Access Information Network system</i>
CCIR	<i>Comité Consultatif International des Radiocommunications</i>
CCITT	<i>Comité Consultatif International pour le Télégraphe et le Téléphone</i>
CEPT	<i>Conférence Européene des Postes et Télécommunications</i>
CIF	<i>Common Interchange Format</i>
DCT	<i>Discrete Cosine Transform</i>
DPCM	<i>Differential Pulse Code Modulation</i>
DRCS	<i>Dynamically Redefinable Character Set</i>
ETSI	<i>European Telecommunications Standards Institute</i>
FIFO	<i>First-In-First-Out</i>
IDCT	<i>Inverse Discrete Cosine Transform</i>
ISO	<i>International Standards Organization</i>
JPEG	<i>Joint Photographic Experts Group</i>
KLT	<i>Karhunen-Loeve Transform</i>
LAPB	<i>Link Access Protocol, Balanced</i>
LAPD	<i>Link Access Protocol, D channel</i>
MIPS	<i>Millions of Instructions per Second</i>
MOS	<i>Mean Opinion Score</i>
MSE	<i>Mean Square Error</i>
NAF	<i>Network Access Facility</i>
NAPLPS	<i>North American Presentation Level Protocol Syntax</i>
NUF	<i>Network User Facility</i>
OSI	<i>Open Systems Interconnection</i>
PAD	<i>Packet Assembly-Disassembly</i>
PC	<i>Personal Computer</i>
PCI	<i>Programmable Communication Interface</i>
PCM	<i>Pulse Code Modulation</i>

QoS	<i>Quality of Service</i>
RDIS	Rede Digital com Integração de Serviços
TFI	<i>Terminal Facility Identifier</i>
VQ	<i>Vector Quantization</i>
VTX	Videotex

Capítulo 1

Introdução

O progresso verificado nas redes de comunicação e a integração dos computadores e das tecnologias de telecomunicações possibilitaram a introdução de novos serviços que, por sua vez, viabilizassem os investimentos para a instalação dos novos meios de transmissão e permitissem a satisfação das exigências crescentes na partilha de informação normalmente designada por multimédia (texto, gráficos, audio, imagem, etc.).

O acesso a informação multimédia estava até agora limitado a soluções locais ou proprietárias, o que limitava os utilizadores destas tecnologias a campos de actividade muito restritos.

Surgiu então a ideia de desenvolver um sistema baseado no serviço de videotex, capaz de, numa primeira fase, permitir o acesso remoto a bases de dados contendo informação gráfica e imagens de qualidade fotográfica tendo como suporte físico a Rede Digital com Integração de Serviços (RDIS) e que posteriormente poderia ser melhorado pela introdução de novos módulos. Pretendia-se utilizar as potencialidades da RDIS, que representam uma melhoria significativa relativamente aos sistemas de telecomunicações tradicionais, oferecendo um serviço inovador mas mantendo compatibilidade com o serviço de videotex alfamosaico já existente. A completa integração com o antigo serviço de videotex alfamosaico seria obtida pela interligação das várias redes através de *Pontos de Acesso de Videotex*. Durante o estabelecimento da chamada o servidor interroga o terminal sobre as facilidades que tem implementadas sendo assim possível a distinção entre os vários tipos de terminais. Foi esta ideia que levou à realização do trabalho que conduziu a esta dissertação.

A introdução deste novo serviço e o desenvolvimento do equipamento terminal indispensável foram apoiados e incentivados pelas operadoras de telecomunicações nacionais de forma a viabilizar a implantação da RDIS em Portugal e porque não existia no mercado internacional um produto que permitisse satisfazer as necessidades.

Mas, apesar da RDIS vir aumentar consideravelmente a velocidade de transmissão, o acesso a uma imagem continuaria a representar um tempo inaceitável já que a quantidade de informação tradicionalmente usada para a definir é muito elevada. A evolução verificada na área do processamento de imagem, nomeadamente em algoritmos de compressão, permite no entanto ultrapassar este problema e tornar mais atraente o serviço de videotex fotográfico.

Dada a grande penetração dos computadores pessoais (PCs) no mercado profissional e também, cada vez mais, em alguns sectores residenciais, optou-se por utilizar o PC como base para o terminal a desenvolver. O terminal devia ter uma arquitectura modular que facilitasse a sua evolução e a integração de novas funcionalidades.

Para a completa implementação de um serviço não basta haver equipamento e infra-estruturas de comunicação sendo necessária a definição de protocolos. Foi decidido que, para o equipamento terminal, os protocolos correspondentes aos 3 níveis inferiores de comunicação RDIS seriam entregues a uma carta externa que oferecia uma interface normalizada, sendo o nível de aplicação responsável pelo estabelecimento de uma ligação *end-to-end* utilizando os serviços fornecidos através dessa interface.

Basicamente, pode dizer-se que o terminal funciona do seguinte modo: a informação recebida da rede é identificada e entregue a uma das unidades de processamento onde a informação de controlo é extraída e os dados processados. O tratamento dos dados fotográficos e alfamo-saico é efectuado por módulos de *software* independentes e a apresentação utiliza uma placa gráfica que permite visualizar imagens de boa qualidade sem custos adicionais para o sistema. Para aplicações mais exigentes em termos de tempos de apresentação, o terminal pode utilizar uma placa aceleradora baseada num processador de sinal e que substitui o módulo de *software* responsável pela descompressão de imagem. O terminal funciona sobre MS-DOS e o controlo de todos os processos é feito por uma entidade supervisora que faz a gestão de toda a informação retornada por cada um deles.

A implementação de um sistema com estas características envolve áreas científicas muito vastas e distintas: por um lado todo o protocolo de comunicação entre a base de dados e o terminal e, por outro, o tratamento e gestão da informação. A descrição exaustiva de todos os blocos funcionais do serviço tornaria este trabalho demasiado longo. Optou-se, por isso, por fazer nesta dissertação uma descrição que permitisse obter uma visão global do sistema e uma análise mais detalhada dos pontos que foram propostos para implementação. O trabalho desenvolvido centrou-se na especificação do equipamento terminal e na implementação de uma aplicação fotográfica.

A concretização deste trabalho foi possível através da participação e colaboração em grupos de trabalho de organismos de normalização (ISO e ETSI), que permitiram que a definição e o desenvolvimento do serviço decorressem em simultâneo. Este facto possibilitou a detecção

de problemas e necessidades e a contribuição para a definição de normas com base em testes reais e efectuados em colaboração com algumas instituições de investigação europeias.

Esta dissertação está organizada em 6 capítulos.

No capítulo 2 é feito um levantamento da situação actual no campo das infra-estruturas de telecomunicações e da evolução em curso que permitirá viabilizar a oferta de novos serviços. Apresenta-se de forma sumária o serviço de videotex alfamosaico — a sua configuração e limitações — e propõe-se uma evolução para um serviço público multimédia. Esta evolução é enquadrada na necessidade das operadoras promoverem a utilização da nova rede digital com integração de serviços e também na necessidade crescente de acesso a vários tipos de informação.

No capítulo 3 faz-se a especificação do terminal de videotex alfafotográfico em termos de blocos funcionais e da comunicação entre estes. Focam-se aspectos como a importância de utilizar interfaces normalizadas e os protocolos de comunicação entre o terminal e o servidor. Apresenta-se a pilha protocolar do serviço e algumas das mensagens do nível de aplicação específicas do videotex.

O capítulo 4 faz uma apresentação das técnicas de compressão de imagem mais utilizadas e surge como introdução ao capítulo 5 e base teórica que permitiu desenvolver a aplicação fotográfica apresentada no capítulo seguinte.

No capítulo 5 descreve-se o algoritmo utilizado e a sintaxe fotográfica que possibilita uma correcta transferência e visualização de imagens entre sistemas com características diferentes. Descrevem-se algumas técnicas utilizadas na implementação do *software* de tratamento e descompressão de imagens de forma a otimizar o tempo de cálculo. Propõe-se ainda a utilização de um algoritmo de compressão que permite otimizar a base de dados e os tempos de transmissão e visualização de uma imagem.

Finalmente, no capítulo 6 é feita a síntese do trabalho realizado, indicando alguns caminhos para desenvolvimentos futuros.

Capítulo 2

Enquadramento

A evolução das redes de telecomunicações é influenciada por uma forte e complexa interacção entre dois aspectos — a inovação tecnológica e a exigência de novos serviços [LMR92] que, pelas suas características, podem ser classificados em várias classes. A divisão em serviços de distribuição e interactivos, sugerida pelo CCITT, pode ser alargada a outras categorias que permitam especificar, com mais detalhe, aspectos como a configuração da comunicação, as entidades intervenientes e o tipo de informação disponibilizado. Em [Rac91] é sugerido um conjunto de classes que incluem serviços conversacionais, de distribuição, de consulta e transferência de informação e também de vigilância. Estes novos serviços de telecomunicações — também designados por *Serviços de Valor Acrescentado* — são de importância vital em muitos sectores de actividade [Sto92]. A necessidade de troca de informação sob a forma de dados, voz, texto e imagens coloca-se não só durante todo o processo de projecto e produção numa unidade industrial mas também na fase final de divulgação de um produto ou serviço.

2.1 Infra-estruturas de Comunicações

O cenário tradicional das telecomunicações caracteriza-se por uma grande especialização das infra-estruturas de comunicação que se traduz na existência de múltiplas redes independentes concebidas e optimizadas para um dado serviço [Pry91]. A rede telefónica comutada e a rede de dados de comutação de pacotes (X.25) são exemplos perfeitos para ilustrar esta situação: por um lado o serviço clássico de transferência bidireccional de voz e por outro a transferência de dados. Em ambientes informáticos privados surgem ainda as redes locais das quais se destacam a Ethernet e Token Ring.

As grandes diferenças existentes, no que diz respeito a qualidade, velocidade e técnicas de transmissão, impedem, normalmente, a utilização de uma rede para um fim para o qual não foi

concebida — a transmissão de voz sobre a rede pública de comutação de pacotes seria bastante problemática devido à existência de atrasos variáveis que conduziriam a um serviço telefónico de qualidade inaceitável. Apenas num número limitado de casos é possível a utilização de uma rede para um fim diferente daquele para o qual foi projectada — a transferência de dados sobre a rede telefónica é possível desde que existam disponíveis modems de ambos os lados da ligação.

Como se pode facilmente concluir, esta situação apresenta um número elevado de desvantagens entre as quais se podem destacar as seguintes:

- Dependência do serviço — cada rede é apenas capaz de suportar um serviço específico para o qual foi preparada;
- Inflexibilidade — a especialização de uma rede levanta graves dificuldades à sua adaptação a novos serviços que possam surgir como consequência dos avanços tecnológicos verificados tanto a nível da electrónica como de algoritmos de codificação de áudio, vídeo e voz;
- Ineficiência — a existência de recursos disponíveis numa rede não significa a sua utilização por um outro serviço que deles necessite.

Todos estes problemas e exigências conduziram ao aparecimento do conceito de *Integração de Serviços* e à identificação de uma solução no campo das telecomunicações — a Rede Digital com Integração de Serviços (RDIS)[Tan89],[Sta94]. Esta, ao suportar uma gama alargada de serviços (voz, dados, imagem) num único meio físico, permite uma redução dos custos de operação e manutenção e um aumento de flexibilidade. Do ponto de vista do utilizador comum, esta integração traduz-se num aumento do número de serviços disponíveis, maiores velocidades de transmissão e melhor qualidade de serviço.

Ao disponibilizar, num acesso básico, dois canais de 64Kbps e um canal de 16Kbps (2B+D), a RDIS constitui um suporte muito conveniente para serviços como o videotelefone, a televigilância e o videotex. No entanto, este aspecto por si só não garante a aceitação desta nova rede de telecomunicações por parte do público em geral. Os factores cruciais, que determinarão o sucesso ou insucesso da RDIS, serão a existência de equipamento terminal adequado, a oferta de novos serviços e aplicações e uma política de preços atraente.

A introdução de serviços multimédia que permitissem a um utilizador aceder a várias formas de informação através de um único suporte físico e utilizando um terminal multifuncional foi considerada de importância vital por parte das operadoras portuguesas de telecomunicações. A inexistência no mercado nacional e internacional de uma aplicação que satisfizesse estes requisitos levou as operadoras a apoiar e incentivar o desenvolvimento de um serviço deste tipo.

2.2 Serviço Público Multimédia

A necessidade de aceder a informação de vários tipos levou ao aparecimento de várias soluções que pretendiam colmatar uma falha até agora existente. No entanto os produtos multimédia e hipermédia actualmente disponíveis apresentam ainda alguns problemas já que se baseiam normalmente em acessos locais e em soluções proprietárias. Este facto inviabiliza aplicações em campos que requeiram a disponibilização de informação a um número elevado de consumidores [VLO⁺93, LOS⁺93].

A dificuldade em obter compatibilidade e integração de produtos provenientes de diversos fabricantes conduziu a um cenário inovador nas telecomunicações na Europa — a convergência em direcção a soluções comuns através do desenvolvimento de normas que permitam ao cliente obter uma solução economicamente interessante. Um exemplo perfeito desta nova linha de orientação e harmonização de serviços de telecomunicações e multimédia é o videotex.

O videotex é um serviço público interactivo através do qual o utilizador, possuidor de um terminal específico, pode aceder a uma base de dados remota na qual se encontra armazenada informação de diversos tipos (texto, gráficos, imagens)[Alb85]. Este serviço está já a ser explorado comercialmente há alguns anos utilizando para tal o acesso telefónico. A ligação ao servidor de videotex é obtida utilizando a rede telefónica até a um nó denominado *Ponto de Acesso de Videotex* (PAV), a partir do qual a transmissão é suportada pela rede X.25 (figura 2.1).

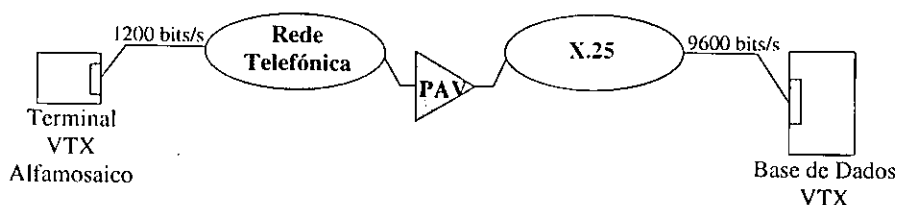


Figura 2.1: Configuração actual do serviço de videotex

As limitações de velocidade impostas pela configuração actual conduzem a tempos de transmissão relativamente elevados e, conseqüentemente, a uma necessidade de reduzir a quantidade de informação que deve circular na rede. Por este facto, o serviço de videotex a funcionar em Portugal desde 1989 está limitado a um único modo designado alfamosaico. As imagens são construídas através de um conjunto pré-definido de caracteres (mosaicos) que conduzem a uma figura com pouca definição (figura 2.2). Imagens gráficas podem ser obtidas através da utilização de tabelas especiais com caracteres definidos pelo servidor para uma aplicação

específica e enviadas ao terminal DRCS (*Dynamically Redefinable Character Set*).

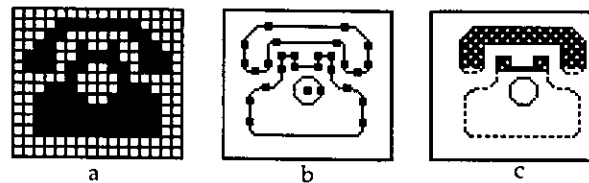


Figura 2.2: Sintaxes de videotex a)mosaico b)geométrico c)fotográfico

Embora deficiente em termos de apresentação da informação disponível, este serviço conseguiu implantar-se de forma significativa em muitos países. No entanto, devido a desenvolvimentos paralelos, não foi possível obter uma solução uniforme e única a nível mundial, surgindo várias versões incompatíveis entre si [MH89]:

- sintaxe I: norma japonesa também designada por *CAPTAIN* — *Character and Pattern Telephone Access Information Network system*
- sintaxe II: norma europeia desenvolvida pela CEPT — *Conférence Européenne des Postes et Télécommunications*
- sintaxe III: norma utilizada nos Estados Unidos da América e conhecida pela designação de *NAPLPS* — *North American Presentation Level Protocol Syntax*.

A nível europeu a norma CEPT apresenta ainda três variantes (CEPT 1, 2 e 3) que tornam difícil a comunicação e partilha de informação. Para evitar a repetição deste problema, a evolução do videotex em direcção a um ambiente multimédia tem sido acompanhada por um trabalho de normalização por parte do ETSI (*European Telecommunications Standards Institute*). As novas sintaxes definem um formato baseado em primitivas geométricas, que permitem a interligação de pontos utilizando rectas ou curvas, e um outro que permite obter imagens de qualidade fotográfica.

2.3 O Novo Serviço de Videotex

A introdução de serviços de telecomunicações, com características inovadoras e que permitam satisfazer as exigências dos utilizadores, torna-se viável com a implantação da RDIS.

Aplicações como a transferência de dados sobre uma rede mais fiável e rápida, a videoconferência, a televigilância e o trabalho cooperativo são exemplos de novas aplicações que se tornaram possíveis. A consulta a bases de dados remotas contendo informação fotográfica sobre a forma de videotex surgiu como uma evolução natural de um serviço já existente mas não completamente satisfatório em muitos aspectos.

O desenvolvimento de um serviço de videotex alfafotográfico (alfamosaico + fotográfico) foi apoiado pela Telecom Portugal de forma a permitir a partilha de informação multimédia e o acesso a servidores internacionais, mas mantendo compatibilidade com o anterior serviço. A utilização de terminais específicos foi considerada uma limitação ao crescimento e divulgação do serviço e dada a grande penetração de computadores pessoais IBM ou compatíveis (PC — Personal Computer) optou-se pela sua utilização como base para o terminal a desenvolver. A figura 2.3 ilustra a configuração do futuro serviço de videotex.

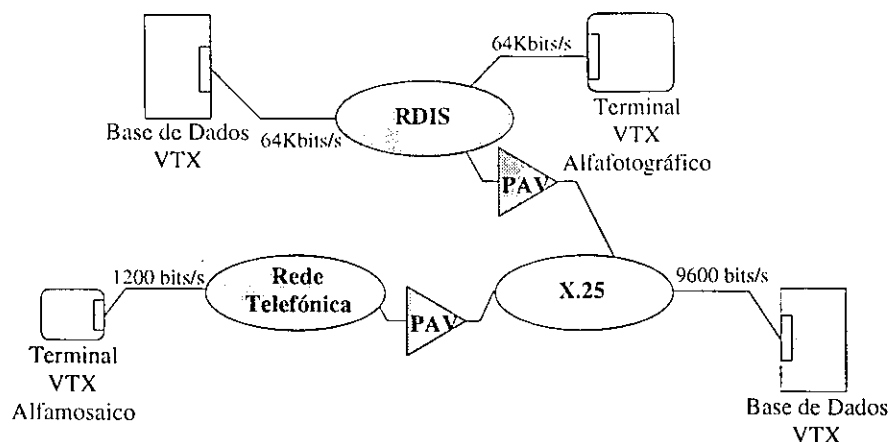


Figura 2.3: Configuração do futuro serviço de videotex

Capítulo 3

Especificação do Equipamento Terminal

Na arquitectura global do futuro serviço de videotex (figura 2.3), o equipamento terminal funciona como interface entre o sistema e o utilizador sendo por isso de importância vital para o bom funcionamento, aceitação e evolução do serviço. O terminal fornece a plataforma de acesso a um serviço multimédia interactivo e deve, por isso, ser de fácil utilização e oferecer todas as funcionalidades necessárias.

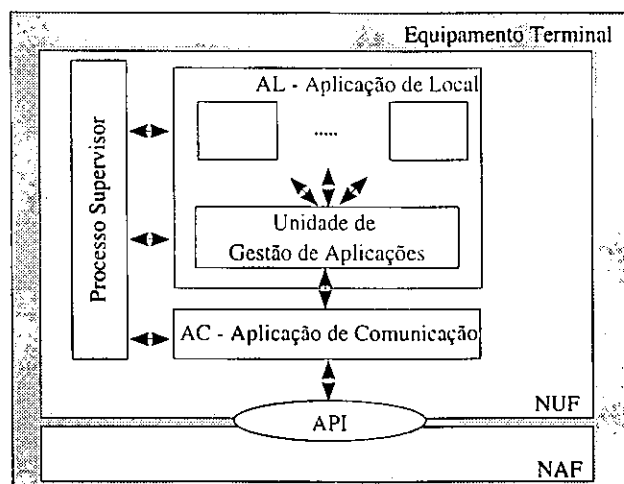


Figura 3.1: Modelo do Equipamento Terminal de Videotex

Devido à sua complexidade, o desenvolvimento de um equipamento terminal deste tipo deve ser baseado num modelo que permita identificar mais facilmente as necessidades e estruturar

a solução de uma forma clara. A figura 3.1 representa o modelo proposto para a especificação do terminal de videotex alfafotográfico, salientando os diferentes blocos funcionais.

3.1 API — *Application Program Interface*

As inovações introduzidas pela RDIS no campo das telecomunicações conduziram a um crescente interesse na criação de novos serviços e na ligação de equipamentos terminais não dedicados à rede pública com integração de serviços. Do ponto de vista das comunicações, um grande esforço foi feito pelos organismos de normalização de forma a obter uma uniformização do sistema de sinalização no canal D e de comunicação *end-to-end* no canal B, tendo por objectivo permitir um elevado grau de compatibilidade entre as redes de países distintos. No entanto, a especificação de protocolos deixou em aberto questões como o acesso a serviços fornecidos por camadas inferiores: um dos problemas com que actualmente se depara quem pretenda oferecer novos equipamentos ou serviços, é a existência de adaptadores de rede, para o desenvolvimento de aplicações de comunicação, com interfaces de *software* diferentes (figura 3.2). Como resultado, a construção de *software* para novas aplicações apresenta, à partida, um conjunto de limitações:

- é dependente do *software* e *hardware* fornecidos por um determinado fabricante;
- é dependente do sistema operativo e da arquitectura do computador.

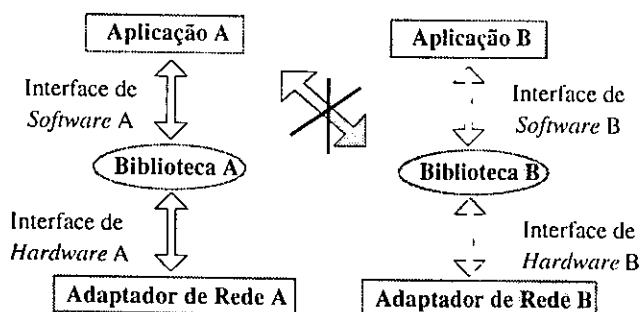


Figura 3.2: Problema actual de incompatibilidade entre aplicações e interfaces RDIS

Estes factores podem desencorajar a entrada de novos fornecedores de serviços no campo da RDIS e, conseqüentemente, limitar o crescimento do número de potenciais utilizadores. As aplicações devem ser portáteis para as várias cartas RDIS que implementam as 3 camadas OSI (*Open Systems Interconnection*) mais baixas [WR91]. Estas cartas estão integradas num

módulo de *software* que implementa as camadas altas e a aplicação e que exige, por isso, uma interface normalizada para aceder às primitivas de serviço RDIS para os canais B e D.

O termo API está associado a um conjunto de interfaces de *software* que têm em comum o facto de permitirem o acesso a serviços fornecidos por um sistema operativo, facilitando a utilização e controlo dos recursos disponíveis — disco, memória e outros periféricos. As interfaces fornecidas por sistemas operativos como o MS-DOS e o UNIX são exemplos de APIs que atingiram já um elevado grau de normalização.

A utilização de APIs ao nível das comunicações, e segundo a filosofia OSI, levou à criação de uma designação equivalente — PCI (*Programmable Communication Interface*) — por vezes usada para referir a interface entre os dois módulos de comunicação da figura 3.1:

- NAF (*Network Access Facility*) — disponibiliza todo um conjunto de funções necessárias para acesso à rede e consiste basicamente em vários módulos de *hardware* e *software*;
- NUF (*Network User Facility*) — utiliza as funções fornecidas pelo NAF para implementar as aplicações no equipamento terminal. Dentro deste módulo é ainda possível distinguir um bloco relacionado com aspectos de comunicação — Aplicação de Comunicação (AC) — e outro que lida apenas com os aspectos locais ao terminal — Aplicação Local (AL).

Em termos de modelo OSI, o NAF implementa os níveis 1 a 3 e comunica, através de uma interface normalizada (API), com os níveis superiores (AC). Este bloco, responsável pelos protocolos de comunicação dos níveis 4 a 7, comunica por sua vez com a aplicação local (AL) por intermédio de uma outra interface.

A definição e utilização de uma interface API para RDIS é portanto de importância vital. O módulo de acesso à rede deve fornecer uma interface normalizada que facilite a comunicação do nível 3 OSI com os níveis superiores.

3.1.1 Esquema de Comunicação com a API

A comunicação entre a aplicação e a API [API93] faz-se através de uma fila de mensagens por cada aplicação registada (figura 3.3). As filas de espera, do tipo FIFO (*First-In-First-Out*), formam a ligação entre a aplicação e a API permitindo o diálogo com os controladores RDIS.

O controlo das filas de espera é possível através de uma das seguintes operações:

- **API_register**

é utilizada pela aplicação para pedir à API um número único que a identifique (APPL_ID). Esta operação cria a fila de espera necessária para a comunicação entre a placa e a aplicação;

- **API_put_message**

permite a transferência de mensagens da aplicação para a API, colocando-as na fila de espera;

- **API_get_message**

utilizando esta primitiva, as aplicações retiram da fila de espera que lhes está associada as mensagens enviadas pela API;

- **API_release**

primitiva que permite libertar todos os recursos alocados a uma determinada aplicação. O APPL_ID deixa de lhe estar atribuído e o espaço de memória associado à sua fila de espera é também libertado.

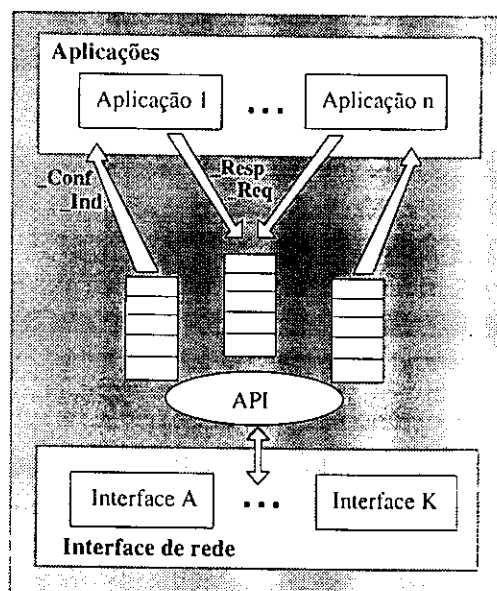


Figura 3.3: Estrutura de filas de espera entre a API e a Aplicação

As aplicações e a API colocam nas filas de espera mensagens numeradas sequencialmente entre 0x0000 e 0x7FFF para mensagens provenientes da aplicação e entre 0x8000 e 0xFFFF no caso de mensagens vindas da rede. Estas mensagens podem ser de 4 tipos — `_Request` (`_Req`), `_Response` (`_Resp`), `_Confirm` (`_Conf`) e `_Indication` (`_Ind`) — sendo as duas primeiras

enviadas no sentido aplicação → API e as restantes no sentido API → aplicação. Quando a aplicação envia um *_Req*, a API deve responder com uma mensagem do tipo *_Conf* com o mesmo número. Sempre que a API envia uma *_Ind*, a aplicação responde com uma *_Resp* com o mesmo número que a *_Ind*.

3.2 Protocolos de Comunicação

A pilha de protocolos de comunicação do serviço de videotex caracteriza-se pela não existência dos níveis 4 a 6. A implementação dos níveis OSI 1 a 3 é efectuada recorrendo a uma carta de rede que oferece uma API normalizada [API93], sendo o módulo AC responsável pela implementação do nível 7 [ETSB]. Funções como o estabelecimento e libertação da chamada, o reset de ligação ou a identificação de recepção de dados, são realizadas através de um mapeamento dos serviços em sequências de mensagens da interface API.

3.2.1 NAF — *Network Access Facility*

O serviço de videotex baseia-se em ligações do tipo comutação de circuitos [Sta94], sendo os níveis inferiores os representados na pilha protocolar da figura 3.4 [ETSA]. Através do canal D são veiculadas mensagens responsáveis pelos protocolos de sinalização RDIS, que implementam as recomendações Q.931 [Q931], para o nível 3, e Q.921 (LAPD — *Link Access Protocol, D channel*) [Q921] para o nível 2. A informação dos canais B e D é multiplexada na camada física de acordo com a recomendação I.430 [I430] que define a interface básica. A comunicação sobre o canal B é efectuada utilizando, no nível de ligação lógica, o protocolo LAPB (*Link Access Protocol, Balanced*) e no nível rede o protocolo X.25.

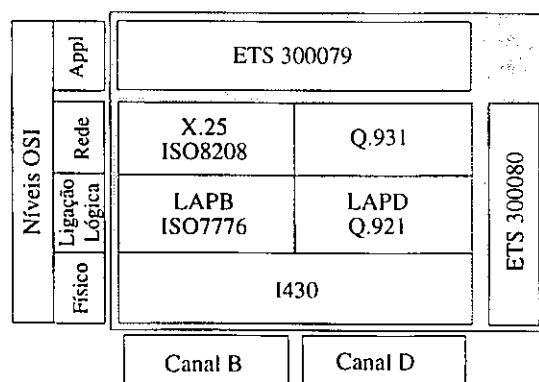


Figura 3.4: Pilha protocolar do serviço de videotex

A carta de rede utilizada oferece um acesso básico S_0 e comunica com a API através de uma memória de duplo porto mapeada na memória do PC. Esta memória está dividida em dois bancos utilizados, respectivamente, no sentido PC $\rightarrow$ placa e placa $\rightarrow$ PC, ou seja, a API escreve no primeiro e lê do segundo e vice-versa no que diz respeito à placa. O processo de sinalização entre a API e a placa é efectuado por meio de interrupções que são activadas pela escrita em zonas específicas da memória, interrupções essas desactivadas quando a posição de memória responsável pela interrupção é lida. O valor lido identifica o motivo de interrupção — chegada de uma mensagem da placa ou simplesmente confirmação de recepção.

As mensagens provenientes da aplicação são primeiro depositadas na fila de espera da API referida em 3.1.1 e só depois colocadas nesta memória partilhada. Após confirmação de recepção de uma ou mais mensagens pela placa, o espaço por elas ocupado no banco de memória é libertado podendo ser preenchido por outras que estejam na fila. Este processo gera uma interrupção à placa que permite sinalizar a chegada de mais dados. No sentido placa $\rightarrow$ aplicação, cada mensagem lida da memória será colocada na fila de espera referente à aplicação em causa. Casos de *overflow* poderão acontecer, sendo a mensagem rejeitada e a aplicação avisada deste facto. A primitiva `API_get_message` permite à aplicação obter um apontador para a primeira mensagem armazenada na sua fila de espera.

3.2.2 Aplicação de Comunicação

Os vários serviços de videotex são mapeados em sequências de mensagens da API que, por sua vez, podem ser mapeadas em mensagens de sinalização canal D, pacotes de rede X.25 ou em troca local de informação entre a aplicação e o controlador. A figura 3.5 ilustra os procedimentos de sinalização entre um terminal e um servidor de videotex através da utilização de alguns dos serviços de aplicação:

- **Sbv_Establish**

este serviço gera um conjunto de mensagens de sinalização que permitem o estabelecimento de um canal de comunicação de acordo com a informação trocada localmente com o controlador para seleccionar os protocolos de nível inferior a utilizar (`Select_prc1`);

- **Sbv_Release**

libertação de uma chamada. Esta mensagem pode ser iniciada por ambos os lados e, ao contrário do anterior, é não confirmada;

- **Sbv_Vtx_Data**

transferência de dados codificados de acordos com as normas apropriadas [ETSc],[ETSD],[ETSe], [ETSI],[ETSG],[ETSh]. Este serviço, não confirmado, pode também ser iniciado por ambos os lados;

- **Sbv_Reset**

colocação do terminal no estado básico que corresponde ao estado inicial após o estabelecimento, com sucesso, de uma ligação. Nesta situação, os parâmetros X.3 (referidos à frente, neste capítulo) são colocados no valor por defeito e a lista de teclas de função é anulada.

Além das mensagens básicas que permitem iniciar, manter e terminar uma ligação, o bom funcionamento do serviço, após o estabelecimento do canal B, exige a troca de informação entre o terminal e servidor sob a forma de *comandos telemáticos*. Estes são codificados numa sequência completa de pacotes de dados X.25 com o bit Q (*Qualified Data*) colocado a 1 [X25]. Sbv_Dfk e Sbv_Tfi são exemplos deste tipo de serviços:

- **Sbv_Dfk**

o serviço de videotex utiliza sequências de caracteres associados a determinadas acções repetidas várias vezes durante uma sessão. A tabela 3.1 refere alguns dos comandos mais frequentes. O código associado a uma acção poderá depender do serviço em causa sendo diferente para os serviços de videotex X e Y. A utilização do comando telemático Sbv_Dfk permite, ao servidor, definir ou anular uma lista de teclas especiais (teclas de função) que substituem toda a sequência de caracteres correspondente a uma função, tornando assim esta acção transparente ao utilizador. A tabela 3.2 exemplifica o mapeamento da sequência *# na tecla de função 1 com a definição do nome que será visível ao utilizador;

- **Sbv_Tfi**

pedido de informação do servidor sobre as facilidades concedidas pelo terminal (secção 3.3.1).

A sequência de mensagens entre a aplicação e a API resultantes destes serviços pode ser representado através do fluxograma da figura 3.6.

Tabela 3.1: Comandos de Videotex

Comando	Ação
#	Página seguinte
*#	Repetir última página
*90#	Abandonar o serviço
*0#	Menu inicial

Tabela 3.2: Exemplo de codificação de um comando telemático

Código	Significado
0x40	Identificação de comando telemático
0x9E	Videotex sobre RDIS
0x0A	Código do comando Sbv_Dfk
0x11	Comprimento da mensagem: 17 bytes
0xFF	Estrutura de codificação estendida
0xE1	Indicador de lista de teclas de função
0x0E	Comprimento: 14 bytes
0x60	Indicador de início de tecla de função
0x0C	Comprimento: 12 bytes
0x61	Indicador de identificação
0x01	Comprimento: 1 byte
0x01	Número da tecla
0x62	Indicador de nome
0x03	Comprimento: 3 bytes
R	
E	
P	
0x63	Indicador de sequência de caracteres
0x02	Comprimento: 2 bytes
*	
#	

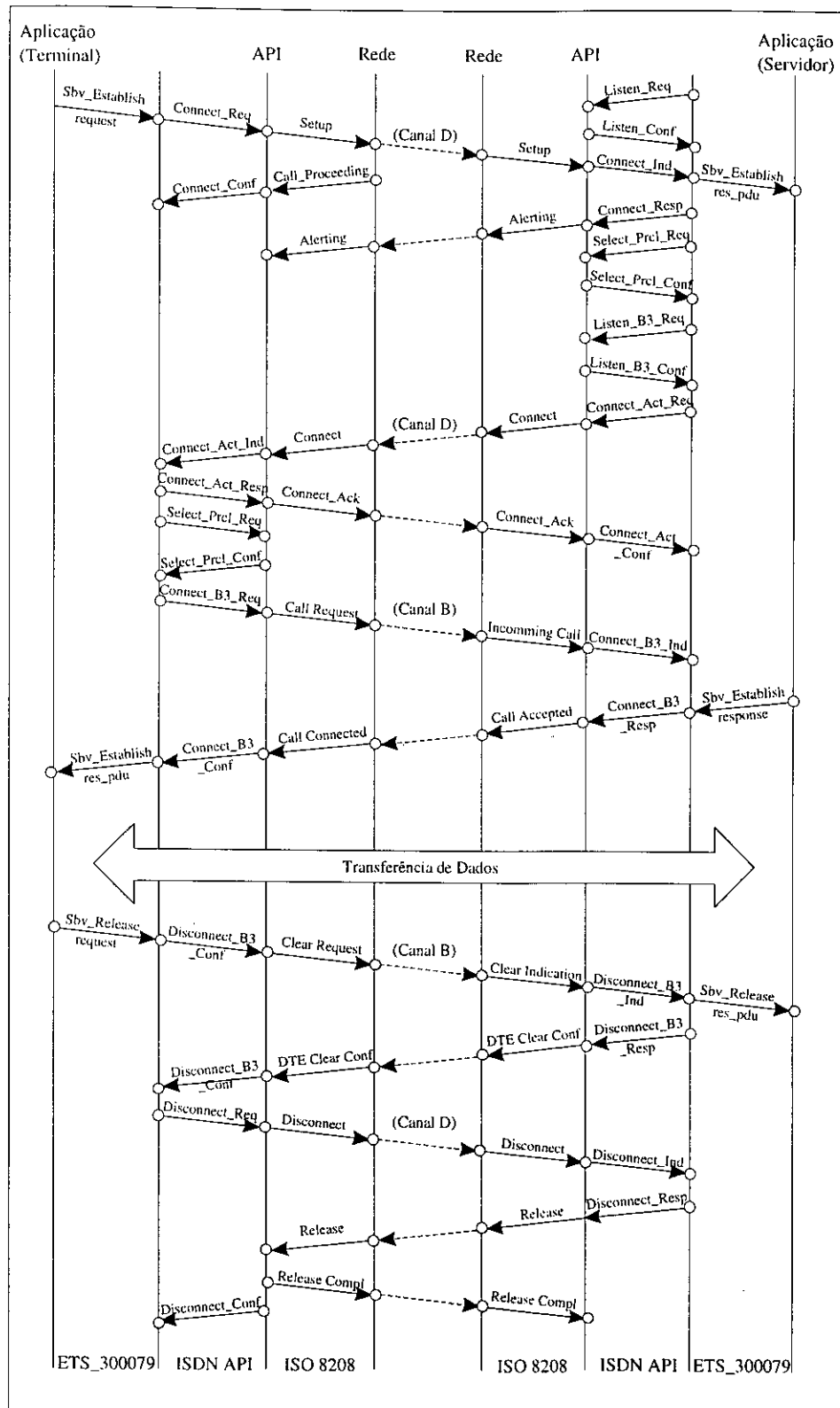


Figura 3.5: Procedimentos de sinalização entre um terminal e um servidor de videotex RDIS

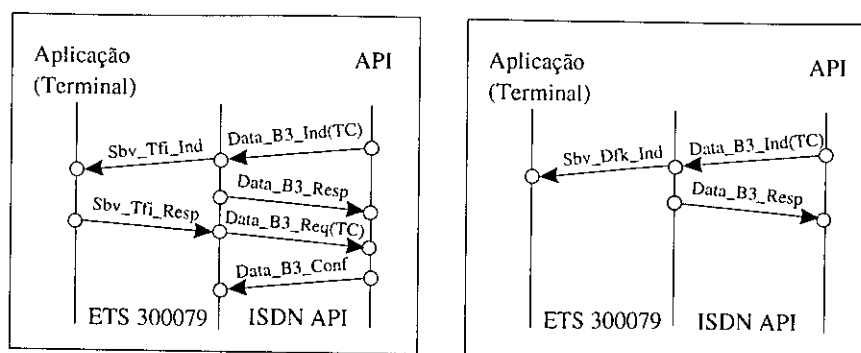


Figura 3.6: Fluxograma dos serviços de nível aplicação Sbv_Tfi e Sbv_Dfk

Tendo o servidor de videotex um modo de funcionamento orientado ao pacote (*packet-mode*) e o terminal ao carácter (*character-mode*), é necessária a implementação de um mecanismo de PAD (*Packet Assembly-Disassembly*) como definido nas normas X.3 [X3] e X.29 [X29]. Na manipulação dos parâmetros X.3 (definição de caracteres de envio, tamanho dos campos de entrada de dados, eco, etc.) são utilizadas as seguintes mensagens:

- **Sbv_Set_Param**

este serviço é utilizado para alterar um ou mais parâmetros X.3 do terminal. É um serviço não confirmado e que apenas pode ser iniciado pelo servidor;

- **Sbv_Read_Param**

serviço iniciado pelo servidor para interrogar o terminal quanto ao estado de de um ou mais parâmetros X.3. É também não confirmado;

- **Sbv_Set_Read_Param**

utilizando este serviço não confirmado, o servidor pode simultaneamente determinar o estado de uma lista de parâmetros e alterar outros;

- **Sbv_Param_Ind**

este serviço permite ao terminal indicar o estado dos seus parâmetros X.3 como resposta a um `_Read` ou `_Set_Read`.

Os parâmetros X.3 suportados no serviço de videotex são enviados por meio de mensagens X.29 PAD codificadas em pacotes de dados X.25 com $Q = 1$. Estas mensagens consistem num código (que indica uma das acções acima mencionadas) e num campo de parâmetros com uma referência e um valor.

3.3 Aplicação Local

Uma vez estabelecida a ligação entre um terminal e um servidor, a gestão e tratamento dos pacotes de dados recebidos é da responsabilidade do módulo de aplicação local.

A unidade responsável pela gestão das aplicações implementa a máquina de estados da figura 3.7 transferindo os dados codificados segundo um dos formatos referidos em 3.2.2 para a aplicação apropriada. Um terminal que implemente apenas a sintaxe II permanece no estado 2 até detectar uma das entradas permitidas. Nessa altura os dados são enviados à aplicação que, após executar as acções exigidas, repõe o sistema no estado 2. No caso de suporte de várias sintaxes, o terminal deve ser colocado no estado 0.

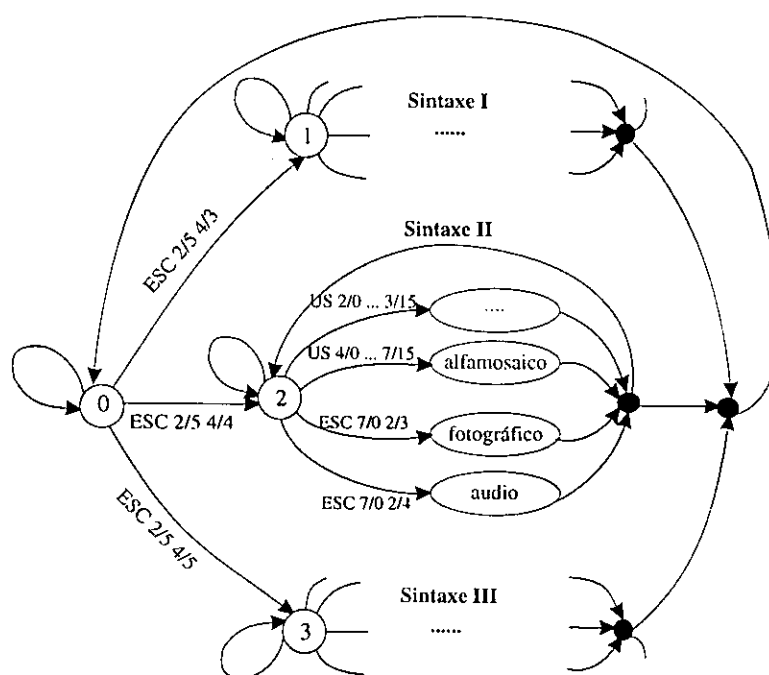


Figura 3.7: Máquina de estados de um terminal

Uma vez que os terminais de videotex não foram concebidos para suportar todas as técnicas possíveis de codificação (texto, gráficos, imagens) e todas as facilidades disponíveis em cada uma, tornou-se necessário criar um mecanismo que permitisse estabelecer um diálogo entre o terminal e o servidor com o fim de identificar as capacidades do terminal — TFI (*Terminal Facility Identifier*) [ETSg].

3.3.1 TFI — *Terminal Facility Identifier*

Para garantir o sucesso da comunicação, o serviço de videotex deverá, no início da ligação, inquirir o terminal acerca do suporte dos serviços que especificamente vai utilizar.

O TFI pode ser utilizados para obter as seguintes informações:

- sintaxes de videotex implementadas no terminal (secção 2.2);
- tipos de dados suportados ([ETSc] – [ETSf]) e facilidades adicionais (definição de DRCS, tabelas de cor, etc);
- conjunto de funcionalidades implementadas para cada um dos tipos de dados suportados (perfis fotográficos referidos em 5.2, tabelas de caracteres específicos de uma língua, etc).

A obtenção destas informações é feita através do comando telemático *Sbv_Tfi*, indicando um pedido de informação global ou de existência de uma facilidade em particular. Depois de devidamente informado, o servidor de videotex poderá comutar o terminal para um dos serviços suportados através de uma sequência de caracteres adequada (figura 3.7).

3.3.2 Interface com o Utilizador

Um das deficiências do serviço de videotex alfamosaico é a interface pouco amigável com o utilizador — as ordens de execução de acções comuns, como estabelecer ou terminar uma ligação, repetir ou pedir uma nova página, são dadas por comandos de videotex referidos na tabela 3.1. A introdução do videotex fotográfico exige uma redefinição da interface homem-máquina de modo a permitir uma utilização intuitiva do serviço. No esquema da figura 3.8 é possível distinguir 5 zonas diferentes:

- Menus — reúnem o conjunto de acções que permitem estabelecer ou terminar uma chamada, armazenar localmente uma página recebida durante uma sessão, visualizar uma imagem gravada numa sessão anterior, consultar um conjunto de informações sobre o serviço e abandonar a sessão;
- Teclas de Funções Especiais — após um comando telemático *Sbv_Dfk*, a cada uma das teclas definidas vai ser associada uma acção específica;
- Mensagens — zona utilizada pela unidade supervisora para dar ao utilizador indicações de erro ou processos a decorrer;
- Comandos de VTX — a definição de teclas de funções especiais não elimina a hipótese de utilizar caracteres específicos de comandos de videotex: os caracteres * e # da

norma portuguesa correspondem na verdade aos caracteres especiais INI e TER. Ao iniciar uma sessão, o terminal define as duas teclas especificadas de forma que para o utilizador sejam semelhantes a teclas de função;

- Área útil de Dados — zona da interface onde são colocadas as páginas alfafotográficas.

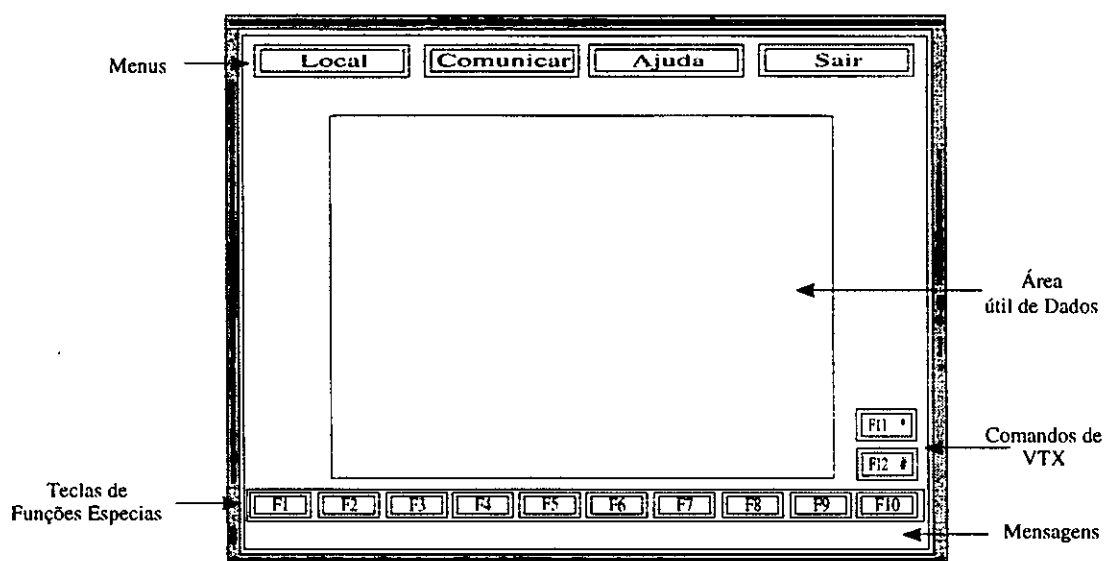


Figura 3.8: Esquema da interface de utilizador do terminal de videotex alfafotográfico



Figura 3.9: Interface de utilizador do terminal de videotex alfafotográfico

Capítulo 4

Fundamentos de Compressão de Imagem

A introdução de informação fotográfica no serviço de videotex exige o recurso a técnicas de compressão que permitam reduzir, para quantidades aceitáveis, o fluxo de dados. As bases teóricas de compressão de sinal datam já de há alguns anos e surgem como resultado do trabalho apresentado por Shannon em 1948 [Sha48]. Shannon demonstrou que informação digital podia ser comprimida de forma a permitir a sua reconstrução posterior sem erros desde que um número suficiente de bits fosse utilizado (secção 4.4).

No entanto, a teoria clássica de compressão apresenta algumas falhas já que não analisa aspectos actualmente considerados fundamentais. No caso específico de compressão de imagem, além da redundância de informação resultante de zonas de cor constante e textura regular, há ainda que entrar também em consideração com o facto de o olho humano ser pouco sensível a determinadas características. Isto vai permitir obter novas representações que requerem uma menor quantidade de informação, mantendo no entanto uma qualidade aceitável. Pode pois afirmar-se que existem duas operações intrínsecas ao processo de compressão de imagem: eliminação da redundância e redução de aspectos irrelevantes [JJS93], [Jay92] (figura 4.1).

A eliminação da redundância, que resulta de um processo de amostragem quando da digitalização de uma imagem, pode ser feita no domínio dos tempos (compressão por predição) ou das frequências (compressão por transformadas). As técnicas de compressão com perdas, geralmente associadas a um processo de quantificação, foram rapidamente combinadas com as técnicas sem perdas introduzidas por Shannon. A colocação em cascata de vários blocos permite obter um sistema de compressão com taxas mais elevadas, sendo as entradas do quantificador, geralmente, sequências de erros de predição ou coeficientes transformados. A ideia básica é quantificar as amostras ou coeficientes de modo a obter uma distorção imperceptível

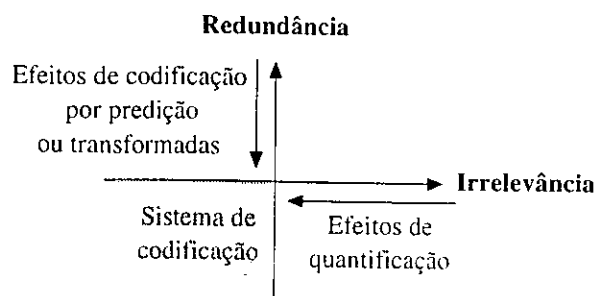


Figura 4.1: Informação redundante e irrelevante

apesar de, matematicamente, diferente de zero. Se as taxas de compressão a obter são demasiado elevadas para permitir este tipo de transparência, então tenta-se minimizar a distorção visível fazendo com que esta seja mascarada pelas características próprias da imagem. O termo *Compressão Perceptual* é utilizado para designar esta tentativa de otimizar o sistema de quantificação, tendo em consideração que o último elemento da cadeia de compressão é um auditório humano, de forma a minimizar a distorção visível ou mesmo, quando possível, levá-la a zero. Estes objectivos não têm uma correspondência directa com a maximização da relação sinal-ruído (minimização do erro quadrático médio).

Os blocos do codificador responsáveis pelo tratamento das partes redundantes e irrelevantes da informação são, em muitos casos, distintos. No entanto existem algoritmos em que estas duas funções não podem ser facilmente separadas — é o caso da quantificação vectorial apresentada em 4.5.

A maior parte dos sistemas de compressão depende de uma correcta interligação entre as duas operações descritas. A codificação PCM (*Pulse Code Modulation*), sendo baseada num sistema sem memória, é uma rara excepção pois não há qualquer tentativa para utilizar a redundância do sinal. No entanto, a maior parte das aplicações exige uma sofisticada conjugação de um processo de eliminação de redundância seguido por uma quantificação ajustada à percepção humana.

Nos últimos anos verificou-se um progresso significativo nas técnicas de compressão de áudio, imagem e vídeo, como resultado de um trabalho conjunto nas áreas da teoria da informação, do processamento de sinal e da codificação perceptual.

Os princípios, conceptualmente ortogonais, de eliminação de redundância e irrelevância são desde há muito tempo conhecidos no campo do processamento de sinal. Neste sentido, o conceito de *Codificação Perceptual* não é novo. O que é significativo é que ambas as direcções da figura 4.1 têm avançado consideravelmente nos últimos anos e a interacção entre elas é cada vez melhor compreendida.

Existem pois basicamente dois modos de abordar o problema da compressão:

- a informação comprimida deve ser exactamente igual à original sendo apenas representada de forma mais eficiente. Este método de compressão sem perdas é de importância vital em sectores de actividade como a medicina pois um determinado diagnóstico ou conclusão poderão ser afectados por alterações da imagem resultantes da compressão;
- o objectivo é obter elevados graus de compressão mesmo que se proceda a alterações irreversíveis nos dados — compressão com perdas — as quais podem introduzir artefactos visíveis na imagem.

O problema genérico em compressão de imagem é maximizar a taxa de compressão mantendo, no entanto, níveis de qualidade, complexidade de implementação e atrasos de comunicação aceitáveis. A complexidade de um algoritmo representa o esforço computacional necessário para o implementar e é geralmente medida em termos de memória necessária e capacidade de processamento (MIPS - *Millions of Instructions per Second*). A qualidade pode ser medida em termos de um valor objectivo (erro quadrático médio, por exemplo) ou recorrendo a uma medida subjectiva (MOS — *Mean Opinion Score*). Qualquer uma destas medidas de desempenho representa um factor que terá que ser ponderado na escolha de um algoritmo para uma aplicação específica. Os eixos da figura 4.2 definem um espaço de 4 dimensões no qual algumas regiões são teoricamente possíveis e outras são desejáveis para algumas aplicações.

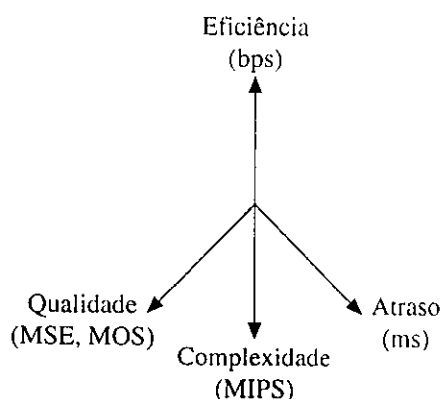


Figura 4.2: Desempenho de um algoritmo

Neste capítulo tenta-se dar uma visão geral de várias técnicas de compressão sem haver, no entanto, a preocupação de se ser exaustivo. Uma análise mais detalhada do problema pode ser obtida em [GW87],[Pra78],[Cla85],[NH89],[Jai89], [DP91] e [Jai81].

4.1 Codificação por Predição

Uma das técnicas básicas para representar e transmitir um sinal é o sistema de codificação PCM em que o sinal é amostrado e quantificado utilizando um número fixo de bits. No caso particular de um sinal de vídeo, é comum utilizar 8 bits por amostra (ou por componente no caso de imagens policromáticas) sendo esse valor superior quando a qualidade é uma exigência importante para uma boa utilização do sistema — no caso de imagens médicas poderão ser necessários 10 a 12 bits por pixel (*picture element*).

Neste método, cada amostra é pois processada independentemente, ignorando a interdependência entre amostras adjacentes. A filosofia básica por trás das técnicas preditivas é a eliminação de redundância entre pontos sucessivos e a codificação apenas da nova informação: uma amostra x_n é codificada com base numa quantidade $\hat{x}_n$ calculada com base nas amostras anteriormente reproduzidas no decodificador ($\hat{x}_{n-1}, \hat{x}_{n-2}, \dots$). A quantidade $\hat{x}_n$ — uma estimativa de x_n — é obtida por

$$\hat{x}_n = \psi(\hat{x}_{n-1}, \hat{x}_{n-2}, \dots). \quad (4.1)$$

em que $\psi(\cdot, \cdot, \dots)$ representa a regra de predição.

O erro de predição, definido como

$$e_n = x_n - \hat{x}_n, \quad (4.2)$$

é quantificado num número de níveis pré-definido originando a quantidade $\hat{e}_n$. Assumindo que a estimativa de x_n é razoavelmente precisa, a diferença $x_n - \hat{x}_n$ será, em média, significativamente inferior à amplitude da amostra x_n . Isto permite a utilização de um menor número de níveis de quantificação e daí decorre a necessidade de menos bits para codificar a sequência das diferenças comparativamente com o que seria necessário para codificar a sequência de amostras.

Se $\hat{e}_n$ representar o valor quantificado de e_n , então o valor de x_n reproduzido é

$$\hat{x}_n = \hat{x}_n + \hat{e}_n. \quad (4.3)$$

Este processo de codificação repetido recursivamente é designado de DPCM (*Differential Pulse Code Modulation*) e está representado na figura 4.3. Uma vez que o codificador utiliza os valores das amostras anteriores afectadas do erro de quantificação ($\hat{x}_{n-1}, \hat{x}_{n-2}, \dots$) para calcular a estimativa de x_n , a malha de realimentação que constitui o decodificador é também

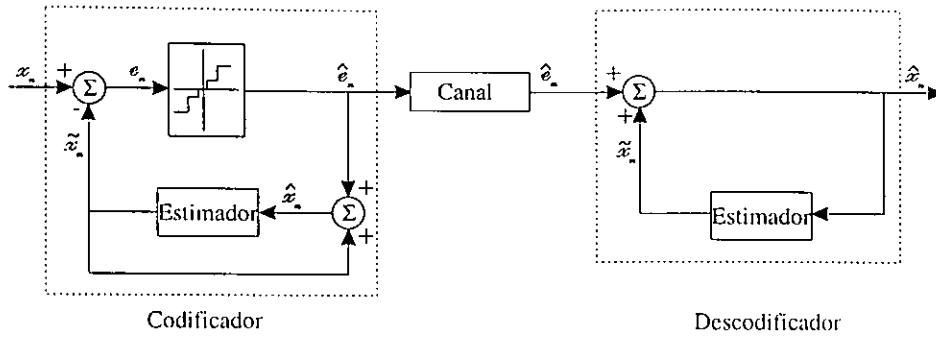


Figura 4.3: Sistema DPCM

implementada no codificador. Isto garante que as estimativas calculadas no codificador e decodificador são iguais.

Escrevendo (4.2) na forma

$$x_n = \tilde{x}_n + e_n, \quad (4.4)$$

e subtraindo (4.3), obtém-se

$$\delta x_n = x_n - \hat{x}_n = e_n - \hat{e}_n = q_n. \quad (4.5)$$

O erro resultante deste processo de codificação preditiva é por isso igual a q_n , o erro de quantização de e_n . Como já foi referido, o valor médio do sinal diferença e_n deverá ser bastante inferior a x_n (se uma estimativa adequada for utilizada). Isto significa que, para o mesmo erro quadrático médio, e_n requer menos bits que x_n , permitindo assim obter compressão.

4.2 Codificação por Transformadas

O método de compressão por transformadas efectua uma decomposição espectral da informação no domínio espacial. Para imagens com elevada correlação, muitas das componentes espectrais de maior ordem são pequenas, podendo ser codificadas com uma quantidade de bits reduzida. A operação de transformação pode ser interpretada como produzindo sequências de espectros cujos coeficientes são pouco correlacionados e em que a informação está concentrada num número restrito de coeficientes.

A figura 4.4 tenta ilustrar o processo de decorrelação entre duas amostras adjacentes de

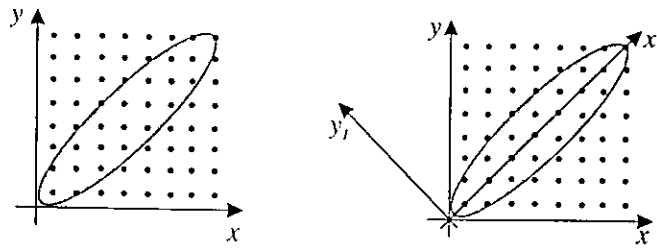


Figura 4.4: Exemplo da decorrelação efectuada por uma transformada

uma imagem, quantificadas em 8 níveis [GW87]. Dada a redundância da imagem, as duas amostras apresentarão, com elevada probabilidade, amplitudes semelhantes — vizinhança da zona $x = y$. Ao efectuar a mudança de coordenadas apresentada, os valores mais prováveis deixarão de estar na vizinhança de $x_1 = y_1$ passando a estar sobre o eixo x_1 . Assim, as variáveis x_1 e y_1 são mais independentes que x e y , isto é, é esperado que y_1 tenha uma amplitude pequena independentemente do valor de x_1 . As propriedades de decorrelação e compactação de informação ficam pois exemplificadas através da figura 4.4:

- o conhecimento do valor de x_1 não fornece qualquer informação sobre o provável valor de y_1 ;
- a variância de x_1 é superior à de y_1 o que corresponde a uma compactação de informação.

A mudança de coordenadas referida na figura 4.4 pode ser alargada a um sistema de dimensão n e corresponde a uma transformação linear. Considerando n amostras agrupadas num vector $\mathbf{x} = (x_0, x_1, \dots, x_{n-1})^t$ e efectuando a transformação linear

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \quad (4.6)$$

em que $\mathbf{A}$ representa a matriz de dimensão $n \times n$ que determina a rotação, obtém-se o vector $\mathbf{y} = (y_0, y_1, \dots, y_{n-1})^t$ no novo sistema de coordenadas.

A utilização de transformadas ortonormais permite obter muito facilmente a transformação inversa

$$\mathbf{x} = \mathbf{B}\mathbf{y}, \quad (4.7)$$

uma vez que $\mathbf{B} = \mathbf{A}^{-1} = \mathbf{A}^t$.

De entre o conjunto de transformadas ortonormais [Cla85], a transformada de Karhunen-Loeve (KL) é óptima já que permite obter coeficientes completamente descorrelacionados e concentrar a informação num número pequenos de coeficientes.

4.2.1 Transformada de Karhunen-Loeve

Considerando um vector $\mathbf{x} = (x_0, x_1, \dots, x_{n-1})^t$, a matriz de covariância associada a esse vector $\mathbf{x}$ é definida como

$$\mathbf{C}_x = E[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^t], \quad (4.8)$$

em que $\bar{\mathbf{x}} = E[\mathbf{x}]$.

A matriz $\mathbf{A}$ que determina a transformada KL é aquela cujas linhas, geralmente designadas por *vectores base*, são os vectores próprios de $\mathbf{C}_x$.

A matriz de covariâncias do vector dos coeficientes da transformada, $\mathbf{y}$, é dada por

$$\mathbf{C}_y = E[\mathbf{A}(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^t \mathbf{A}^t] = \mathbf{A} \mathbf{C}_x \mathbf{A}^t \quad (4.9)$$

que se reduz facilmente a

$$\mathbf{C}_y = \text{diag}(\lambda_0, \lambda_1, \dots, \lambda_{n-1}), \quad (4.10)$$

em que λ_k é o valor próprio associado ao vector próprio $\mathbf{v}_k$.

Uma vez que todos os elementos da matriz $\mathbf{C}_y$, com excepção da diagonal, são nulos, os elementos de $\mathbf{y}$ são não correlacionados. Demonstra-se [Cla85] que a transformada KL é óptima também do ponto de vista de concentração de energia, isto é, se se seleccionarem m entre os n coeficientes, a fracção de energia retida é máxiima em relação a outras transformadas.

Apesar da sua optimalidade, a implementação desta transformada é pouco eficiente uma vez que exige um esforço computacional elevado pouco apropriado para a grande parte das aplicações — cálculo da matriz das covariâncias, determinação dos vectores próprios e finalmente obtenção dos coeficientes da transformada. Além disso, e uma vez que as estatísticas de uma imagem são geralmente não estacionárias, para cada novo bloco de imagem uma nova matriz de transformação tem que ser calculada e transmitida para o decodificador o que acarreta um *overhead* adicional. Esta transformada é, no entanto, útil como referência para avaliar o desempenho de outras transformadas.

Felizmente é possível ultrapassar estas dificuldades utilizando transformadas que podem ser não óptimas. Estas transformadas sub-óptimas utilizam um conjunto de vectores base fixos que permitem rapidez de cálculo e têm desempenhos próximos da transformada KL.

4.2.2 Transformada de Cosenos

A transformada de cosenos (DCT - *Discrete Cosine Transform*), que surgiu pela primeira vez em 1974, tem vindo a ganhar importância no campo do processamento digital de sinal pois aproxima-se da KL em termos de capacidade de descorrelação de sinais e porque existem algoritmos rápidos para o seu cálculo.

Considerando um vector $\mathbf{x} = (x_0, x_1, \dots, x_{n-1})^t$, a DCT que o transforma no vector $\mathbf{y} = (y_0, y_1, \dots, y_{n-1})^t$ é definida como

$$y_k = \alpha_k \sqrt{\frac{2}{n}} \sum_{i=0}^{n-1} x_i \cos \frac{\pi(2i+1)k}{2n}, \quad k = 0, \dots, n-1 \quad (4.11)$$

em que

$$\alpha_k = \begin{cases} \frac{1}{\sqrt{2}} & k = 0 \\ 1 & k = 1, \dots, n-1. \end{cases} \quad (4.12)$$

A transformada inversa (IDCT — *Inverse Discrete Cosine Transform*) é definida como

$$x_i = \sqrt{\frac{2}{n}} \sum_{k=0}^{n-1} \alpha_k y_k \cos \frac{\pi(2i+1)k}{2n}, \quad i = 0, \dots, n-1. \quad (4.13)$$

Da equação (4.11) obtém-se a expressão para os vectores base $\mathbf{t}_k$:

$$\mathbf{t}_k = \alpha_k \sqrt{\frac{2}{n}} \cos \frac{\pi(2i+1)k}{2n}, \quad i = 0, \dots, n-1. \quad (4.14)$$

A análise da figura 4.5 permite concluir que os vectores base da DCT são sinusoides de frequência crescente com k . Uma vez que $\mathbf{t}_0^t = \frac{1}{\sqrt{n}}(1, 1, \dots, 1)$, o primeiro coeficiente da transformada, y_0 , obtido por

$$y_0 = \mathbf{t}_0^t \mathbf{x}, \quad (4.15)$$

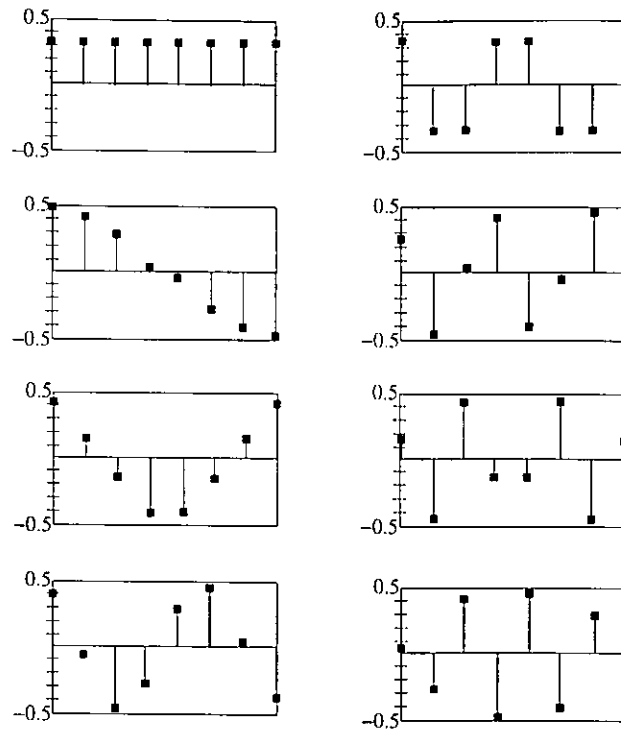


Figura 4.5: Vectores Base para uma DCT de 8 elementos

representa a média do vector de entrada, sendo normalmente referido como *coeficiente DC* da transformada. Os restantes elementos, de média nula, são designados, por analogia, por *coeficientes AC*.

A utilização da DCT em compressão de imagem exige, geralmente, a passagem para a versão bidimensional da transformada (2D-DCT) de forma a explorar as correlações horizontal e vertical entre pixels. A 2D-DCT é definida como

$$y_{k,l} = \alpha_k \alpha_l \frac{2}{n} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} x_{i,j} \cos \frac{\pi(2i+1)k}{2n} \cos \frac{\pi(2j+1)l}{2n}, \quad k, l = 0, \dots, n-1, \quad (4.16)$$

em que

$$\alpha_k = \alpha_l = \begin{cases} \frac{1}{\sqrt{2}} & k = l = 0 \\ 1 & k \neq 0, l \neq 0. \end{cases} \quad (4.17)$$

A transformada inversa bidimensional (2D-IDCT) é obtida através da seguinte expressão:

$$x_{i,j} = \frac{2}{n} \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} \alpha_k \alpha_l y_{k,l} \cos \frac{\pi(2i+1)k}{2n} \cos \frac{\pi(2j+1)l}{2n}, \quad i, j = 0, \dots, n-1. \quad (4.18)$$

Alternativamente, é possível obter a transformação de um bloco de amostras através da aplicação sucessiva de transformadas unidimensionais às linhas e às colunas. O primeiro passo permite explorar a correlação horizontal existente na imagem, sendo o mesmo resultado conseguido na segunda fase no que diz respeito à correlação vertical.

A existência de algoritmos rápidos para o cálculo desta transformada, juntamente com o facto de esta se aproximar, em termos de descorrelação e compactação de energia, da KLT, tornaram-na uma alternativa útil em grande parte das aplicações.

4.3 Quantificação

Um quantificador é um dispositivo utilizado para diminuir, à saída de um sistema de codificação, o número de elementos do conjunto formado pelos diferentes valores possíveis de uma amostra ou coeficiente. Um dos processos para obter este resultado é a divisão do intervalo formado pelas entradas num conjunto de sub-intervalos a que se associam níveis de reconstrução e para os quais os valores à entrada do quantificador são forçados. Este processo, sendo irreversível, é o responsável pela degradação imposta pelo método de compressão, que pode ser minimizada através de uma adequada especificação do quantificador.

A construção de um quantificador envolve a especificação de um conjunto de níveis de decisão d_j , que limitam os sub-intervalos referidos, e de um conjunto de níveis de reconstrução, r_j , de tal modo que se uma amostra x_n é tal que

$$d_j \leq x_n < d_{j+1}, \quad (4.19)$$

então é quantificada para o nível de reconstrução r_j .

Os níveis de decisão e reconstrução são geralmente escolhidos de forma a minimizar uma qualquer medida de distorção $\mathcal{D}$ entre x_n e $\hat{x}_n$, em que $\hat{x}_n$ representa o valor da amostra x_n depois do processo de quantificação. Por ser muito simples de tratar, o erro quadrático médio (MSE - *Mean Square Error*) é uma medida bastante utilizada e, para um sistema com N níveis de quantificação, é definido como

$$\mathcal{D} = E[(x_n - \hat{x}_n)^2] = \sum_{j=1}^N \int_{d_j}^{d_{j+1}} (x_n - r_j)^2 p_{x_n}(x_n) dx_n, \quad (4.20)$$

em que $p_{x_n}(x_n)$ é a função densidade de probabilidade de x_n . A determinação dos valores óptimos para r_j e d_j envolve a minimização da função $\mathcal{D}$ que resulta nas seguintes expressões:

$$d_j = \frac{r_j + r_{j-1}}{2} \quad (4.21)$$

$$r_j = \frac{\int_{t_j}^{t_{j+1}} x_n p_{x_n}(x_n) dx_n}{\int_{t_j}^{t_{j+1}} p_{x_n}(x_n) dx_n}. \quad (4.22)$$

Max [Max60] apresentou a solução para este problema, para variáveis aleatórias com distribuição Gaussiana, e calculou tabelas de valores óptimos como função do número de níveis de quantificação. Uma vez que a função densidade de probabilidade do sinal não é, em geral, conhecida e também porque a determinação do quantificador óptimo, também conhecido como *Lloyd-Max* [Max60], [Loy82], envolve a resolução, por métodos numéricos, das equações (4.21) e (4.22), é vulgar recorrer a quantificadores uniformes para aplicações práticas. Nestes quantificadores, a distância entre dois níveis de decisão consecutivos d_j e d_{j+1} é constante, o que implica, pela análise da equação (4.21), níveis de reconstrução também igualmente espaçados (figura 4.6). O quantificador assim definido não é mais do que um quantificador de *Lloyd-Max* para distribuições uniformes e a sua aplicação, de forma a obter $\hat{x}_n$, exige apenas uma operação de divisão do valor da amostra x_n pelo passo de quantificação $\Delta = d_j - d_{j-1}$.

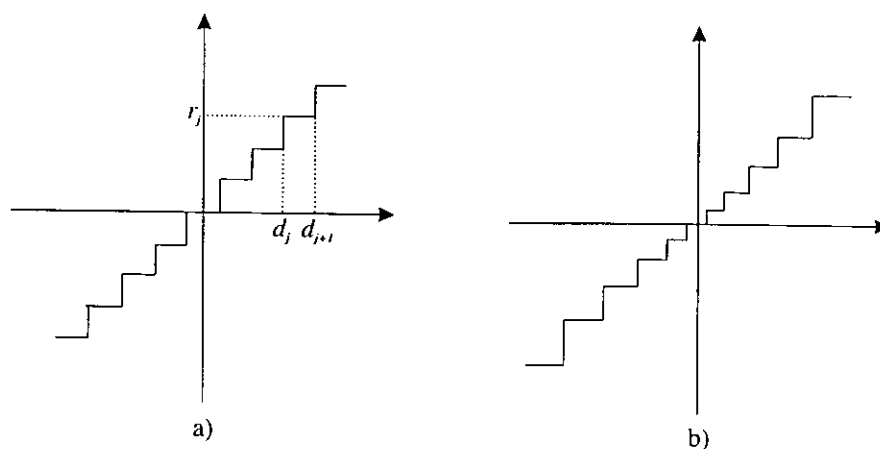


Figura 4.6: Quantificador a) uniforme b) não uniforme

A quantificação de um erro de predição num sistema DPCM afecta relativamente poucas amostras do sinal reconstruído, enquanto que a quantificação de um coeficiente resultante de uma transformada afecta todas as amostras do bloco. Esta propagação do erro de quan-

tificação às amostras vizinhas conduz, geralmente, a técnicas de construção do quantificador no domínio espectral bastante diferentes das utilizadas nos quantificadores DPCM. A sensibilidade da visão não é igual para os diferentes coeficientes resultantes de uma transformada, sendo inferior para frequências mais elevadas. Este aspecto pode ser aproveitado para quantificar com mais precisão o coeficiente DC, cujo efeito de degradação se reflecte em descontinuidades visíveis entre blocos adjacentes, principalmente em zonas de pouco detalhe, e utilizar menos bits para os coeficientes AC correspondentes às frequências mais elevadas. Uma redução elevada no número de níveis de quantificação nestes coeficientes irá reflectir-se numa perda de qualidade nas zonas de transições rápidas.

A solução clássica para o problema de alocação de bits a cada um dos coeficientes, sujeita a um número máximo pré-definido B e que pode ser definida como [GG92]

- determinar b_i para $i = 1, \dots, N$ de forma a minimizar uma medida de distorção $\mathcal{D} = \sum_{i=1}^N W_i(b_i)$, em que $W_i(b_i)$ representa o erro resultante de quantificar o coeficiente i com b_i bits, sujeito a $\sum_{i=1}^N b_i \leq B$,

relaxa a restrição $b_i \geq 0$ e obtém a solução óptima que permite valores negativos de b_i . A solução é depois modificada por processos *ad hoc* que eliminam os valores negativos e ajustam a solução inicial para obter resultados inteiros.

Várias soluções para o problema de alocação de bits são apresentadas por diferentes autores [Cla85],[KR81], sendo a mais divulgada a apresentada em [HS63]:

$$b_i = \frac{B}{n} + \frac{1}{2} \left[\log_2 \sigma_i^2 - \frac{1}{n} \sum_{j=1}^n \log_2 \sigma_j^2 \right]. \quad (4.23)$$

4.4 Códigos de Comprimento Variável

À última etapa de um sistema de compressão está, geralmente, associado um processo de codificação que não introduz perdas suplementares. Este processo explora as propriedades estatísticas dos dados a codificar e tem como base a *Teoria da Informação*.

Supondo uma fonte que gera um conjunto discreto de L símbolos independentes r_k (no caso de uma imagem monocromática, por exemplo, r_k poderá representar níveis de cinzento), com probabilidades $p_k, k = 1, \dots, L$, então a informação associada a r_k é definida como

$$I_k = -\log_2 p_k. \quad (4.24)$$

Uma vez que

$$\sum_{k=1}^L p_k = 1 \quad (4.25)$$

então $p_k \leq 1$ e $I_k \geq 0$. Pela equação (4.24) verifica-se que a informação contida num símbolo (que pode ser uma amplitude, um coeficiente, etc.) é uma função da sua probabilidade de ocorrência. Mais ainda, que quanto mais provável for esse símbolo menor a informação nele contida.

A entropia de um sinal é definida como sendo a informação média nele contida, isto é,

$$H = - \sum_{k=1}^L p_k \log_2 p_k. \quad (4.26)$$

Para uma imagem digital, considerada uma fonte de pixels independentes, a entropia pode ser estimada recorrendo a um histograma de ocorrências. Para um dado L , a entropia de um sinal é máxima no caso de uma distribuição uniforme, isto é, $p_k = 1/L, k = 1, \dots, L$. Nesse caso

$$\max_{p_k} H = - \sum_{k=1}^L \frac{1}{L} \log_2 \frac{1}{L} \quad (4.27)$$

ou

$$\max_{p_k} H = \log_2 L. \quad (4.28)$$

Por outro lado, se $p_1 = 1, p_2 = p_3 = \dots = p_L = 0$ então $H = 0$.

Conclui-se pois que a entropia para L variáveis aleatórias pode variar entre 0 e $\log_2 L$:

$$0 \leq H \leq \log_2 L. \quad (4.29)$$

A entropia representa portanto a quantidade de informação associada a uma fonte e impõe um limite mínimo para o número médio de bits necessários para codificar um sinal. De acordo com os teoremas apresentados por Shannon [Gal68], é possível codificar, sem distorção, um sinal de entropia H utilizando $H + \epsilon$ bits em que $\epsilon \geq 0$ é uma quantidade infinitesimal. Ou, de uma forma alternativa: *é impossível codificar, sem distorção, um sinal com menos de H bits* (equação (4.30)),

$$b \geq H. \quad (4.30)$$

O conceito de entropia fornece então um critério que pode ser utilizado para medir o desempenho de um codificador particular. Para um codificador com símbolos $s_1, \dots, s_L$, de comprimento respectivamente $\beta_1, \dots, \beta_L$, o número médio de bits necessários é:

$$b = \sum_{k=1}^L \beta_k p_k. \quad (4.31)$$

Se b se aproximar de H , então o codificador é aproximadamente ótimo. Caso contrário, se houver uma diferença significativa entre estes dois valores, será sub-ótimo.

A tentativa de minimizar o valor b , referido na equação (4.31), origina um código de comprimento variável em que os símbolos mais prováveis são representados por palavras de menor comprimento. Estes códigos devem ter uma estrutura tal que permita, na descodificação, mapear de forma única cada palavra de código num símbolo. Se considerarmos o exemplo da tabela 4.1, em que são apresentados dois códigos para representar um mesmo conjunto de símbolos $\mathcal{S} = \{s_1, s_2, s_3\}$, verifica-se que $\mathcal{C}_1$ torna a descodificação ambígua. De facto, a sequência 0101010 pode ser interpretada pelo descodificador como sendo $s_1 s_2 s_2 s_2$ ou, por exemplo, como $s_1 s_3 s_0 s_1$.

Tabela 4.1: Códigos de comprimento variável

Símbolo	$\mathcal{C}_1$	$\mathcal{C}_2$
s_1	0	0
s_2	10	10
s_3	101	110

Uma condição suficiente para satisfazer a exigência de descodificação única é que nenhuma palavra de código seja o prefixo de outra palavra de maior comprimento. Prova-se [GG92] que a satisfação da *desigualdade de Kraft* referida na equação 4.32 é uma condição suficiente para a existência de descodificação única.

$$\sum_{k=1}^L 2^{-\beta_k} \leq 1. \quad (4.32)$$

Esta desigualdade fornece a base para a obtenção dos limites superior e inferior do desempenho de um código:

$$H \leq b \leq H + 1. \quad (4.33)$$

O desempenho descrito por (4.33) pode ser melhorado se, em vez de codificar independentemente cada um dos símbolos, estes forem codificados e transmitidos em blocos de N amostras. Neste caso a entropia de ordem N é definida como

$$\mathbf{H} = - \sum_{k=1}^M \mathbf{p}_k \log_2 \mathbf{p}_k. \quad (4.34)$$

em que M representa o número de vectores possíveis à entrada do codificador e $\mathbf{p}_k$ a probabilidade de ocorrência de cada um desses vectores. Sendo L o número de símbolos independentes possíveis, então $M = 2^{NL}$.

Estendendo agora a equação (4.33) ao caso vectorial, tem-se

$$\mathbf{H} \leq \mathbf{b} \leq \mathbf{H} + 1, \quad (4.35)$$

em que $\mathbf{b}$ representa o número médio de bits por vector.

Uma vez que $b = \mathbf{b}/N$ e $H = \mathbf{H}/N$, então

$$H \leq b \leq H + \frac{1}{N}, \quad (4.36)$$

de onde se conclui que, aumentando o número de símbolos a codificar conjuntamente, se podem obter representações mais próximas da entropia. Na prática, a implementação deste processo pode levantar alguns problemas para vectores de comprimento muito elevado — atrasos introduzidos pelo processamento em bloco e complexidade associada a grandes exigências de memória.

4.4.1 Codificação de Huffman

Uma técnica que satisfaz a regra mencionada, e cujo desempenho se aproxima do óptimo, foi apresentada por Huffman [Huf52]. A obtenção deste código envolve a construção de uma árvore binária utilizando o seguinte algoritmo [Jai89]:

- ordenar os símbolos por ordem decrescente de probabilidade de ocorrência e considerar cada um como sendo um nó de uma árvore:
- enquanto houver mais nós:
 - unir os 2 nós de menor probabilidade de modo a obter um novo nó cuja probabilidade é a soma das probabilidades desses 2;
 - atribuir arbitrariamente os valores 1 e 0 a cada um dos ramos que convergem no novo nó;
- partindo da raiz, percorrer a árvore até cada um dos símbolos originais. A sequência de bits associados a esse caminho dá a palavra de código que representa esse símbolo.

Pela análise das equações (4.26) e (4.31) conclui-se que o codificador óptimo pode ser obtido se todas as probabilidades forem potências negativas de 2, isto é, $p_k = 2^{-\beta_k}$, $k = 1, \dots, L$.

No entanto, é também fácil concluir que, por mais pequena que seja a entropia, uma palavra tem que ser representada por um número inteiro de bits.

De facto, imaginando um caso real de um símbolo com probabilidade $4/5$ [HL92] verifica-se que apesar de, segundo a teoria da informação, este conter apenas $\log_2 \frac{5}{4} \simeq 0.32$ bits de informação, o código de Huffman utiliza 1 bit para representar este acontecimento. Uma vez que a probabilidade de este símbolo ocorrer é elevada, a eficiência do código de Huffman afasta-se do valor óptimo.

Existem, no entanto, processos que permitem ultrapassar algumas das deficiências detectadas neste código. A codificação aritmética, apresentada a seguir, é uma técnica alternativa para obtenção de compressão sem perdas.

4.4.2 Codificação Aritmética

No método de codificação aritmética [BCW90], [PMK91], [HV92] uma mensagem é representada por um intervalo de números reais entre 0 e 1. Quanto maior for a mensagem a codificar, menor o intervalo utilizado para a representar e maior o número de bits necessários para o especificar. Símbolos sucessivos de uma mensagem reduzem o intervalo de acordo com a probabilidade de ocorrência desse símbolo segundo o modelo utilizado. Símbolos mais prováveis reduzem menos o intervalo comparativamente com os menos prováveis, acrescentando assim menos bits à mensagem.

O processo descrito exige o conhecimento das probabilidades de cada um dos símbolos. O modelo de probabilidades pode ser fixo, semi-adaptativo, o que exige uma contagem prévia

dos símbolos para obter o histograma das frequências, ou puramente adaptativo, caso em que as probabilidades são estimadas dinamicamente. Num modelo adaptativo consideram-se inicialmente probabilidades iguais para todos os símbolos, reflectindo este facto a não existência de informação prévia. Estas probabilidades são, no entanto, actualizadas com a ocorrência de cada novo símbolo de modo a reflectir as frequências observadas até então. Desde que os codificadores e decodificadores tenham o mesmo valor inicial e o mesmo algoritmo de actualização, os seus modelos permanecerão equivalentes ao longo de toda a operação. O codificador recebe o próximo símbolo, codifica-o e actualiza o modelo. O decodificador identifica-o de acordo com o modelo corrente e em seguida actualiza o modelo. A alteração do modelo utilizado corresponde pois a uma actualização das frequências de ocorrência de cada um dos símbolos, um processo bastante mais simples que a actualização da árvore binária num processo de codificação de Huffman adaptativo. Esta característica e a não exigência de um número inteiro de bits para representar cada símbolo representam vantagens da codificação aritmética comparativamente aos códigos de Huffman.

O algoritmo básico é o seguinte [HV92]:

- Inicializar o intervalo actual identificado por $[L, H)$ com o valor $[0, 1)$;
- Enquanto houver símbolos a codificar:
 - dividir o intervalo actual em sub-intervalos proporcionais à probabilidade de cada um dos símbolos possíveis, de acordo com o modelo utilizado;
 - seleccionar o sub-intervalo correspondente ao símbolo a codificar. Este passa a ser o intervalo actual;
- Identificar os limites do sub-intervalo actual de forma a poder distingui-lo de qualquer outro intervalo possível (de facto a identificação do intervalo pode ser feita através de um qualquer valor pertencente a esse intervalo, não sendo necessário fornecer os limites superiores e inferiores do intervalo).

A descrição acima apresentada é uma versão simplista do algoritmo. Problemas como precisão aritmética, identificação de fim de mensagem e atrasos introduzidos pela codificação de toda a informação antes da transmissão, têm que ser considerados. Um modo de operação incremental permite a utilização de aritmética em vírgula fixa eliminando os problemas de precisão e atrasos. A utilização de um símbolo especial permite a identificação de fim de mensagem.

Na figura 4.7 apresenta-se um exemplo de codificação de uma mensagem partindo, para simplificar, de uma distribuição de probabilidades fixa definida na tabela 4.2. Inicialmente, todo o intervalo limitado pelos valores 0 e 1 está disponível para a codificação da mensagem mas, à

Tabela 4.2: Probabilidades e intervalos atribuídos no processo de codificação aritmética

Símbolo	Probabilidade	Intervalo
s_1	0.2	$[0, 0.2)$
s_2	0.2	$[0.2, 0.4)$
s_3	0.3	$[0.4, 0.7)$
s_4	0.2	$[0.7, 0.9)$
s_5	0.1	$[0.9, 1.0)$

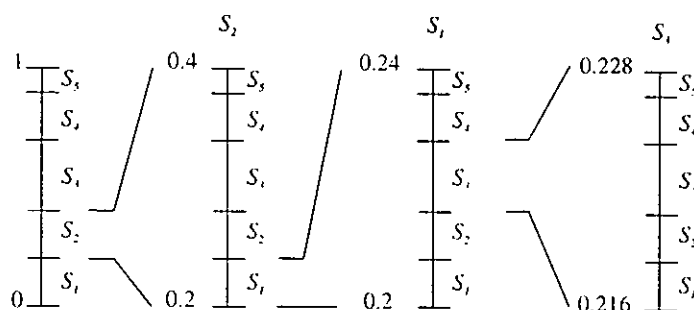


Figura 4.7: Exemplo de um codificador aritmético

medida que cada símbolo é processado, esse intervalo é reduzido para uma porção proporcional à probabilidade do símbolo. A codificação de s_2 provoca uma redução do intervalo de $[0, 1)$ para $[0.2, 0.4)$, que corresponde à percentagem da probabilidade de s_1 sobre um intervalo de dimensão 1. Esta passa a ser a nova área a considerar quando se pretende continuar o processo. s_1 reduz também para $1/5$ ($P(s_2) = 0.2$) o novo intervalo, impondo os limites inferior e superior de 0.2 e 0.24, respectivamente. Finalmente, a codificação do símbolo s_3 implica o mapeamento do intervalo original que lhe foi atribuído, $[0.4, 0.7)$, no intervalo $[0.2, 0.24)$ originando, no fim da mensagem, o intervalo $[0.216, 0.228)$.

4.5 Quantificação Vectorial

As técnicas de codificação convencionais apresentadas nas secções 4.1 e 4.2 apresentam uma deficiência comum pelo facto de a quantificação ser aplicada a amostras (ou coeficientes) individuais. O processo de redução de redundância é efectuado, no caso da codificação por transformadas, sobre um vector de amostras mas com aplicação posterior de um quantificador escalar. No caso da codificação por predição, a quantificação é realizada também sobre uma grandeza escalar que representa um erro entre uma amostra e a sua estimativa. Estas técnicas são sub-óptimas uma vez que os valores processados são ainda correlacionados e

dependentes. Segundo a teoria apresentada por Shannon, melhores resultados podem sempre ser obtidos, em teoria, codificando vectores em vez de escalares. Estes esquemas que envolvem a codificação de vectores ficaram conhecidos por *Quantificação Vectorial* — VQ (*Vector Quantization*) [Gra84], [NK88], [GG92].

Um quantificador Q , de dimensão k e tamanho N , efectua um mapeamento de um vector do espaço Euclidiano k -dimensional $\mathcal{R}^k$ num conjunto finito $\mathcal{C}$ contendo N palavras de código, isto é,

$$Q : \mathcal{R}^k \rightarrow \mathcal{C}, \quad (4.37)$$

em que $\mathcal{C} = (\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \dots, \hat{\mathbf{x}}_N)$ e $\hat{\mathbf{x}}_i \in \mathcal{R}^k$.

O conjunto $\mathcal{C}$, normalmente designado por tabela de códigos, tem N elementos distintos sendo cada um deles um vector em $\mathcal{R}^k$. Associada a um quantificador vectorial de tamanho N está uma partição de $\mathcal{R}^k$ em N regiões ou células. A i -ésima célula, R_i , é definida como sendo

$$R_i = \{\mathbf{x} \in \mathcal{R}^k : Q(\mathbf{x}) = \hat{\mathbf{x}}_i\} \quad (4.38)$$

$$\bigcup_{i=1}^N R_i = \mathcal{R}^k \text{ e } R_i \cap R_j = \emptyset \text{ para } i \neq j. \quad (4.39)$$

O modelo matemático que identifica o codificador pode ser representado por:

$$Q(\mathbf{x}) = \sum_{i=1}^N \hat{\mathbf{x}}_i S_i(\mathbf{x}). \quad (4.40)$$

em que $S_i(\mathbf{x})$ é a função

$$S_i(\mathbf{x}) = \begin{cases} 1 & \text{se } \mathbf{x} \in R_i \\ 0 & \text{nos outros casos.} \end{cases} \quad (4.41)$$

O decodificador, por sua vez, tem apenas que consultar uma tabela para obter a palavra associada ao índice i que lhe é fornecido.

O número médio de bits por amostra para um quantificador com estas características é

$$b = \frac{\log_2 N}{k}. \quad (4.42)$$

É importante notar que, para k fixo, o número de bits necessários é determinado pelo tamanho N da tabela de códigos e não pelo número de bits utilizado para especificar os vectores armazenados na tabela. Aumentando o valor de k é possível obter compressões mais elevadas com o prejuízo de um aumento de complexidade do processo. Outro aspecto de interesse é que b não tem necessariamente que ser um valor inteiro ao contrário do que se passa com a quantificação escalar (secção 4.3).

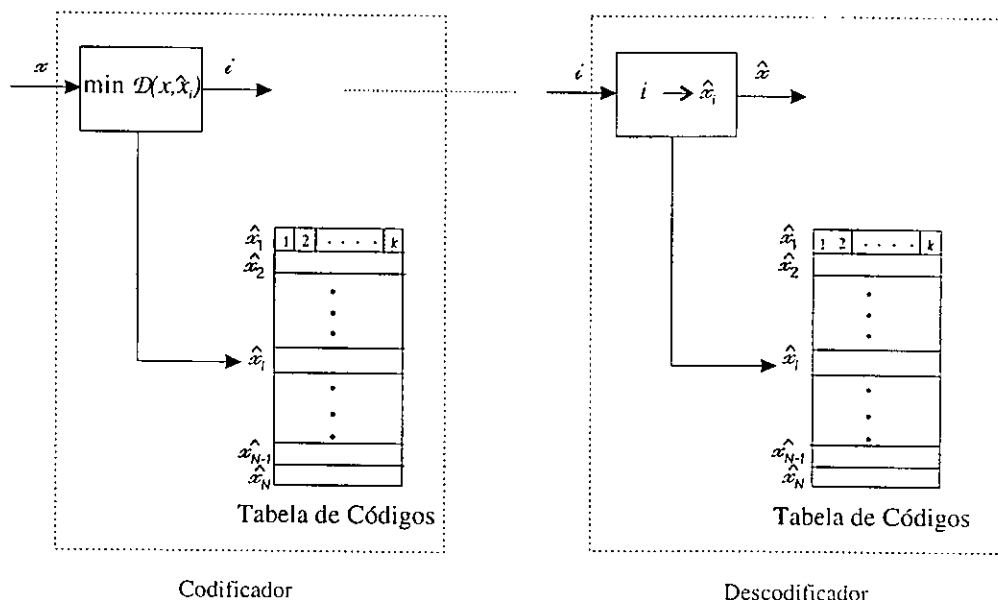


Figura 4.8: Diagrama de blocos de um quantificador vectorial

O processo de VQ pode ser visto como uma combinação de duas funções: um codificador que recebe à entrada um vector $\mathbf{x}$ e gera um índice i que identifica uma palavra da tabela de códigos $\mathcal{C}$, e um decodificador que utiliza esse índice para gerar uma reprodução $\hat{\mathbf{x}}$ do vector $\mathbf{x}$ (figura 4.8). A operação de mapeamento, efectuada no codificador, exige a definição de uma medida de distorção $\mathcal{D}(\mathbf{x}, \hat{\mathbf{x}})$, que representa o custo associado a reproduzir $\mathbf{x}$ como $\hat{\mathbf{x}}$. O melhor resultado é aquele que minimiza $\mathcal{D}(\mathbf{x}, \hat{\mathbf{x}})$. Várias medidas de distorção podem ser utilizadas sendo as mais usadas na prática as seguintes:

- *norma l* , definida como

$$\mathcal{D}(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^l, \quad (4.43)$$

em que l é um valor inteiro positivo. Entre os mais comuns encontram-se os valores de $l = 1$ (distorção igual à soma dos erros absolutos), $l = 2$ (erro quadrático médio) e

$l = \infty$ (erro máximo absoluto – a distorção é determinada pela componente do vector de erro que contribui com o maior erro absoluto);

- erro quadrático pesado

$$\mathcal{D}(\mathbf{x}, \hat{\mathbf{x}}) = (\mathbf{x} - \hat{\mathbf{x}})\mathbf{W}(\mathbf{x} - \hat{\mathbf{x}}) \quad (4.44)$$

em que $\mathbf{W}$ é uma matriz simétrica e definida positiva. Esta medida inclui o erro quadrático no caso particular em que $\mathbf{W} = \mathbf{I}$. No caso de $\mathbf{W}$ ser uma matriz diagonal com valores $w_{ii} \geq 0$ obtém-se um erro quadrático em que cada uma das componentes do vector contribui com um peso diferente para o erro total permitindo assim obter resultados subjectivos de melhor qualidade.

Uma vez definida uma medida de distorção, o algoritmo utilizado para obter o índice i pode ser descrito através dos seguintes passos:

- fazer $d = d_0$, $j = 1$ e $i = 1$, em que d_0 deve ser superior a qualquer valor de distorção que se possa obter com a tabela de códigos a utilizar (na prática é costume utilizar o maior valor positivo suportado pelo processador);
- enquanto $j \leq N$ fazer
 - calcular $\mathcal{D}_j = \mathcal{D}(\mathbf{x}, \hat{\mathbf{x}}_j)$
 - se $\mathcal{D}_j < d$, então fazer $d = \mathcal{D}_j$ e $i = j$
 - fazer $j = j + 1$
- o resultado é o índice i .

Este método, normalmente designado por *regra do vizinho mais próximo*, envolve uma pesquisa exaustiva em toda a tabela pelo que pode tornar-se computacionalmente muito pesado. Existem, no entanto, algoritmos propostos por diferentes autores que permitem otimizar este processo [HBSH92],[Mat92], [GG92],[Gra84], [SM89].

Um dos problemas base do processo de VQ envolve a criação da tabela de códigos a utilizar. O algoritmo normalmente referido como LBG [LBG80] recorre a uma técnica iterativa que permite, partindo de uma tabela inicial, melhorar sucessivamente (de acordo com uma medida de distorção) a tabela para uma dada sequência de entrada, designada por sequência de treino:

- inicializar a constante $\epsilon \geq 0$ correspondente ao limiar de decisão para o valor de distorção; inicializar o número de iterações $m = 0$; definir uma tabela de códigos inicial $\mathcal{C}_0$ e uma sequência de treino $\mathcal{T} = \{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_n\}$;

- enquanto $(\mathcal{D}_{m-1} - \mathcal{D}_m)/\mathcal{D}_m \geq \epsilon$
 - dada a tabela $\mathcal{C}_m$, dividir a sequência de treino em partições R_i utilizando a regra do vizinho mais próximo

$$R_i = \{\mathbf{t} \in \mathcal{T} : \mathcal{D}(\mathbf{t}, \mathbf{x}_i) \leq \mathcal{D}(\mathbf{t}, \mathbf{x}_j) \forall j \neq i\} \quad (4.45)$$

- para cada R_i calcular a *centroide* para obter $\mathcal{C}_{m+1} = \{\text{cent}(R_i)\}$.

Existem inúmeras outras técnicas para a construção da tabela de códigos tais como os *Mapas de Kohonen* [Koh90] e as *Máquinas de Boltzmann (simulated annealing)* [HSA84], [KGV83]. Kohonen introduziu uma técnica de agrupamento baseada em redes neuronais em que os pesos entre neurónios são adaptados de acordo com os padrões de entrada. Os valores, para que convergem assintoticamente os pesos, definem a tabela de códigos. *Simulated annealing* é uma técnica de optimização estocástica que pode ser utilizada para determinar a tabela óptima para a sequência de treino pois evita os mínimos locais introduzindo ruído no sistema.

4.6 Codificação por Sub-Bandas

O princípio básico da codificação por sub-bandas é a divisão do espectro do sinal em várias bandas de frequência cada uma das quais codificada e transmitida (armazenada) separadamente. Isto é particularmente interessante no caso de codificação de imagens uma vez que os espectros são, geralmente, não uniformes e também porque a tolerância ao ruído resultante da compressão é dependente da frequência. Com esta divisão é possível ajustar a distorção de acordo com um critério perceptual e, uma vez que a imagem é processada sem recorrer a uma divisão em blocos, a distorção que ocorre nos algoritmos baseados em VQ e transformadas, designada por *efeito de bloco*, é também eliminada. Outra das desvantagens da codificação em blocos — a não exploração da correlação entre blocos — deixa também de existir.

As diferentes bandas são obtidas utilizando um conjunto de filtros passa-banda. Uma vez que a banda de cada uma das sub-imagens é reduzida, elas podem ser sub-amostradas de forma a obter imagens de dimensão inferior que em seguida são quantificadas e codificadas para serem transmitidas. Na decodificação, cada uma das sub-imagens é reconstruída para a dimensão original e, após a passagem por filtros, são adicionadas de forma a obter a imagem final (figura 4.9).

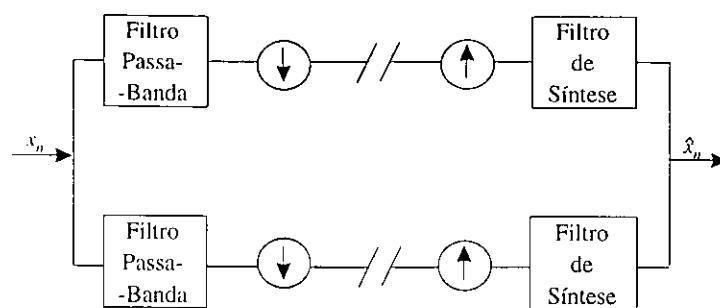


Figura 4.9: Decomposição de uma imagem em bandas de frequências, sub-amostragem e reconstrução

Capítulo 5

Aplicação Fotográfica

A introdução de imagens de qualidade fotográfica no serviço de videotex envolve o armazenamento e a transmissão de grandes quantidades de informação. Uma imagem de 24 bits e dimensões 640x480 pixels, exigiria quase 1 Mbyte de capacidade de armazenamento no servidor e aproximadamente 2 minutos de tempo de transmissão num canal B. Estes problemas sugerem a utilização de algoritmos de compressão eficientes mas que, simultaneamente, mantenham a qualidade necessária a este serviço. No capítulo 4 foram apresentadas várias técnicas de compressão que são utilizadas nos campos mais diversos. No entanto, e uma vez que no serviço público de videotex se pretende obter compatibilidade e interligação de vários sistemas, a tecnologia de compressão por si só não é suficiente. Este tipo de exigências só pode ser satisfeito pela definição e utilização de técnicas normalizadas de compressão de imagem. Nas aplicações fotográficas para videotex adoptou-se um algoritmo normalmente designado por JPEG (*Joint Photographic Experts Group*) [JPEG].

5.1 Algoritmo JPEG

Este algoritmo pretende ser genérico e suportar uma grande variedade de aplicações. De forma a cobrir diferentes necessidades, inclui dois métodos básicos de compressão, cada um com vários modos de operação: um método baseado na DCT (secção 4.2.2), para compressão com perdas, e outro baseado em codificação por predição (secção 4.1), que permite uma reprodução exacta da imagem original.

Os modos de operação são os seguintes:

- **Sequencial** — cada componente da imagem é codificada num só varrimento da esquerda para a direita e de cima para baixo;

- **Progressivo** — a imagem é codificada em vários varrimentos de forma a permitir, na reconstrução, uma melhoria progressiva da qualidade da imagem;
- **Por Predição** — a imagem é comprimida de maneira a garantir uma reconstrução exacta;
- **Hierárquico** — a imagem é codificada em múltiplas resoluções para permitir o acesso à versão de baixa resolução sem primeiro ser necessário descomprimir a imagem com resolução total.

Para cada um dos métodos referidos, uma ou mais variantes podem ser definidas de acordo com a precisão das amostras e o tipo de código de comprimento variável utilizado (secção 4.4).

5.1.1 Método baseado na DCT

A figura 5.1 ilustra os passos chave do método de compressão baseado na DCT para cada componente de uma imagem. No caso de imagens policromáticas, cada componente pode ser codificada de uma só vez ou, alternativamente, cada bloco de uma componente multiplexado com blocos de outras componentes.

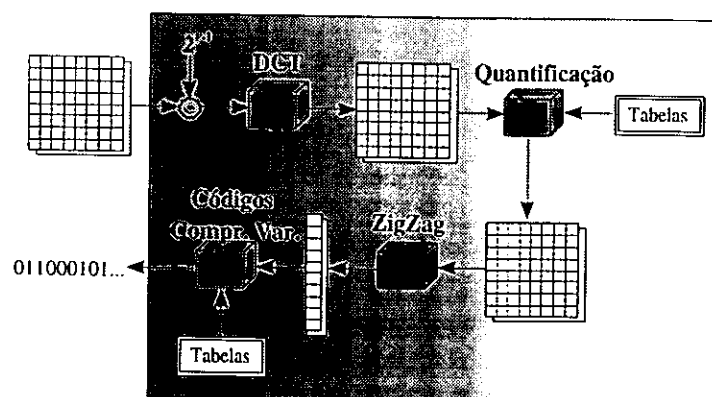


Figura 5.1: Método de compressão baseado na DCT

Basicamente, este método recorre a uma divisão da imagem em blocos de 8x8 amostras, aos quais é aplicada a transformada seguida de quantificação e de um código de comprimento variável. No modo progressivo, é necessário armazenar temporariamente a imagem depois do processo de transformação para o domínio espectral, de forma a permitir uma divisão em várias bandas de frequência (secção 5.1.3).

5.1.1.1 DCT e IDCT 8x8

À entrada do codificador, a imagem é dividida em blocos de 8x8 amostras e convertida de uma representação sem sinal no intervalo $[0, 2^P - 1]$ (em que P representa a precisão, em bits, das amostras) para uma representação com sinal no intervalo $[-2^{P-1}, 2^{P-1} - 1]$. A transformada de cosenos é então aplicada a cada bloco utilizando a equação (4.11) simplificada para este caso:

$$y_{k,l} = \alpha_k \alpha_l \frac{1}{4} \sum_{i=0}^7 \sum_{j=0}^7 x_{i,j} \cos \frac{\pi(2i+1)k}{16} \cos \frac{\pi(2j+1)l}{16}, \quad k, l = 0, \dots, 7. \quad (5.1)$$

No decodificador, a IDCT realiza o processo inverso e reconstrói um vector de 64 elementos através da expressão

$$x_{i,j} = \frac{1}{4} \sum_{k=0}^7 \sum_{l=0}^7 \alpha_k \alpha_l y_{k,l} \cos \frac{\pi(2i+1)k}{16} \cos \frac{\pi(2j+1)l}{16}, \quad i, j = 0, \dots, 7. \quad (5.2)$$

Matematicamente, esta operação é biunívoca pelo que as amostras resultantes da transformada inversa são exactamente as originais no caso de não existir quantificação. No entanto, em aplicações práticas, é impossível obter implementações com precisão exacta. Vários algoritmos que calculam, aproximadamente, as transformadas de coseno directa e inversa podem ser obtidos em [RY89].

Para os modos de operação baseados na DCT, a norma JPEG especifica duas precisões — 8 e 12 bits por amostra (por componente) — que exigem capacidades de cálculo diferentes para obter a exactidão desejada.

5.1.1.2 Quantificação

À saída da DCT, os coeficientes são quantificados uniformemente (secção 4.3) utilizando uma *Tabela de Quantificação* que deve ser especificada pela aplicação como entrada do codificador. Cada um dos elementos dessa tabela é um inteiro entre 1 e 255 que indica o passo de quantificação para o coeficiente correspondente. O processo de quantificação é definido como uma divisão de cada coeficiente da DCT pelo correspondente passo de quantificação, seguido de um arredondamento para o inteiro mais próximo. Na descodificação, esta normalização é retirada, antes de efectuar a IDCT, através da multiplicação pelo mesmo passo de quantificação.



5.1.1.3 Códigos de Comprimentos Variável

A fase final do método de compressão baseado na DCT utiliza códigos de comprimentos variável. A compactação de informação resultante da DCT, juntamente com a utilização de quantificadores diferentes para cada um dos coeficientes, dependendo da sua posição no bloco, sugere uma ordenação dos coeficientes em *zig-zag* (figura 5.2). Esta ordenação facilita a utilização de códigos de comprimento variável ao colocar os coeficientes de baixa frequência antes dos de alta frequência que, com maior probabilidade, serão nulos.

Uma vez que existe uma correlação elevada entre coeficientes DC de blocos adjacentes, estes são codificados como valor diferença em relação ao bloco anterior, isto é, $Diff = DC_i - DC_{i-1}$. Este tratamento especial deve-se ao facto de o coeficiente DC conter, geralmente, uma fracção significativa da energia total da imagem.

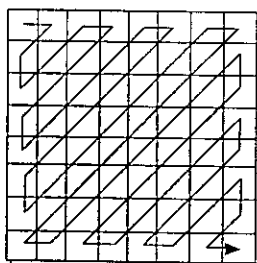


Figura 5.2: Ordenação dos coeficientes em *zig-zag*

Antes da utilização de códigos de Huffman, cada um dos coeficientes AC diferente de zero é representado como uma combinação do seu valor e do número de coeficientes AC nulos que o precedem na ordenação em *zig-zag*. Esta combinação é representada por um par de símbolos: $S_1 = (Z, B)$ e $S_2 = (A)$ em que A é simplesmente a amplitude do coeficiente; Z ($Z \in [0, 15]$) é o número de coeficientes AC nulos consecutivos que precedem o coeficiente não nulo a ser codificado; e B representa o número de bits utilizados para codificar A , isto é, para codificar o símbolo S_2 utilizando a tabela de códigos de Huffman fornecida. A extensão deste código, para sequências de zeros superiores a 15, é possível utilizando o símbolo especial $S_1 = (15, 0)$ que representa uma sequência de 16 coeficientes nulos. A repetição deste símbolo permite codificar qualquer sequência de zeros dentro de um bloco. Para os casos em que a última sequência de zeros inclui o último coeficiente, foi criado um símbolo especial — EOB (*End Of Block*) — que pode ser visto como um símbolo que termina o bloco que se está a codificar.

A gama dinâmica dos coeficientes AC quantificados determina a gama de valores que A e B devem permitir representar. A análise da equação de uma DCT 8x8 mostra que, se as

64 amostras na entrada têm uma precisão de N bits, então a parte inteira dos coeficientes AC à saída da transformada tem, no máximo, $N + 3$ bits. Este é também o valor máximo dos coeficientes quantificados que corresponde à utilização de um quantificador com passo de quantificação igual a 1. Particularizando para o caso de amostras de 8 bits no intervalo $[-2^7, 2^7 - 1]$, a amplitude dos coeficientes AC quantificados tem valores no intervalo $[-2^{10}, 2^{10} - 1]$. S_2 representa, por isso, amplitudes até 10 bits o que define também a gama de valores B .

A representação dos coeficientes DC, antes da codificação de Huffman, faz-se de uma forma semelhante eliminando, no entanto, o valor Z do símbolo S_1 . Uma vez que este coeficiente é codificado diferencialmente, a sua gama dinâmica é acrescida passando a estar no intervalo $[-2^{11}, 2^{11} - 1]$.

Efectuado este processo, S_1 é codificado com um código de comprimento variável da tabela de Huffman especificada para a componente da imagem correspondente. A codificação de S_2 é também feita através de um código de comprimento variável mas que não corresponde a um código de Huffman. Uma diferença fundamental entre estes dois códigos é que, ao contrário de um código de Huffman em que o seu comprimento não é conhecido até ele ser decodificado, o número de bits utilizado para codificar S_2 é especificado em S_1 .

A norma JPEG permite também a utilização de codificação aritmética (secção 4.4.2) a substituir os códigos de Huffman.

5.1.2 Método de Codificação por Predição

O método de compressão sem perdas (figura 5.3) é baseado em técnicas muito simples de predição (secção 4.1). Para cada amostra x da imagem, é calculada uma estimativa baseada nas amostras vizinhas (figura 5.4). Esta estimativa é subtraída ao valor da amostra e a diferença é codificada com um dos códigos de comprimento variável referidos. A tabela 5.1 refere os preditores que podem ser usados neste método.

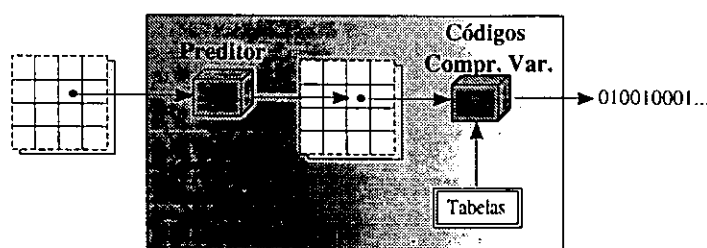


Figura 5.3: Método de compressão por predição

Tabela 5.1: Preditores para o método de compressão sem perdas

Preditor	Estimativa
$\hat{x}_1$	x
$\hat{x}_2$	a
$\hat{x}_3$	b
$\hat{x}_4$	c
$\hat{x}_5$	$a + b - c$
$\hat{x}_6$	$a + (b - c)/2$
$\hat{x}_7$	$b + (a - c)/2$
$\hat{x}_8$	$(a + b)/2$

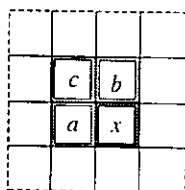


Figura 5.4: Vizinhança de uma amostra para o cálculo da estimativa no método de compressão por predição

A utilização de códigos de Huffman é semelhante à descrita em 5.1.1.3 para os coeficientes DC.

5.1.3 Método DCT Progressivo

A codificação progressiva de uma imagem é iniciada com a fase de transformada e quantificação tal como no modo sequencial. No entanto, enquanto que no método sequencial a imagem é comprimida num só varrimento, neste método cada componente da imagem é segmentada em várias bandas. O primeiro varrimento permite obter uma versão grosseira da imagem que vai sendo refinada à medida que as várias bandas são decodificadas.

Este processo é mais exigente em termos de quantidade de memória necessária uma vez que requer a utilização de um *buffer* à saída do processo de quantificação. Este *buffer* deve ter capacidade suficiente para armazenar toda a imagem sob a forma de coeficientes da DCT quantificados. Depois da quantificação, os coeficientes são colocados no *buffer* e só depois parcialmente codificados num de múltiplos varrimentos.

A divisão de um bloco em vários segmentos faz-se através de um de dois métodos:

- **Seleção espectral** — os coeficientes ordenados em zig-zag são divididos em bandas de frequência e cada uma codificada separadamente;
- **Aproximações sucessivas** — os coeficientes são inicialmente codificados com pouca precisão seleccionando apenas os N (N especificado pela aplicação) bits mais significativos de cada um. Nas etapas seguintes os bits menos significativos são codificados sucessivamente até à precisão final.

Estes dois procedimentos podem ser usados independentemente ou combinados para formar algoritmos flexíveis. A figura 5.5 ilustra, de forma simbólica, as diferenças entre estes dois métodos de codificação progressiva.

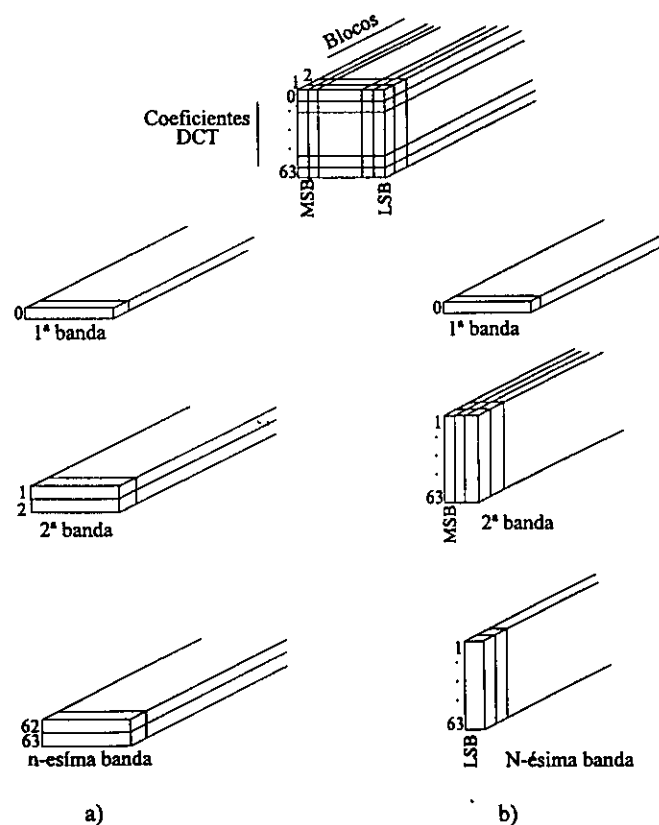


Figura 5.5: Método de compressão progressivo por a) selecção espectral e b) aproximações sucessivas



Figura 5.6: Etapas intermédias de apresentação de uma imagem construída no modo progressivo por selecção espectral com 4 bandas



Figura 5.7: Versão final da imagem construída nos modos progressivo por selecção espectral e sequencial

Na figura 5.6 apresentam-se, da esquerda para a direita e de cima para baixo, os varrimentos resultantes da construção de uma imagem no modo progressivo por selecção espectral. A primeira imagem resulta da descompressão apenas dos coeficientes DC, sendo notório, na segunda fotografia, o início de um novo varrimento. A última etapa do processo (figura 5.7), que corresponde também ao resultado obtido pelo método sequencial, apresenta já poucas diferenças em relação ao varrimento anterior.

5.1.4 Método Hierárquico

O modo hierárquico fornece um modelo de codificação em pirâmide sendo a imagem codificada em múltiplas resoluções. O algoritmo utilizado neste método pode ser descrito pelos seguintes passos:

- sub-amostrar a imagem original por um número múltiplo de 2 em ambas as direcções;
- codificar a imagem resultante do processo de sub-amostragem utilizando um dos métodos permitidos (DCT sequencial, DCT progressivo, predição);
- enquanto não for atingida a resolução original

- descomprimir esta imagem e utilizar um processo de sobre-amostragem de factor 2 horizontal e verticalmente;
- utilizar a imagem resultante do processo de sobre-amostragem como estimativa da original com esta resolução; codificá-la por um dos métodos descritos.

5.1.5 Tratamento de Imagens com Múltiplas Componentes

O tratamento de imagens com várias componentes deve ter em consideração as características do sistema de descompressão. Muitas aplicações exigem um paralelismo dos processos de descompressão e visualização sendo apenas viáveis se as componentes forem multiplexadas durante a fase de compressão. O algoritmo JPEG prevê dois tipos de ordenação das várias componentes de uma imagem:

- as componentes são comprimidas separadamente e transmitidas (armazenadas) por inteiro uma de cada vez;
- a imagem comprimida é ordenada em unidades que contêm um ou mais blocos multiplexados de cada uma das componentes. Neste caso, cada componente C_i é dividida em regiões rectangulares com H_i por V_i blocos de 8×8 amostras. Esses blocos são ordenados formando grupos em que o número total de blocos é determinado pelo número de componentes e pela relação entre os factores H_i e V_i (figura 5.8).

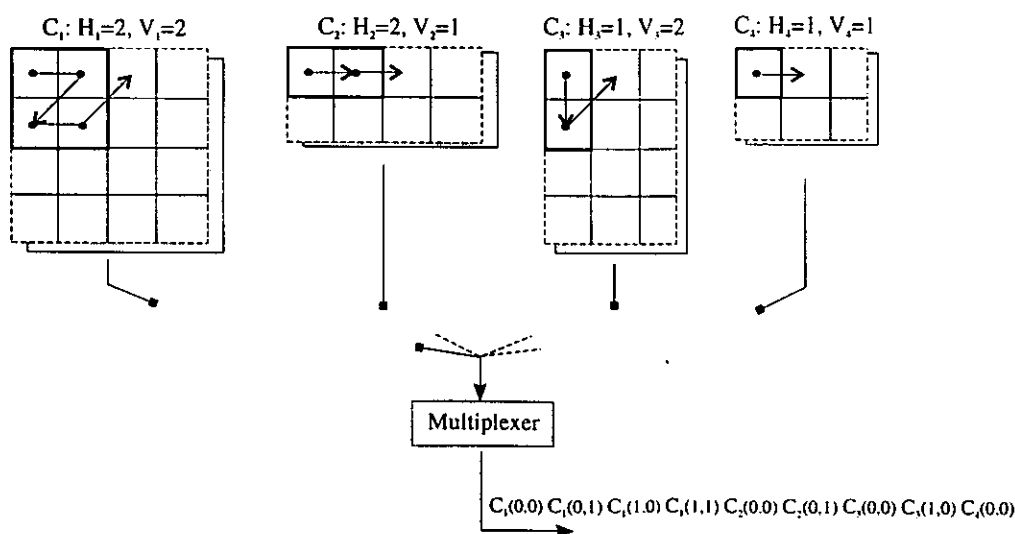


Figura 5.8: Exemplo de multiplexagem e ordenação das componentes de uma imagem

Ao efectuar o entrelaçamento das componentes, o multiplexer controla também a utilização das tabelas correspondentes a cada uma. O JPEG prevê a utilização simultânea de até 4 tabelas de quantificação e 4 tabelas com os códigos de comprimento variável. Todas as amostras de uma componente devem ser comprimidas com a mesma tabela de quantificação e o mesmo conjunto de tabelas de códigos de comprimento variável (geralmente definem-se tabelas diferentes para os coeficientes DC e AC).

5.2 Sintaxe Fotográfica

O modelo utilizado pelos algoritmos JPEG é uma abstracção que permite representar vários tipos de imagens de diferentes aplicações. A informação de controlo acrescentada à imagem comprimida não especifica aspectos importantes para a sua correcta representação, fornecendo apenas os dados necessários à descompressão e reconstrução da imagem. Parâmetros como espaço cromático, características do sistema de aquisição ou colocação da imagem, devem ser fornecidas pela aplicação. A sintaxe fotográfica define um conjunto de atributos necessários ao tratamento e visualização da imagem no terminal, mas não especifica técnicas de pós-processamento.

A estrutura utilizada na definição destes atributos está organizada em camadas seguindo a filosofia OSI. A camada mais alta indica a utilização, ou não, de um conjunto de valores pré-definidos (tabelas de quantificação e de Huffman, dimensões da imagem, etc.) e a imediatamente abaixo define os parâmetros que permitem a colocação de uma imagem de acordo com um sistema de janelas definido em unidades normalizadas. As características dependentes do sistema de aquisição (dimensões da imagem, resolução, etc.), as componentes da imagem, o algoritmo de compressão utilizado e as características do canal de transmissão, são definidas nas camadas seguintes.

A sintaxe fotográfica permite a definição de diferentes sistemas de videotex fotográfico mas, para facilitar a interligação entre eles, especifica um conjunto de classes baseadas na norma CCIR (*Comité Consultatif International des Radiocommunications*) para televisão digital [CCIRa] e no formato CCITT CIF (*Common Interchange Format*). Cada uma das classes determina as facilidades implementadas e é identificada por um código único na mensagem Sbv_Tfi (secção 3.3.1).

Os parâmetros que diferenciam as diversas classes são o método de compressão e a resolução das imagens utilizadas no serviço:

- **Classe 1** — suporta o método de compressão JPEG baseado na DCT e define, como área útil de colocação da imagem, uma janela com 320 pixels por 240 linhas. As imagens

são representadas segundo as componentes YCrCb, sendo a crominância sub-amostrada horizontal e verticalmente, em relação à luminância, como definido no formato CIF (a cada bloco de crominância correspondem 4 blocos de luminância);

- **Classe 2** — define uma janela de dimensão 320 por 480 pixels e a utilização do formato de imagem CCIR 2:1:1. Imagens deste tipo são obtidas por um processo de sub-amostragem na horizontal de imagens 4:2:2. As componentes YCrCb são comprimidas também pelo método sequencial baseado na DCT;
- **Classe 3** — utiliza o formato definido para televisão digital com qualidade de estúdio (CCIR 4:2:2) e uma janela de 640 por 480. As componentes Cr e Cb são também amostradas na horizontal em relação à luminância. Esta classe permite uma melhoria na qualidade das imagens uma vez que tem o dobro da definição na horizontal relativamente à classe anterior. Tal como nas anteriores, as imagens são codificadas no modo sequencial e pelo método de transformadas;
- **Classe 4** — acrescenta à classe 3 o modo de compressão progressivo por selecção espectral para imagens em qualquer um dos formatos referidos;
- **Classe 5** — implementa o modo de compressão progressivo por aproximações sucessivas.

Cada uma destas classes suporta imagens contruídas de acordo com as classes anteriores, isto é, um terminal que se identifique como pertencente à classe 5 consegue descodificar imagens elaboradas segundo qualquer uma das classes 1 a 5.

A definição destas classes, ao limitar o número de variantes permitidas, permite obter algum grau de compatibilidade entre diferentes sistemas. Para aumentar a flexibilidade do serviço de videotex fotográfico, é possível definir uma classe de aplicações privadas que vá ao encontro das necessidades específicas de uma aplicação em particular (resolução, espaço cromático, precisão, etc). Terminais construídos de acordo com esta classe podem, no entanto, não ser compatíveis com aplicações privadas de serviços diferentes.

5.2.1 Adaptação de Resoluções entre as Várias Classes

Ao admitir várias classes, nas quais as imagens têm representações diferentes em termos de resolução, surge o problema de como reproduzir uma imagem elaborada, por exemplo, segundo o formato CIF num terminal Classe 3 (4:2:2). A utilização de unidades normalizadas, para a definição da janela onde será colocada a imagem, permite obter sistemas independentes do *hardware* utilizado na aquisição da imagem.

Todos os parâmetros necessários à correcta representação da imagem são dados em função de uma área base de colocação da informação fotográfica com dimensões 1 e 0.75 na horizontal e vertical, respectivamente. Esta área corresponde, no caso de um terminal Classe 3, a 640 por 480 pixels, mas a apenas 320 por 240 pixels num terminal Classe 1. A cada imagem está associada uma janela, com dimensões *sizw* e *sizh*, colocada na área base mencionada de acordo com *loch* e *locv* (figura 5.9). Qualquer um destes parâmetros é codificado segundo coordenadas normalizadas e permite obter representações independentes da resolução do sistema. Isto viabiliza a construção de páginas mistas mantendo no terminal, independentemente da classe a que pertence, a posição relativa das várias imagens e gráficos. Um terminal com resolução superior à de uma imagem implementa funções de mapeamento de cada uma das amostras num ou vários pontos da área útil de dados. Um terminal 4:2:2 duplica, na horizontal, o número de pixels de imagens 2:1:1 e CIF de maneira a converter os 320 pixels correspondentes, em coordenadas normalizadas, a 0.1, nos 640 que representam o valor 0.1 no formato 4:2:2. Este processo de duplicação é também repetido, para imagens CIF, na direcção vertical (CIF $\rightarrow$ 0.75=240; 4:2:2 $\rightarrow$ 0.75=480).

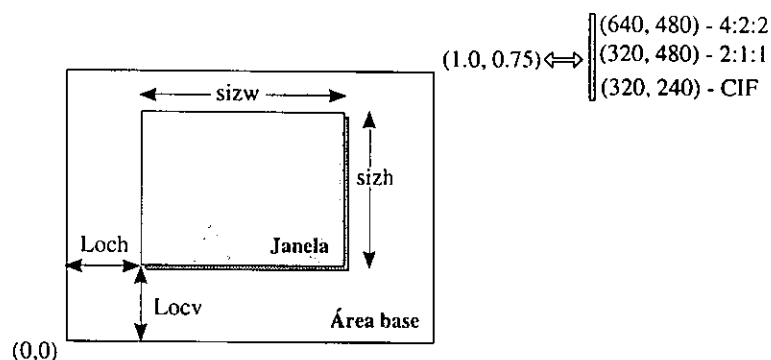


Figura 5.9: Definição de uma área segundo coordenadas normalizadas

5.3 Técnicas de Implementação

Sendo a mais exigentes das aplicações em termos de cálculo, a implementação da aplicação fotográfica requer um conjunto de optimizações que permitam dotar o serviço de uma qualidade aceitável. Nos pontos seguintes descrevem-se algumas técnicas utilizadas na implementação que permitiram uma melhoria do serviço.

IDCT

De todo o processo de descompressão, a transformada é a etapa que requer o maior dispêndio de tempo, sendo por isso necessário recorrer a métodos rápidos para o seu cálculo [RY89]. A utilização da DCT em inúmeras aplicações levou ao aparecimento de diversos algoritmos que a implementam [RY89],[CSC77],[Lee84],[Haq85],[Hou87], [CH91],[CC91],[CL91]. É possível encontrar nestas referências comparações em termos de número de multiplicações e somas necessárias para cada um deles. Outro aspecto a não desprezar está relacionado com a facilidade de implementação utilizando técnicas de cálculo em vírgula fixa que são essenciais quando o terminal não está equipado com um processador aritmético. A figura 5.10 apresenta o fluxograma da IDCT implementada no terminal [RY89]. A transformada total de um bloco é obtida pela aplicação sucessiva desta rotina a cada uma das linhas e colunas do bloco.

Em blocos com pouca informação, é normal encontrar linhas ou colunas em que os coeficientes são todos nulos. Optou-se por realizar, antes de iniciar o cálculo da transformada, uma análise por linha e coluna que detecte estes casos, evitando assim operações desnecessárias com um aumento de tempo, imposto por esta pesquisa, desprezável.

Imagens codificadas no modo progressivo sugerem também uma optimização do algoritmo da IDCT uma vez que nas bandas iniciais grande parte dos coeficientes são desprezados, formando por isso blocos em que o número de zeros é elevado. É importante ter em consideração esta característica já que o objectivo deste método é obter rapidamente, nos primeiros varrimentos, uma versão preliminar da imagem. A alteração da IDCT da figura 5.10, para versões dependentes do número de coeficientes da banda, permite aumentar a rapidez de acesso às primeiras versões da imagem. Esta alteração consiste na redefinição do fluxograma eliminando os ramos correspondentes aos coeficientes que não são incluídos nessa banda, diminuindo assim o número de multiplicações e somas. A figura 5.11 apresenta o fluxograma correspondente a uma IDCT em que os coeficientes 5, 6 e 7 não estão incluídos na banda a descomprimir. Durante o processo de descompressão, a selecção da IDCT apropriada pode ser feita muito facilmente, sem ser necessário analisar cada um dos coeficientes para verificar se é nulo, através da indicação do número de elementos pertencentes a essa banda.

Descompressão parcial de uma imagem

A sintaxe fotográfica permite definir, além da localização da janela onde será colocada a imagem, o posicionamento da imagem em relação a essa janela. Dependendo dos parâmetros *refh*, *offh*, *refv* e *offv* (figura 5.12 a)), parte da imagem poderá cair fora da janela definida sendo por isso invisível. A descompressão dessa porção de imagem implica um dispêndio de tempo que não se traduz numa acção de interesse para o utilizador. Para aumentar a eficiência da aplicação fotográfica, faz-se o processamento inicial dos parâmetros que determinam a parte visível da imagem, determinando as linhas e colunas de blocos que contêm pontos

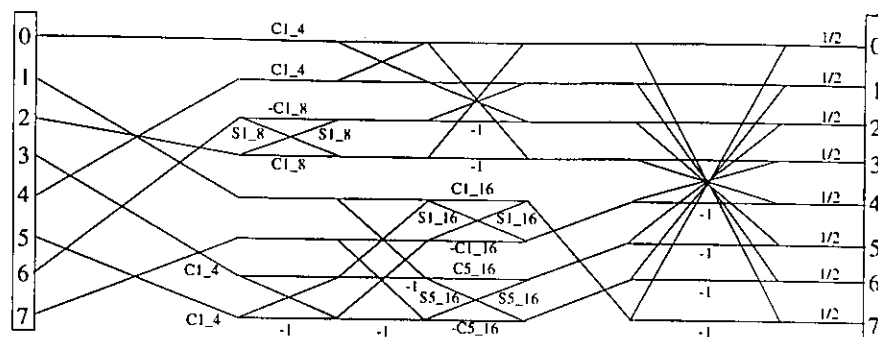


Figura 5.10: Fluxograma da transformada rápida de cossenos

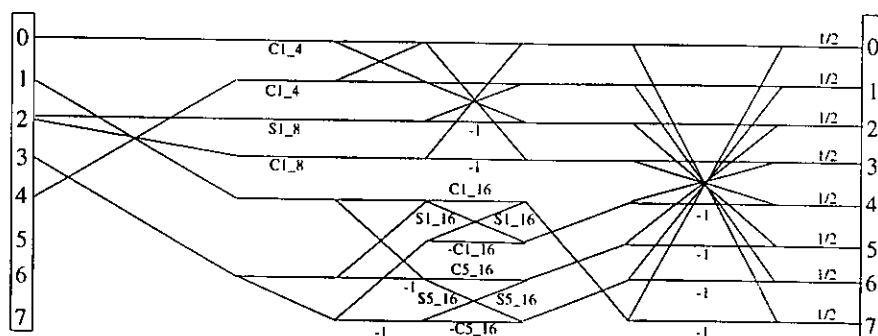


Figura 5.11: Simplificação do fluxograma da transformada rápida de cossenos quando os coeficientes 5, 6 e 7 são nulos

visíveis. O processo de descompressão de um bloco só será completo se este contiver amostras dentro da janela. Caso contrário, e após efectuar a descodificação dos códigos de comprimento variável, desprezam-se os valores obtidos e passa-se directamente à descodificação do próximo bloco sem passar pelas fases de desquantificação, IDCT e visualização e evitando assim as etapas de processamento mais pesadas. Este processo pode ser descrito de forma sistemática pelo seguinte algoritmo:

- Determinar (i, j) e (k, l) , que representam os índices do primeiro e último pixels visíveis da imagem, respectivamente;
- determinar o bloco $b(m, n)$ tal que $(i, j) \in b(m, n)$;
- determinar o bloco $b(o, p)$ tal que $(k, l) \in b(o, p)$;
- inicializar $b(r, s) = b(0, 0)$
- enquanto houver blocos fazer:
 - descodificar códigos de comprimento variável;

- se $r \in [m, o] \wedge s \in [n, p]$ fazer desquantificação, IDCT e apresentação do bloco;
- incrementar (r,s) .

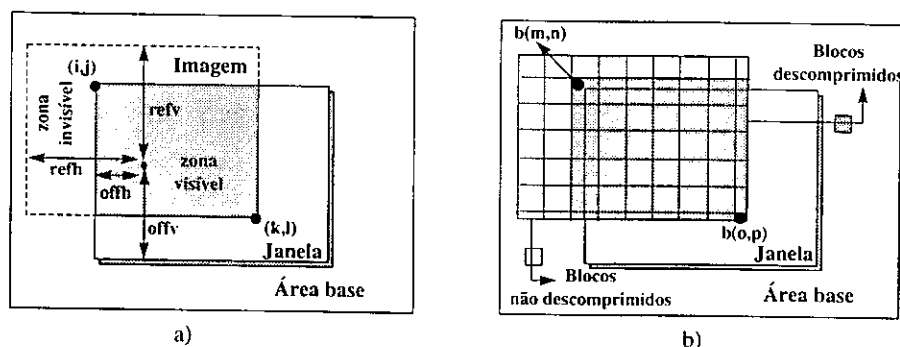


Figura 5.12: Sintaxe fotográfica: a) exemplo de colocação parcial de uma imagem na janela definida pela sintaxe e b) selecção dos blocos a descomprimir

Multiplexagem de componentes

O tratamento e apresentação de imagens policromáticas vai depender da ordenação relativa das componentes. O facto de o algoritmo JPEG prever a não multiplexagem de componentes, levanta problemas de complexidade do terminal. Do ponto de vista de eficiência, a melhor solução será a criação, no servidor, de páginas fotográficas em que as componentes Y, Cr e Cb são multiplexadas, permitindo a escrita imediata de um bloco na placa gráfica após descompressão. No entanto, prevendo a utilização, no serviço de videotex fotográfico, de imagens geradas nas mais diversas aplicações, o suporte de componentes não multiplexadas foi também implementado. A detecção de uma imagem nestas condições origina uma alteração da fase final de processamento, sendo cada um dos blocos colocado na memória extendida do terminal, em vez de escrito directamente na placa gráfica. Logo no início da recepção da última componente, os blocos armazenados temporariamente são conjugados com a componente em descompressão e inicia-se o processo de apresentação. A organização das componentes neste formato não é a mais adequada ao serviço uma vez que o utilizador terá que aguardar a descompressão completa das duas primeiras componentes antes de ter acesso à imagem.

Uma implementação alternativa foi também estudada. A ideia básica consistia na apresentação inicial de uma componente como uma imagem monocromática que seria progressivamente colorida quando se iniciasse o processo de descompressão das outras componentes. Este processo permitia obter mais rapidamente uma primeira versão monocromática da ima-

gem, mas iria aumentar o tempo total necessário à apresentação da versão final pois implicava vários acessos à placa gráfica.

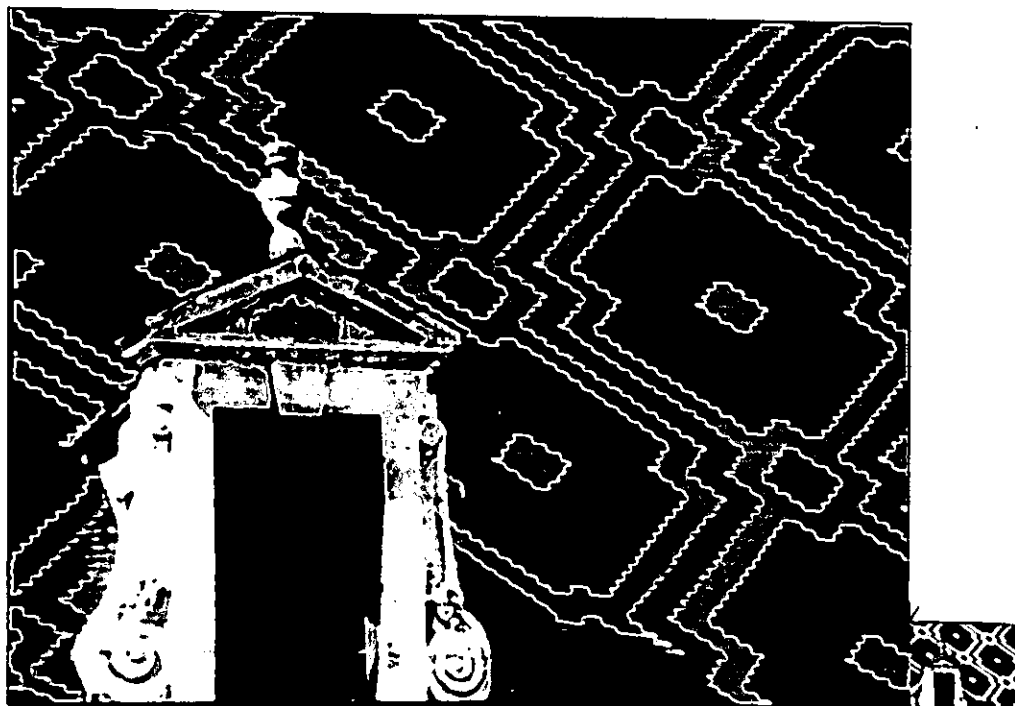


Figura 5.13: Tamanhos relativos de uma imagem e da imagem DC correspondente

Imagens DC

Uma pesquisa eficiente na base de dados de imagens permite melhorar a qualidade do serviço diminuindo o tempo de acesso e de selecção da informação fotográfica. A criação de aplicações em que ao utilizador é fornecida uma primeira apresentação das imagens disponíveis pode ser feita recorrendo a *imagens DC*. Estas imagens são versões reduzidas das originais (1/64) (figura 5.13) mas que permitem um grau de realismo elevado com tempos de descompressão reduzidos e sem aumentar o número de imagens a armazenar na base de dados. As figura 5.14 e 5.15 representam a mesma imagem descomprimida no formato original de 640x480 pixels e no formato DC. A imagem DC é obtida da original sem ser necessário efectuar a IDCT, reduzindo, por isso, o tempo de descompressão. O servidor transmite a imagem comprimida com uma indicação de que a apresentação deve ser feita no formato DC e não no tamanho real. A aplicação fotográfica, no terminal, efectua a descodificação dos códigos de comprimento variável e pela expressão

$$x = \frac{DC + Quant[0][0]}{8} + 128 \quad (5.3)$$

obtem, para cada um dos blocos, um valor que corresponde à média das 64 amostras do bloco.

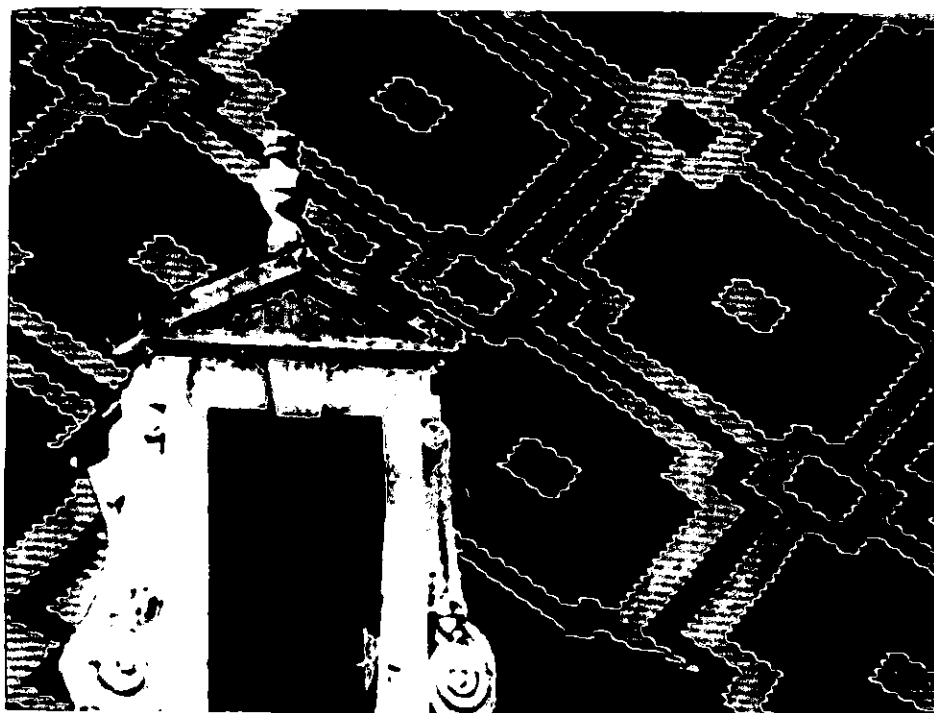


Figura 5.14: Imagem descomprimida para a resolução original de 640x480 pixels



Figura 5.15: Imagem DC correspondente

A figura 5.16 apresenta um conjunto de imagens DC utilizadas pela aplicação para criar um menu auxiliar que permita ao utilizador seleccionar as imagens que pretende ver na versão integral.



Figura 5.16: Exemplo da criação de um menu auxiliar utilizando imagens DC

Seleccção da taxa de compressão

A taxa de compressão de uma imagem é especificada pelas tabelas de quantificação utilizadas durante a compressão. Em [JPEG] e [ETSh] são fornecidas tabelas ($Q[i][j]$), que definem os passos de quantificação para cada um dos coeficientes do bloco, calculadas com base em testes subjectivos de qualidade. A alteração da taxa de compressão de uma imagem obriga à definição de novas tabelas que permitam um ajuste dessa taxa de acordo com a qualidade desejada.

Para a fase de criação de uma nova página fotográfica no servidor, foi criado um mecanismo que permite especificar, entre outras coisas, um factor de qualidade F_Q que define a tabela de quantificação de acordo com uma das seguintes expressões:

$$Quant[i][j] = 1 + (1 + i + j) \times F_Q; \quad (5.4)$$

$$Quant[i][j] = Q[i][j] \times F_Q. \quad (5.5)$$

A equação (5.4) define uma tabela em que o passo de quantificação é função da posição do coeficiente no bloco, de tal maneira que elementos pertencentes à mesma diagonal utilizem o

mesmo quantificador. O segundo método (equação 5.5) calcula uma tabela de quantificação baseada na tabela Q pré-definida. O factor $F \cdot Q_2$ indica a relação percentual entre a tabela a utilizar e a tabela Q . Experiências realizadas demonstraram que é possível definir valores para $F \cdot Q_1$ e $F \cdot Q_2$ que conduzam a taxas de compressão semelhantes para os mesmos resultados em termos de qualidade subjectiva. Na versão final do compressor, optou-se pelo segundo método para permitir a utilização da tabela pré-definida quando o factor de qualidade é 1.

O *software* de compressão desenvolvido prevê também a opção que permite definir qual o formato da imagem comprimida, isto é, a criação de imagens 4:2:2, 2:1:1 e CIF. Uma vez que o terminal suporta qualquer uma das resoluções mencionadas, é possível entrar com este factor para obter taxas de compressão superiores. Uma imagem com dimensões 1×0.75 , em coordenadas normalizadas, corresponde a 9600 ($4800 Y + 2400 Cr + 2400 Cb$), 4800 ($2400 Y + 1200 Cr + 1200 Cb$) e 1800 ($1200 Y + 300 Cr + 300 Cb$) blocos nos formatos 4:2:2, 2:1:1 e CIF, respectivamente. A perda de qualidade introduzida pelo processo de sub-amostragem pode não ser significativa em muitas aplicações e permite, além do aumento do factor de compressão, uma diminuição do tempo de transmissão e descompressão.

A optimização da base de dados e também do terminal, se considerarmos tempos de processamento, pode ser conseguida pela escolha correcta do factor de compressão e do formato utilizado. A figura 5.17 apresenta uma imagem 2:1:1 comprimida a uma razão de 1 para 10 e decodificada num terminal 4:2:2. Os tempo de transmissão e descompressão são aproximadamente metade dos necessários, para uma imagem equivalente no formato 4:2:2, mantendo, no entanto, uma qualidade aceitável.

5.4 Especificação de uma Aplicação Privada

O serviço de videotex é definido de forma independente das características do equipamento terminal. A implementação de um terminal com limitações, em termos de memória e capacidades gráficas, exige uma gestão local dos meios disponíveis e da informação, que é da inteira responsabilidade do equipamento terminal e transparente ao serviço. Com o objectivo de servir um conjunto vasto de aplicações, algumas das quais exigentes em termos de qualidade, são utilizados 24 bits para especificar a cor de cada um dos pixels da imagem. No entanto, muitos equipamentos têm capacidades gráficas reduzidas em termos de número de cores que podem apresentar simultaneamente exigindo, na recepção da imagem, a execução de um processo, normalmente designado de *Quantificação de Cor*, que se destina a reduzir o número de cores original a um conjunto limitado — *palette*. Este processo, a ser implementado no terminal, provocaria um perda de qualidade acentuada, em termos de tempo de processamento, já que é mais exigente que todo o algoritmo de descompressão e apresentação. A definição de uma aplicação privada permite a passagem deste processo para o servidor eliminando assim



Figura 5.17: Imagem comprimida no formato 2:1:1 e apresentada num terminal 4:2:2

o problema do tempo de cálculo pois a criação de páginas fotográficas não é um processo a realizar em tempo real. A ideia base para esta especificação de uma nova classe, consiste na re-utilização do *software* de compressão e descompressão desenvolvido e integrado na base de dados e terminal, e na inclusão de um módulo de pré-processamento apenas na base de dados. O recurso a técnicas de programação destinadas a otimizar o programa em termos de tempo de execução (tabelas, *macros*, etc.) torna crítico, em termos de memória, o crescimento da quantidade de *software* no terminal.

Os objectivos desta secção são a especificação de uma aplicação com as seguintes características: apresentação das imagens utilizando apenas 256 cores (número utilizado em grande parte dos modos gráficos em PCs); reutilização do *software* já desenvolvido para a aplicação fotográfica; acréscimos introduzidos nos tempo de processamento transferidos para a fase de criação das páginas.

O algoritmo proposto pode ser descrito, de uma forma simplificada, pelos seguintes passos:

- utilizar um método de quantificação de cor que permita reduzir as 2^{24} cores possíveis para um conjunto de 256 cores que constituem uma *palette*;
- aplicar o método de compressão JPEG;
- acrescentar a sintaxe fotográfica com indicação de *Aplicação Privada* e a definição do

mapa de cores utilizado.

O terminal, ao receber uma imagem classificada como pertencente à classe privada, executa as seguintes operações:

- carrega a *palette* na placa gráfica;
- efectua a descodificação da sintaxe fotográfica e descomprime a imagem pelo método JPEG.

5.4.1 Quantificação de Cor

A designação de quantificação de cor é atribuída ao processo de selecção de um conjunto de cores (*palette*), que permitam representar toda a gama de cores de uma imagem, e ao mapeamento de cada uma das cores originais numa das seleccionadas. Este processo não é mais que um caso particular do método de quantificação vectorial apresentado na secção 4.5, envolvendo, normalmente, um espaço tridimensional definido pelas três cores primárias. A quantificação multidimensional é um processo complexo para o qual não existem métodos rápidos que permitam determinar o óptimo, recorrendo-se normalmente a heurísticas que apenas permitem obter uma solução sub-óptima. A degradação imposta por este processamento pode ser superior à resultante das perdas de compressão pois gera contornos falsos muito visíveis devido à quantidade limitada de cores disponível. Tal como o algoritmo base de VQ, a quantificação de cor pode ser dividida em duas fases:

1. criação da *palette* com o melhor conjunto de cores possível, para uma determinada imagem;
2. associação de cada um dos pixels da imagem a uma das cores da *palette* de maneira a obter a melhor qualidade possível.

5.4.1.1 Criação da *palette*

A fase de criação do mapa de cores é geralmente antecedida por uma recolha de estatísticas de cor da imagem que permite a construção de um histograma. Por questões práticas de implementação, uma quantificação de 24 para 15 bits, desprezando os 3 bits menos significativos de cada componente, pode ser efectuada sem que isso implique uma degradação visível da qualidade final da imagem. A tabela de frequências terá neste caso apenas de 32768 posições ao contrário das 16 777 216 resultantes da utilização da precisão total.

Para a criação da *palette* propriamente dita, existe alguma literatura [Hec82], [OB91], [Wu92], embora escassa, que sugere os seguintes métodos:

- **Distribuição uniforme**

é efectuada uma divisão da gama de valores possíveis de uma imagem, num conjunto de intervalos com a mesma dimensão. Uma distribuição possível é a atribuição de 8 níveis às componentes R e G e 4 níveis à componente B. A *palette* é o conjunto de valores RGB com todas as combinações possíveis destes níveis. Este processo é o mais simples mas também o que conduz a piores resultados uma vez que a tabela de cores é calculada sem recurso às características da imagem. O efeito de contornos falsos pode ser bastante grave pois é possível que nem todas as cores da *palette* sejam utilizadas, havendo um desperdício de algumas posições da tabela que poderiam conter outras combinações mais apropriadas. Este método é o único que não recorre à recolha de estatísticas da imagem que permite assegurar que todas as cores disponíveis são utilizadas;

- **Cores mais frequentes (*Popularity Algorithm*)**

a ideia base deste algoritmo é a utilização das k , em que k é a dimensão da *palette*, cores mais frequentes do histograma. A qualidade das imagens quantificadas por este processo depende da quantidade de cores existentes na imagem, sendo os resultados piores em casos em que a gama dinâmica é muita elevada. Isto deve-se, como é evidente, ao facto de algumas regiões do espaço cromático não serem consideradas devido ao fraco número de ocorrências e a uma multiplicidade de cores muito próximas com elevada frequência. Uma heurística, que pode aumentar o desempenho do algoritmo, consiste na selecção de uma cor do histograma, seguida de uma redução de amplitude na escala de ocorrências dos pontos que se encontram na vizinhança. Isto evita que as cores disponíveis pertençam a uma área restrita onde acontecem a maior parte das ocorrências;

- **Median Cut**

o conceito básico deste método é a utilização de cada uma das cores da *palette* para representar o mesmo número de pixels da imagem. O algoritmo divide, recursivamente, o espaço tri-dimensional formado pelas três componentes, em cubos de dimensões sucessivamente inferiores de forma a que cada um contenha aproximadamente o mesmo número de pixels da imagem. Uma vez gerados os k cubos, é calculada a média dos valores contidos em cada um para obter os k representantes que formam a tabela.

Os métodos apresentados podem ser considerados como etapas que permitem obter uma versão inicial da *palette*, que pode ser depois melhorada pela aplicação de processos iterativos que permitem refinar a solução. A primeira versão é calculada de maneira a atribuir um maior número de níveis de quantificação a regiões com mais amostras associadas, mas em

nenhum dos métodos é feita uma tentativa de minimizar qualquer medida de erro objectiva. O algoritmo LBG, referido na secção 4.5, pode ser utilizado para otimizar a solução em relação a uma medida de distorção definida. No entanto, este processo além de destruir a estrutura em árvore criada na obtenção da *palette* inicial, não garante a obtenção do óptimo global, sendo de esperar que se atinja um mínimo local dependente dos dados iniciais do problema.

5.4.1.2 Mapeamento de cada pixel num valor da *palette*

Uma vez seleccionada a *palette*, inicia-se a segunda fase do processo de quantificação de cor que corresponde à determinação do vizinho mais próximo segundo uma medida de distorção, como foi referido em 4.5. Este método, ao efectuar uma pesquisa exaustiva, torna-se demasiado lento pois analisa distâncias a pontos que não seriam nunca candidatos ao lugar de vizinho mais próximo. A organização da *palette*, de maneira a permitir uma pesquisa localizada numa zona restrita, é a solução para diminuir o número de cálculos e comparações e assim reduzir o tempo necessário a esta fase. Várias aproximações a este problema podem ser encontradas em [Hec82],[Wu92] e [GG92] e baseiam-se numa reordenação da tabela de cores, de maneira a permitir uma pesquisa mais eficiente, ou na imposição de restrições na fase 1 que permitem obter, com base numa heurística, tabelas estruturadas que, não sendo óptimas, podem conduzir a bons resultados subjectivos.

Uma heurística utilizada, que embora muito simples conduziu a resultados próximos dos obtidos pelo processo de pesquisa exaustiva, baseou-se na ordenação da *palette* por ordem crescente da componente R, depois G e finalmente B. O algoritmo para selecção do índice i da tabela que corresponde ao mapeamento do vector $\mathbf{x}$ no vector $\hat{\mathbf{x}}$ efectua uma pesquisa localizada sobre a tabela, ordenada segundo o critério referido, utilizando a seguinte heurística:

- considerando apenas a componente R, procura o elemento i da tabela que corresponde ao vizinho mais próximo ($\mathcal{D}_{R_i} = \min \mathcal{D}(x_R, \hat{x}_{R_i})$). Determina a distorção total, $\mathcal{D}_i = \mathcal{D}(\mathbf{x}, \hat{\mathbf{x}}_i)$, e faz $d_{\min} = \mathcal{D}_i$;
- partindo do elemento i , pesquisa, nos sentidos ascendentes e descendentes da tabela, novo valor mínimo de distorção.
- termina quando $\mathcal{D}_{R_j} > d_{\min}$.

A minimização de uma medida objectiva como o erro quadrático não garante necessariamente os melhores resultados subjectivos porque o erro resultante da quantificação de um pixel não pode ser tratado de forma independente da quantificação das amostras vizinhas. Dependendo da imagem, os erros de quantificação podem ser evidentes ou invisíveis. Zonas de transições

rápidas são menos afectadas pelos efeitos da quantificação, quando comparadas com zonas suaves da imagem onde os contornos falsos se tornam visualmente desagradáveis. Este efeito traduz-se pela substituição de zonas na imagem original em que a cor varia de forma suave, por pequenas zonas de cor constante com mudanças abruptas de cor nas fronteiras entre si. Técnicas que permitem reduzir este efeito são normalmente designadas por *dithering* e tentam, em lugar de minimizar o erro de cada um dos pixels individualmente, aproximar a cor média local da imagem quantificada à cor média da imagem original, na mesma zona.

- **Dithering aleatório**

Este método consiste na modulação da imagem original por um sinal de frequência elevada (ruído aleatório) antes da quantificação. De todas as técnicas apresentadas, é a mais simples mas também a que conduz a piores resultados.

- **Dithering por padrão (*Ordered Dither*)**

A um conjunto x , de pontos adjacentes da imagem, é somado um ruído pseudo-aleatório com um determinado padrão P , em que P é uma matriz de $n \times n$. Cada amostra, antes da quantificação, sofre uma transformação de acordo com a seguinte expressão:

$$\tilde{x}_{i,j} = p_{i \bmod n, j \bmod n} + x_{i,j}. \quad (5.6)$$

A amplitude do padrão de ruído é escolhida de forma que uma área de cor aproximadamente constante seja quantificada num conjunto de cores próximas e com isso sejam eliminadas zonas de contornos falsos.

Este método apresenta alguns problemas quando aplicado em conjunto com *palettes* em que as cores não estão uniformemente distribuídas. A distância entre cores próximas varia significativamente ao longo do espaço cromático no caso de tabelas calculadas especificamente para uma imagem, dificultando a determinação das amplitudes do padrão de *dithering* que permitam quantificar as zonas de cor constante sem simultaneamente introduzir ruído visível noutras regiões. Um processo para ultrapassar este problema é o ajuste do padrão à distância entre duas cores próximas da *palette*. Para isso, escolhem-se dois candidatos para quantificar um valor e o ruído introduzido é função da distância entre eles:

$$\tilde{x}_{i,j} = (q_2 - q_1)p_{i \bmod n, j \bmod n} + x_{i,j}, \quad (5.7)$$

onde $p_{i,j}$ é um escalar no intervalo $[-1/2, 1/2]$.

- **Dithering por propagação do erro (*Error Diffusion*)**

O objectivo do método de propagação do erro é a obtenção de uma imagem quantificada com valor médio igual ao da imagem original. Para cada ponto da imagem, determina-se, em primeiro lugar, a cor mais próxima contida na *palette* e calcula-se o erro de

quantificação $\epsilon = x - \hat{x}$. O erro resultante da amostra actual é distribuído pelas amostras vizinhas de acordo com a matriz **P** que define a vizinhança e os pesos associados a cada uma das amostras que dela fazem parte. Existem várias maneiras de distribuir o erro sendo uma das mais utilizadas a apresentada na tabela 5.2 com a designação de filtro F-S por ter sido introduzido por Floyd e Steinberg. Para este caso, e considerando a amostra $x_{i,j}$, o erro $\epsilon_{i,j}$ é difundida às amostras vizinhas do seguinte modo:

$$\tilde{x}_{i,j+1} = x_{i,j+1} + \frac{7}{16}\epsilon_{i,j},$$

$$\tilde{x}_{i+1,j-1} = x_{i+1,j-1} + \frac{3}{16}\epsilon_{i,j},$$

$$\tilde{x}_{i+1,j} = x_{i+1,j} + \frac{5}{16}\epsilon_{i,j},$$

$$\tilde{x}_{i+1,j+1} = x_{i+1,j+1} + \frac{1}{16}\epsilon_{i,j}.$$

A técnica de propagação de erros permite obter resultados superiores à anterior pois não impõe um padrão regular, que se pode tornar visível em algumas imagens. Na tabela 5.2 apresentam-se outros filtros que podem ser utilizados para substituir o filtro F-S na propagação do erro. O padrão utilizado em qualquer um destes filtros permite alguma eficiência na implementação pois as divisões necessárias podem ser substituídas por simples rotações uma vez que a soma dos pesos é potência de 2.

Tabela 5.2: Filtros associados à propagação do erro às amostras vizinhas para a operação de *dithering*

Filtro F-S		Filtro a		Filtro b				Filtro c			
x	7	x	3	x 8 4				x 5 3			
3	5 1	3	2	2	4	8	4 2	2	4	5	4 2
									2	3	2

5.4.2 Ordenação da Tabela de Cores

Os algoritmos de compressão, nomeadamente os baseados na transformada de cosenos, exploram a correlação existente entre pixels adjacentes de uma imagem, para obter compressões eficientes. A representação de uma imagem como um conjunto de índices de uma tabela (*palette*) destrói essa estrutura espacial uma vez que, normalmente, o índice não está directamente relacionado com a cor do pixel — dois elementos adjacentes da tabela podem

corresponder a pontos muito distantes no espaço cromático tridimensional. Além de dificultar a compressão, esta característica das *palettes* levanta problemas quando se pretendem utilizar métodos que introduzem perdas como é o caso do algoritmo JPEG baseado na DCT. A simples alteração de 1 ou 2 dos bits menos significativos de uma imagem representada por meio de índices provoca erros que conduzem a uma imagem de fraca qualidade subjectiva ou mesmo irreconhecível. Um vez que a fase de quantificação dos coeficientes da transformada implica a introdução de erros semelhantes aos referidos, a reutilização do *software* de compressão desenvolvido é, à partida, impossível para imagens que sofreram um processo de quantificação de cor. A recuperação da correlação pode, no entanto, ser obtida por uma re-ordenação da *palette* de forma a obter uma relação directa entre os índices e a cor que representam.

A ordenação da *palette* é um problema de optimização combinatória semelhante ao *problema do caixeiro viajante* em que, dado um conjunto de k vectores tridimensionais, $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k)$, e uma medida de distância, $d(i, j)$, entre dois vectores $\mathbf{x}_i$ e $\mathbf{x}_j$, se pretende encontrar uma função de ordenação $\mathcal{O}_i$ que minimiza a distância total D :

$$D = \sum_{i=1}^k d(\mathcal{O}_i, \mathcal{O}_{i+1}). \quad (5.8)$$

A introdução do termo $d(\mathcal{O}_k, \mathcal{O}_1)$ na equação (5.8), transforma a ordenação da *palette* num verdadeiro problema do caixeiro viajante, o qual pertence à classe dos problemas NP-completos [GJ79] sendo $1/2[(k-1)!]$ o número de ordenações a considerar para a sua resolução. A obtenção da solução óptima é possível apenas para valores de k na ordem das 20 cidades sendo necessário, para valores superiores, recorrer a heurísticas que, não garantindo o óptimo, permitem resultados satisfatórios.

Existem duas estratégias básicas na definição de uma heurística — divisão e conquista (*divide and conquer*) e processos iterativos. Na primeira, o problema é dividido em subproblemas de dimensão tratável e cada um deles resolvido individualmente. Nos processos iterativos parte-se de uma solução inicial que é refinada de cada vez que se obtém uma ordenação dos elementos que conduz a uma função custo menor.

O *simulated annealing* [KGV83] é uma técnica estocástica iterativa, que tem como base a termodinâmica, e que pode ser utilizada para a resolução de problemas combinatórios. O algoritmo que implementa este processo pode ser descrito pelos seguintes passos:

- seleccionar uma temperatura inicial T e um factor de arrefecimento α ;
- escolher um novo caminho e calcular a alteração ΔE na função custo. Se $\Delta E \leq 0$ então aceitar o novo caminho;

- se $\Delta E > 0$ aceitar o novo caminho com uma probabilidade $P(\Delta E) = \exp(-(\Delta E/K_B T))$, em que K_B é a constante de Boltzmann. Para isso, gerar um número aleatório, n , utilizando uma distribuição uniforme no intervalo $[0,1)$ e aceitar o novo caminho se $n < P(\Delta E)$;
- continuar a perturbar o sistema, à mesma temperatura, durante I iterações.
- arrefecer o sistema utilizando o factor de arrefecimento α para determinar a nova temperatura de funcionamento $T_j = \alpha T_{j-1}$ e continuar o processo iterativo;
- terminar quando nenhuma alteração de caminho é aceite a uma dada temperatura.

A adaptação deste algoritmo ao problema da ordenação da *palette* corresponde a considerar cada elemento como sendo uma cidade a visitar e a escolha de um novo caminho uma re-organização dos elementos na tabela de cores.

A perturbação do sistema, ou escolha de um novo caminho, exige a definição de um método que não é incluído no algoritmo de *simulated annealing*. Uma possibilidade, é a troca de r segmentos do percurso actual, por r segmentos que não façam ainda parte da solução. [HS93] sugere um processo alternativo em que, em cada iteração, se escolhe aleatoriamente uma de duas possíveis alterações ao caminho: a primeira corresponde a retirar uma cidade da posição actual e colocá-la noutro ponto escolhido aleatoriamente; o segundo método troca a posição relativa entre dois pontos adjacentes do percurso, pontos esses também escolhidos aleatoriamente.

O processo de tomada de decisão para aceitação ou não do novo percurso (algoritmo de Metropolis), ao admitir pequenos aumentos da função custo, evita que a função caia num mínimo local do qual não é possível sair. Este algoritmo é por isso menos sensível ao caminho inicial escolhido que os métodos iterativos simples.

Outra heurística, que adopta uma técnica semelhante de fuga aos mínimos locais ao aceitar uma nova solução com função custo superior à actual, é a *Pesquisa Tabu (Tabu Search)* [WH89]. Este método exige a definição de uma vizinhança da solução actual e aceita um novo percurso p , pertencente a essa vizinhança, ainda que este facto introduza um custo suplementar e desde que essa solução não pertença à lista das soluções visitadas (lista tabu).

Uma opção bastante diferente das referidas consiste na utilização de métodos de divisão e conquista [Hua91]. Partindo de uma *palette* inicial, agrupam-se os seus k elementos em vários conjuntos de cores semelhantes e cada um é sujeito a uma ordenação. O processo continua com a ordenação dos grupos entre si e, para evitar descontinuidades muito notórias entre as fronteiras de dois grupos adjacentes, criam-se novos elementos que preenchem o espaço entre eles resultando no fim uma *palette* de $k + m$ elementos.

Qualquer um destes métodos, sendo heurísticas gerais, exigem a definição de parâmetros e processos de escolha de novos percursos que são específicos de cada uma das aplicações. Não existe, à partida, uma solução ideal para estes valores sendo necessário efectuar simulações para determinar o melhor resultado.

5.4.3 Algoritmo de Compressão

A ordenação da tabela de cores permite restituir à imagem alguma correlação entre pixels adjacentes uma vez que índices próximos passam a representar cores próximas segundo o critério definido. A introdução de erros, por processos de quantificação, passa então novamente a ser possível sem que isso conduza à destruição da imagem. Depois da fase de ordenação da *palette* aplica-se o algoritmos de compressão desenvolvido para as outras classes de serviço, considerando a imagem como sendo monocromática. No entanto, sendo as características da componente a comprimir bastante diferentes da luminância, é de prever que a tabela de quantificação a utilizar tenha que ser optimizada para cada uma das imagens.

5.5 Análise Subjectiva de Qualidade

Sistemas envolvendo técnicas de processamento de imagem exigem a definição de medidas de distorção ou de qualidade que permitam a escolha de parâmetros indispensáveis à implementação do algoritmo utilizado. Por questões de facilidade, medidas de distorção como o MSE são geralmente utilizadas na definição e escolha das variáveis em jogo apesar de não traduzirem as alterações subjectivas impostas pela codificação. A dificuldade em encontrar um critério objectivo que permita classificar as imagens implica o recurso a testes subjectivos de qualidade de serviço (QoS — *Quality of Service*). Esta análise subjectiva é geralmente baseada em testes envolvendo um conjunto de observadores e uma escala de apreciação com a qual cada observador envolvido classifica cada uma das imagens. Resultados publicados baseiam-se normalmente na recomendação CCIR 500 [CCIRb] que define métodos subjectivos para avaliação de qualidade da televisão. Aspectos como número e características dos observadores, escala de classificação, condições ambientais do local de teste, distância ao monitor, sequência das imagens, etc., são sugeridos.

A utilização de um método semelhante para o serviço de videotex alfafotográfico exige algumas alterações que permitam adequar estes testes às características do serviço. A qualidade de serviço para um utilizador usual prende-se com aspectos não só de qualidade das imagens mas também com atrasos de transmissão e descompressão. Como foi referido em 5.3 imagens com a resolução CCIR 2:1:1 impõem tempos de transmissão e descompressão bastante inferiores aos resultantes da utilização de imagens CCIR 4:2:2. Algumas experiências com utilizadores não

especializados em técnicas de processamento de imagem, e por isso menos atentos a pequenos defeitos impostos pelos processos de compressão, permitem prever que, para grande parte das aplicações do serviço de videotex fotográfico, a utilização de um formato com resolução inferior pode resultar, em termos globais de QoS, em valores superiores.

Em termos gerais, os aspectos a considerar para a realização dos testes subjectivos de QoS do serviço de videotex são os seguintes:

- conjunto de observadores não especializados para representar os principais utilizadores do serviço;
- conjunto de observadores especializados que permitam determinar a qualidade das imagens no caso de aplicações mais exigentes;
- imagens com resolução 4:2:2 e 2:1:1 comprimidas com a mesma tabela de quantificação;
- alteração das tabelas de quantificação para obter taxas de compressão superiores e inferiores;
- imagens comprimidas com o formato especificado pela aplicação privada descrita em 5.4;
- condições de teste semelhantes às do funcionamento normal do serviço (distância ao monitor e tempos de transmissão).

Estas experiências exigem sessões prolongadas de observações e análise de resultados pelo que o trabalho de investigação para a determinação de uma medida objectiva, que traduza os resultados subjectivos de qualidade, tem sido intenso. [Miy88] e [KM91] propõem um método baseado em redes neuronais que aproxima uma medida objectiva de qualidade do valor fornecido pelo MOS. A arquitectura proposta é formada por uma rede neuronal, com três camadas, alimentada com a imagem erro e que fornece o MOS à saída. Este processo exige uma fase inicial de treino com base num conjunto de imagens e utiliza o algoritmo de retropropagação do erro [Alm] para actualização dos pesos associados às ligações entre cada um dos neurónios. Os resultados obtidos apresentam uma boa relação entre as medidas subjectiva e objectiva para um sequência de teste composta por 12 imagens. Este método levanta, no entanto, um problema de complexidade da estrutura da rede para imagens do serviço de videotex uma vez que o número de neurónios da camada inicial é igual ao número de pixels da imagem. Os resultados apresentados limitam-se a imagens com dimensão máxima de 256x256 amostras e a imagens monocromáticas.

Capítulo 6

Conclusões

Numa época em que a informação é vital em todas as áreas de actividade, os sistemas ditos multimédia têm vindo a ganhar popularidade. No entanto, as soluções disponíveis actualmente contêm ainda várias falhas ao apresentar versões que, permitindo a introdução de um conjunto de médias, se limitam a acessos locais que viabilizem o sistema, ou então suportando apenas redes privadas e específicas do ambiente em que o serviço é oferecido. A solução para estes problemas e a implementação de um sistema que permita o acesso a informação multimédia exigem a definição clara de um conjunto de requisitos que deverão ser satisfeitos, a especificação detalhada de um terminal com uma arquitectura modular que facilite a inclusão de novas capacidades, o recurso a normas que permitam uma fácil integração com sistemas semelhantes e que sejam capazes de se adaptarem à evolução para novas redes e médias que venham a ser disponibilizados ou exigidos. Um aspecto também de extrema importância que influencia a qualidade de serviço oferecida e o impacto que o sistema em causa terá no público em geral prende-se com as facilidades oferecidas pelo equipamento terminal e pela facilidade de utilização.

O sistema apresentado nesta dissertação representa uma inovação no campo dos serviços de telecomunicações e aplicações multimédia ao ser o primeiro a utilizar, como meio físico de transporte de informação multimédia, a *Rede Digital com Integração de Serviços* e ao oferecer um conjunto de facilidades até agora indisponíveis. O recurso a uma rede pública de comunicações com capacidade razoável, permite o acesso do público em geral e o aparecimento de um tipo de serviços até agora inexistente ou limitado a aplicações locais e a soluções privadas sobre redes locais. Este serviço, ao permitir o acesso a imagens de elevada qualidade num ambiente integrado de comunicação e tratamento de dados, representa uma evolução significativa comparativamente ao serviço público de videotex disponível actualmente. As deficiências do serviço tradicional de videotex, que se traduzem pela fraca qualidade da apresentação dos dados, pela interface de utilizador rudimentar e pouco amigável, pela incompatibilidade com

serviços equivalentes noutros países europeus e pela impossibilidade de expansão e evolução, foram ultrapassadas com a introdução do serviço e do equipamento terminal de videotex alfafotográfico. O mercado potencial para um sistema como o referido nesta tese abrange áreas tão distintas como o turismo, a cultura, o sector imobiliário, a indústria e, também, o meio científico.

A arquitectura modular do *software* implementado no terminal permite a completa integração com o serviço tradicional e a fácil evolução pela inclusão de outros tipos de dados. Na fase actual de desenvolvimento, o terminal aceita dados do tipo alfamosaico e fotográfico nos modos sequencial e progressivo. A apresentação sequencial permite um acesso mais rápido à versão final da imagem, sendo o modo progressivo, apesar de mais lento, adequado a operações de pesquisa uma vez que oferece rapidamente versões preliminares que permitem distinguir a imagem seleccionada e abortar o processo de apresentação antes da conclusão de todas as etapas de descompressão.

Durante a fase de definição do projecto, um ponto considerados de importância vital, para que se atingisse uma boa qualidade de serviço, seria o tempo de acesso à informação. Técnicas de optimização do *software* de descompressão e apresentação das imagens permitiram obter tempos de processamento inferiores aos inicialmente estabelecidos como objectivo, pelo que, na maior na maior parte das aplicações, é possível prescindir do *hardware* auxiliar de processamento de imagem que implica um custo adicional para o terminal. Neste contexto, pode mesmo afirmar-se que os objectivos e expectativas iniciais foram ultrapassados.

A comercialização de um produto implica a satisfação de um conjunto de exigências impostas pelos utilizadores desse produto ou serviço. A utilização de parâmetros de qualidade de serviço, que envolvem não só os resultados subjectivos introduzidos pelos algoritmos de compressão mas, também, o atraso imposto pelo acesso remoto ao servidor e pelo processamento, foram considerados, tendo-se optado pela implementação de um compressor versátil que permite realizar diversas optimizações. As capacidades do terminal acompanham estas funcionalidades tornando o sistema servidor-terminal versátil e atraente. Um pequeno conjunto de testes efectuados permitiram concluir que para um número elevado de aplicações o parâmetro qualidade das imagens oferecidas era considerado de menor importância quando uma pequena diferença nessa qualidade implicava um aumento significativo nos tempos de acesso e visualização. Ao possibilitar a definição da resolução das imagens a utilizar assim como da taxa de compressão imposta, o sistema desenvolvido permite obter versões optimizadas para cada aplicação pela selecção dos parâmetros adequados.

No seguimento deste processo de análise das características das imagens a disponibilizar pelo servidor surgiu a especificação de uma aplicação privada não normalizada. A perda de compatibilidade imposta por esta aplicação não é muito significativa correspondendo, na realidade, a um aumento do desempenho do terminal uma vez que este continua a suportar as

classes normalizadas sendo a referida perda de compatibilidade colocada apenas nos acessos do exterior aos servidores nacionais. No entanto, uma vez que muita da informação armazenada se irá destinar quase exclusivamente a utilizadores nacionais, a classe definida permite uma maior optimização do servidor, ao reduzir a quantidade de memória requerida para armazenar uma imagem, e também do terminal, ao reduzir o tempo de processamento e transmissão e ao libertar os terminais com capacidades gráficas reduzidas das tarefas de quantificação de cor.

A não existência em funcionamento, a nível europeu, de um serviço equivalente, impediu que se efectuassem testes completos de conformidade. No entanto, a utilização de protocolos normalizados e pequenos testes e trocas de informação com outras instituições de investigação permitem prever uma fácil interligação do terminal desenvolvido com os serviços de videotex alfafotográficos que venham a ser disponibilizados.

Como resultado do trabalho desenvolvido no âmbito desta dissertação surgiram, para além de uma primeira versão comercial do terminal de videotex alfafotográfico e do *software* de compressão para a base de dados, várias publicações em conferências nacionais e internacionais.

A introdução de novas facilidades pode ser vista como uma continuação evidente do projecto que conduziu à versão do terminal apresentada nesta dissertação. A arquitectura modular do terminal permite que a sua evolução para um ambiente integrado de aplicações como o MS-Windows implique, basicamente, alterações nas bibliotecas de acesso à memória extensa e à placa gráfica necessitando os módulos de comunicação e tratamento de dados de poucas alterações. A introdução de audio e transferência transparente de dados está também em estudo com vista a tornar o serviço mais atraente e funcional.

Embora já em funcionamento, existe ainda um conjunto de pontos que podem merecer algum trabalho de investigação nesta área. A aplicação privada especificada nesta tese requer ainda algum trabalho de optimização dos algoritmos de ordenação ou a introdução de outros. Métodos baseados em redes neuronais ou algoritmos genéticos deverão ser considerados podendo conduzir a resultados de qualidade superior. A definição final do formato e taxas de compressão adequadas a cada uma das aplicações deve, também, ser efectuada com base num conjunto de testes mais elaborados e com um painel de observadores mais numeroso ou recorrendo a um método objectivo que quantifique os resultados subjectivos.

O trabalho de concepção e desenvolvimento do terminal de videotex alfafotográfico permitiu obter uma visão global do funcionamento de serviços de telecomunicações multimédia. A geração seguinte, envolvendo o acesso a outros tipos de informação, incluindo vídeo, passará pela utilização de objectos hipermédia que implicam a passagem de um conjunto de funcionalidades do servidor para o terminais. A evolução que decorre também no campo das

redes de telecomunicações com a definição da *RDIS de Banda Larga* permitirá viabilizar a introdução de um número alargado de aplicações mais elaboradas. Estas áreas de investigação encontram-se numa fase importante de desenvolvimento permitindo prever, como continuação dos desenvolvimentos realizados, um trabalho futuro aliciente.

Bibliografia

- [Alb85] Antone F. Alber. *Videotex/Teletext Principles & Practices*. McGraw-Hill, 1985.
- [Alm92] Luis B. Almeida. An introduction to multilayer perceptrons. *Técnica*, 67-87, 1992.
- [API93] Telecom Portugal. *Common - ISDN - API, Perfil Portugues*, 1993.
- [BCW90] Timothy C. Bell, John G. Cleary, and Ian H. Witten. *Text Compression*. Prentice Hall, 1990.
- [CC91] W. K. Cham and Y. T. Chan. An order-16 integer cosine transform. *IEEE Trans. Signal Processing*, 39(5):1205-1208, May 1991.
- [CCIRa] CCIR. *Encoding parameters of digital television for studios*. CCIR 601-1.
- [CCIRb] CCIR. *Methods for the subjective assessment of the quality of television pictures*. CCIR 500-4.
- [CH91] S. C. Chan and K. L. Ho. A new two-dimensional fast cosine transform algorithm. *IEEE Trans. Signal Processing*, 39(2):481-485, February 1991.
- [CL91] Nam Ik Cho and Sang Uk Lee. Fast algorithm and implementation of 2-d discrete cosine transform. *IEEE Trans. Circuits Syst.*, 38(3):297-305, March 1991.
- [Cla85] R. J. Clarke. *Transform Coding of Images*. Academic Press, 1985.
- [CSC77] Wen-Hsiung Chen, C. Harrison Smith, and Fralick S. C. A fast computational algorithm for the discrete cosine transform. *IEEE Trans. Commun*, COM-25:1004-1009, September 1977.
- [DP91] Don Pearson, editor. *Image Processing*. McGraw-Hill, 1991.
- [ETSa] ETSI. *Integrated Services Digital Network; ISDN lower layer protocols for telematic terminals*. prETS 300 080.
- [ETsb] ETSI. *Integrated Services Digital Network; Syntax-based Videotex End-to-end protocols*. prETS 300 079.

- [ETSc] ETSI. *Terminal Equipment; Videotex presentation layer protocol, Videotex presentation layer data syntax*. prETS 300 072.
- [ETSD] ETSI. *Terminal Equipment; Videotex presentation layer data syntax; Geometric Display*. prETS 300 073.
- [ETSe] ETSI. *Terminal Equipment; Videotex transparent data*. prETS 300 074.
- [ETSF] ETSI. *Terminal Equipment; Videotex processable data*. prETS 300 075.
- [ETSG] ETSI. *Terminal Equipment; Videotex Terminal Facility Identifier*. prETS 300 076.
- [ETSh] ETSI. *Terminal Equipment; Videotex presentation layer data syntax; Photographic Display*. prETS 300 177.
- [Gal68] R. G. Gallager. *Information Theory and Reliable Communication*. John Wiley, 1968.
- [GG92] Allen Gersho and Robert M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.
- [GJ79] M. Garey and David Johnson. *Computers and Intractability — A guide to the Theory of NP-completeness*. Freeman, 1979.
- [Gra84] Robert M. Gray. Vector quantization. *IEEE ASSP Mag.*, pages 4–29, April 1984.
- [GW87] Rafael C. Gonzalez and Paul Wintz. *Digital Image Processing*. Addison Wesley, 2nd edition, 1987.
- [Haq85] Munsu Haque. A two-dimensional fast cosine transform. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-33(6):1532–1539, December 1985.
- [HBSH92] C. M. Huang, Q. Bi, G. S. Stiles, and R. W. Harris. Fast full search equivalent encoding algorithms for image compression using vector quantization. *IEEE Trans. Image Processing*, 1(3):413–416, July 1992.
- [Hec82] Paul Heckbert. Color image quantization for frame buffer display. *Computer Graphics*, 16:297–307, July 1982.
- [HL92] Daniel S. Hirschberg and Debra A. Lelewer. Context modeling for text compression. In James A. Storer, editor, *Image and Text Compression*. Kluwer Academic Publishers, 1992.
- [Hou87] Hsieh Hou. A fast recursive algorithm for computing the discrete cosine transform. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-35:1445–1461, October 1987.

- [HS63] J. Y. Huang and P. M. Schultheiss. Block quantization of correlated gaussian random variables. *IEEE Trans. Commun.*, 11:289–296, September 1963.
- [HS93] A. C. Hadenfeldt and K. Sayood. Compression of color-mapped images. In *ICC93 — International Conference on Communications*, volume 11, pages 537–541, Geneva, Switzerland, May 1993.
- [HSA84] Geoffrey E. Hinton, Terrence J. Sejnowski, and David H. Ackley. Boltzmann machines: Constraint satisfaction networks that learn. Technical Report CMU-CS-84-119, Carnegie-Mellon University, May 1984.
- [Hua91] Wang Hua. A VQ/DCT scheme for colour images. In *ICC91 — International Conference on Communications*, volume 11, pages 236–240, Denver, USA, June 1991.
- [Huf52] D. A. Huffman. A method for the construction of minimum redundancy codes. *Proc. IRE*, 40:1098–1101, September 1952.
- [HV92] Paul G. Howard and Jeffrey Scott Vitter. Pratical implementation of arithmetic coding. In James A. Storer, editor, *Image and Text Compression*. Kluwer Academic Publishers, 1992.
- [I430] CCITT. *Basic user-network interface — Layer I specification*. CCITT I.430.
- [Jai81] Anil K. Jain. Image data compression: a review. *Proc. IEEE*, 69(3):349–389, March 1981.
- [Jai89] Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.
- [Jay92] Nikil Jayant. Signal compression: Technology targets and research directions. *IEEE J. Select. Areas Commun.*, 10(5):796–818, June 1992.
- [JJS93] Nikil Jayant, James Johnston, and Robert Safranek. Signal compression based on models of human perception. *Proc. IEEE*, 81(10):1385–1421, October 1993.
- [JPEG] ISO/IEC. *Information technology — Digital compression and coding of continuous tone still images*. ISO/IEC IS 10918-1.
- [KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.
- [Koh90] Teuvo Kohonen. The self-organizing map. *Proc. IEEE*, 78(9):1464–1480, September 1990.
- [KM91] Kazunori Kotani and Makoto Miyahara. Objective picture quality scale by neural network. In *Picture Coding Symposium*, pages 13.15.1 –13.15.2, 1990.

- [KR81] Farhad Kamangar and K. R. Rao. Interfield hybrid coding of component color television signals. *IEEE Trans. Commun.*, 29(12):1740–1752, December 1981.
- [LBG80] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Trans. Commun.*, COM-28:84–95, January 1980.
- [Lee84] Byeong Gi Lee. A new algorithm to compute the discrete cosine transform. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-32:1243–1245, December 1984.
- [LMR92] L. Lambarelli, R. Melen, and D. Roffinella. Perspectives on high speed networks and services. Technical Report 6, CSELT, December 1992.
- [LOS⁺93] F. Leal, C. Oliveira, J. Silva, P. Viana. and E. Carrapatoso. PC based architecture to access remote image databases. In *Video Communications and PACS for Medical Applications*, pages 68–74, Berlin, Germany, April 1993.
- [Loy82] S. Lloyd. Least squares quantization in PCM. *IEEE Trans. on Inform. Theory*, IT-28, March 1982.
- [Mat92] V. John Matheus. Multiplication free vector quantization using L1 distortion measure and its variants. *IEEE Trans. Image Processing*, 1(1):11–17, January 1992.
- [Max60] J. Max. Quantizing for minimum distortion. *IEEE Trans. on Inform. Theory*, 6:7–12, March 1960.
- [MH89] Tsutomu Miyasato and Yoshinori Hatori. Inter-videotex conversion from prestel to captain. *IEEE Trans. Commun.*, 37(6):659–663, June 1989.
- [Miy88] Makoto Miyahara. Quality Assessments for Visual Service. *IEEE Commun. Mag.*, 51–60, October 1988.
- [NH89] Arun N. Netravali and Barry G. Haskell. *Digital Pictures: Representation and Compression*. Plenum Press, 2nd edition, 1989.
- [NK88] Nasser M. Nasrabadi and Robert A. King. Image coding using vector quantization: a review. *IEEE Trans. Commun.*, 36(8):957–971, August 1988.
- [OB91] Michael Orchard and Charles A. Bouman. Color quantization of images. *IEEE Trans. Signal Processing*, 39:2677–2690, December 1991.
- [PMK91] Yehoshua Perl, Venkat Maram, and Nageshwar Kadakuntla. The Cascading of the LZW Compression Algorithm with Arithmetic Coding. In *Proceedings of the Data Compression Conference*, April 1991.
- [Pra78] William K. Pratt. *Digital Image Processing*. John Wiley & Sons, 1978.

- [Pry91] Martin de Prycker. *Asynchronous Transfer Mode - Solution for Broadband ISDN*. Ellis Horwood, 1991.
- [Q921] CCITT. *ISDN user-network interface — Data link layer specification*. CCITT Q.921.
- [Q931] CCITT. *ISDN user-network interface layer 3 specification for basic call control*. CCITT Q.931.
- [Rac91] RACE. *Race Common Functional Specifications — Coding, Terminals and CPN*. Race J100.
- [RY89] K. R. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages and Applications*. Academic Press, 1989.
- [Sha48] C. E. Shannon. A mathematical theory of communications. *Bell Syst. Tech. J.*, 27:379–423, jul 1948.
- [SM89] Mohammad R. Soleymani and Salvatore D. Moregera. A fast MMSE encoding technique for vector quantization. *IEEE Trans. Commun*, COM-37(6):656–659, June 1989.
- [Sta94] William Stallings. *Data and Computer Communications*. Macmillan, 4th edition, 1994.
- [Sto92] Matthias-W. Stoetzer. New telecommunication services in europe: Which factors influence their adoption. *Communications & Strategies*, pages 13–31, October 1992.
- [Tan89] Andrew S. Tanenbaum. *Computer Networks*. Prentice-Hall International, 2nd edition, 1989.
- [VLO⁺93] P. Viana, F. Leal, C. Oliveira, J. Silva, and E. Carrapatoso. PC based multimedia videotex. In *4th International Conference on Advances in Communication and Control*, Rhodes, Greece, June 1993.
- [WH89] D. de Werra and A. Hertz. Tabu search techniques: A tutorial and an application to neural networks. *OR Spectrum*, 11:131–141. 1989.
- [WR91] Shukri A. Wakid and Kamie Roberts. Application profile for ISDN. *Proc. IEEE*, 79(2):199–204, February 1991.
- [Wu92] Xialolin Wu. Color quantization by dynamic programming and principal analysis. *ACM Trans. on Graphics*, 11:348–372, October 1992.
- [X25] ISO/IEC. *Information processing systems; Data communication; X.25 Packet Level Protocol for Data Terminal Equipment*. ISO/IEC 8208.

- [X29] CCITT. *Procedures for the exchange of control information and user data between a packet assembly/disassembly (PAD) facility and a packet mode DTE or another PAD.* CCITT X.29.
- [X3] CCITT. *Packet assembly/disassembly (PAD) facility in a public data network.* CCITT X.3.

Agradecimentos

E, finalmente, a sensação de que está terminado! A esta junta-se um conjunto de outras, agradáveis, contraditórias por vezes, a certeza de que, começando tudo de novo, faria melhor mas, ao mesmo tempo a certeza de que esta sensação se irá repetir em tudo o que fizer. Não há dúvida que é fazendo que se aprende! O trabalho de escrita de uma tese exige um esforço individual, isolado. No entanto, não queria deixar de agradecer a um conjunto de pessoas que permitiram que tivesse, afinal, uma componente de trabalho de equipa:

Ao Prof. Doutor Eurico Carrapatoso pelas várias revisões desta tese e sugestões que permitiram, sem dúvida, melhorar a qualidade final do trabalho.

A todos os colegas do grupo de videotex, sem os quais não teria sido possível pôr um sistema a funcionar.

Ao INESC pelas condições de trabalho que me proporcionou.

À JNICT pela bolsa de estudo concedida.

Aos meus pais pela pergunta, repetida vezes sem conta, *Como vai a tua tese? Quando terminas?* Talvez que a devessem ter começado a fazer mais cedo...

Ao Pedro pelo primeiro índice passado de papel para disquete, pelo incentivo para começar a escrever, quando, depois de terminada a implementação inicialmente prevista, o trabalho de escrita parecia não trazer nada de novo. Afinal não era verdade!

