Codificação de Vídeo usando Transformações Geométricas e de Luminância

por

Nuno Miguel Morais Rodrigues

Licenciado em Engenharia Electrotécnica
Departamento de Engenharia Electrotécnica
Faculdade de Ciências e Tecnologia
Universidade de Coimbra
(1997)

Tese submetida para a satisfação parcial dos requisitos
do grau de Mestre em Sistemas e Automação

Departamento de Engenharia Electrotécnica
Faculdade de Ciências e Tecnologia
Universidade de Coimbra

Coimbra
Julho de 2000

Tese realizada sob a orientação de

Vítor Manuel Mendes da Silva

Professor Auxiliar do Departamento de Engenharia Electrotécnica da
Faculdade de Ciências e Tecnologia da Universidade de Coimbra

e de

Sérgio Manuel Maciel de Faria

Professor Coordenador do Departamento de Engenharia Electrotécnica da
Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria
e docente do Mestrado em Sistemas e Automação da F.C.T.U.C.

Aos meus pais,
minha bússola e meu farol

e à Catarina,
meu porto de abrigo

Resumo

Nesta tese estudamos a aplicação de transformações geométricas no domínio da luminância na codificação de sinais de vídeo digital.

As transformações geométricas têm vindo a ser utilizadas na estimação e compensação de movimento pois permitem a representação de um maior número de situações, relativamente aos métodos clássicos baseados apenas em vectores de movimento de translação.

No entanto, a utilização de transformações geométricas espaciais apresenta algumas dificuldades na compensação de fenómenos relacionados com: a exposição de objectos da cena que anteriormente se encontravam encobertos; o aparecimento de novos objectos; e as mudanças causadas pela alteração das condições de iluminação.

É proposta a utilização de um novo tipo de transformações, realizadas no domínio da luminância, de modo a permitir uma compensação mais eficiente dos fenómenos referidos.

Após uma introdução ao problema genérico da codificação de sequências e da utilização de transformações geométricas na estimação e compensação de movimento, é apresentado o estudo das novas transformações no domínio da luminância.

Duas novas técnicas de estimação e compensação de movimento foram desenvolvidas, que combinam as transformações de luminância com as técnicas baseadas em vectores de translação e em transformações geométricas.

As novas técnicas foram testadas em sistemas de codificação de vídeo digital com baixos débitos.

Os resultados obtidos nestes testes foram avaliados e comparados com os dos métodos tradicionais, tendo-se verificado que a sua utilização permite uma melhoria significativa da eficiência da estimação e compensação de movimento.

Foi também estudada a aplicação das novas técnicas desenvolvidas na codificação de sequências estereoscópicas.

Abstract

In this work we study the use of transformations both in the spatial and in the luminance domains, for low bit-rate video coding.

The use of geometric transforms applied in the spatial domain allows for the tracking of complex movements in the scene, in a way that is more efficient than the traditional block matching algorithms.

Although spatial transforms seem to be appropriate to compensate for complex motion, there are still some problems, related with non-uniform intensity changes in spatial and temporal domain and masking between objects.

In order to overcome these problems, new transformations are proposed, that operate in the luminance domain.

After a brief discussion of video coding and motion estimation and compensation using geometric transforms, the new transformations in the luminance domain are introduced.

Two new motion compensation techniques were developed wich combine luminance transformations with block matching algorithms and geometrical transforms.

These techniques were tested in low bit-rate video coding. The results showed that the new transformations are able to compensate a larger number of distortions, improving the motion compensation efficiency.

The use of the new luminance transformations in stereoscopic sequence coding was also studied.

Palavras Chave:

compressão de vídeo, estimação e compensação de movimento, transformações geométricas, transformações de luminância, codificação de sequências estereoscópicas.

Key Words:

video compression, motion estimation, motion compensation, geometric transforms, luminance transforms, low bit-rate video coding.

Agradecimentos

Aos meus orientadores, Doutores Vítor Silva e Sérgio Faria, pelo que me ensinaram, pelo incentivo, pela ajuda e pela amizade.

Ao Instituto de Telecomunicações de Coimbra, pelo apoio ao meu trabalho.

À Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria, pelos meios e condições proporcionadas.

À minha família, pelo que sou e pelo carinho incondicional e apoio infinito.

Aos meus amigos, com alguns dos quais tenho a sorte de trabalhar, pela camaradagem, pelas conversas, pelo que partilharam e me deixaram partilhar, pelo que tornaram mais agradável o tempo deste trabalho.

Ao João Gil, pelo companheirismo e pela inspiração que foi nestes últimos meses e ao João Barreto, pela ajuda e paciência.

E à Catarina, pelo amor, pela paciência, pelo que deu e deixou de receber ao longo deste tempo...

Conteúdo

Lista de Figuras

Glossário

Notação

I_s	Quantidade de informação de um símbolo s
P_s	Probabilidade de ocorrência de um símbolo s
L_s	Número óptimo de bits para o código atribuído a um símbolo s
H	Entropia de uma fonte de informação
L_H	Comprimento médio dos códigos binários de Huffman
B_{ij} ou B(i,j)	Elemento (i, j) de um bloco bidimensional de pixels
y_{ij}	Coeficiente (i, j) da DCT de um bloco de pixels
y_{ij}^q	Coeficiente y_{ij} quantificado
B'_{ij}	Resultado da descodificação do bloco bidimensional de pixels B_{ij}
P ou Q ou R	Bloco de pixels de uma imagem
p_{ij} ou q_{ij}	Elemento (i, j) do bloco P ou Q
Y	Matriz dos coeficientes y_{ij} da DCT de um bloco de pixels
$h(m, n)$	Resposta impulsional de um filtro 2D
$h(m)$ ou $h(n)$	Resposta impulsional de um filtro 1D
$\psi(t)$	*Wavelet* mãe
$\psi_{a,b}(t)$	Família de funções de base obtidas a partir da *wavelet mãe* ψ; a é o factor de dilatação e b é o factor de translação
$\phi_{m,n}(t)$	Família discreta de funções de escalamento
$c_{m,n}$ ou $d_{m,n}$	coeficientes da transformada de *wavelet*
τ	Transformada contractiva
s_τ	Contractividade de uma transformada contractiva τ
$d(v, v')$	Medida de distancia entre um vector v de elementos da imagem e um vector v' pertencente a um dicionário
$e(x, y, t)$	Erro de predição do elemento (x, y) de uma imagem no instante t

(u, v)	Vector de movimento utilizado na compensação de movimento
M ou N	Dimensões de um bloco de pixels de uma imagem
s	Deslocamento máximo utilizado numa pesquisa de vectores de movimento
$X(u, v)$ e $Y(u, v)$	Funções de transformação directa de um vector (u, v)
$U(x, y)$ e $V(x, y)$	Funções de transformação inversa de um vector (x, y)
T_1	Matriz genérica que define uma transformação geométrica
$a_{0..3}$ e $b_{0..3}$	Coeficientes de uma função de transformação bilinear
B_N	Bloco de pixels de uma imagem correspondente a um instante N
B'_N	Resultado da descodificação do bloco B_N
$B_N^{T_1}$	Bloco resultante da aplicação de uma transformação geométrica definida pela matriz T_1 sobre o bloco B_N
A^{T}	Matriz transposta da matriz A
A^+	Matriz pseudo-inversa da matriz A
c	Factor de multiplicação utilizado na compensação da luminância
b	Factor de soma utilizado na compensação da luminância
c_q e b_q	Factores de compensação da luminância quantificados

Abreviaturas

BMA	*Block Matching Algorithms*
BMAI	*Block Matching Algorithm Improved with pixel domain transformation*
BMGT	*Block Matching with Geometrical Transformation*
BMGTI	*Block Matching with Geometric Transform Improved with pixel domain transformation*
CCITT	*Consultative Comitee for International Telephony and Telegraphy*
CGI	*Control Grid Interpolation*
CIF	*Common Intermediate Format*
DC Bidirec	Compensação bidirecional de disparidade
DC	Compensação de disparidade
DCT	Transformada discreta de coseno
DFT	Transformada discreta de Fourier
DHT	Transformada discreta de Hadamard
DST	Transformada discreta de seno

GOB	Grupo de blocos
HDTV	Televisão de alta definição
HGI	*Hierarchical Grid Interpolation*
IDCT	Transformada discreta de coseno inversa
IFS	Sistemas de funções iteradas
inter	Interimagem
intra	Intraimagem
ISO	*International Organization for Standardisation*
ITU	*International Telecommunications Union*
JPEG	*Joint Photographic Experts Group*
KLT	Transformada de Karhunen-Loeve
LAN	Rede de área local
LCD	*Liquid Cristal Display*
LGB	Algoritmo generalizado de Lloyd
MAE	Erro médio absoluto
MB	Macrobloco
MC	Compensação de movimento
MCE	Erro absoluto acumulado
MPEG	*Moving Picture Experts Group*
MSE	Erro médio quadrático
PSNR	Relação sinal-ruído de pico
QCIF	*Quarter CIF*
QMF	Filtros em quadratura
QV	Quantificação Vectorial
RDIS	Rede Digital com Integração de Serviços
SNR	Relação sinal-ruído
TBM	*Triangle-Based Motion Compensation*
TV3D	Televisão estereoscópica
WAN	Rede de área alargada

Capítulo 1

Introdução

O desenvolvimento de novas aplicações que utilizam a codificação e a transmissão de sequências de vídeo digital não se tem limitado à industria televisiva e cinematográfica. Com o aumento da capacidade das técnicas de compressão e dos equipamentos de transmissão, cada vez mais assistimos à proliferação de aplicações que utilizam vídeo digital.

Exemplos destas aplicações são facilmente encontrados em redes de computadores, com ênfase particular para a *internet*, onde se tem vindo a generalizar a transmissão de vídeo digital em tempo real e a utilização de ficheiros com sequências já codificadas e em áreas como a tele-medicina e a tele-operação.

Outra área importante onde um grande esforço tem vindo a ser realizado é a transmissão de vídeo em redes móveis, um campo no qual se espera a curto prazo uma rápida evolução.

Todas estas novas aplicações são possíveis devido à combinação de dois factores principais. Em primeiro lugar, o desenvolvimento de novas tecnologias de transmissão, que permitem o desenvolvimento de aplicações com débitos cada vez maiores. Em segundo lugar, a utilização de técnicas de codificação eficientes de sinais vídeo, que permitem reduzir a quantidade de informação envolvida na codificação destes sinais.

O trabalho descrito nesta tese enquadra-se no campo da codificação de sequências de vídeo digital com baixos débitos.

Neste tipo de aplicações, pretendem-se codificar sequências de imagens com uma taxa de transmissão reduzida e com o mínimo de distorção visual. Aplicações destas técnicas podem ser encontradas nas áreas da videotelefonia e videoconferência, vigilância remota e na referida transmissão de vídeo em redes móveis.

A área da codificação de vídeo digital não é uma nova área de investigação. Muito

trabalho foi realizado e muitos métodos foram já propostos para a resolução dos vários aspectos envolvidos nesta tarefa. Entre as várias propostas, são de realçar as contribuições dos grupos de trabalho da *ITU-T* (sector de telecomunicações da *International Telecommunications Union*), na definição das normas *H.26X*, e do grupo *MPEG* (*Moving Pictures Expert Group*) na definição das normas *MPEG-X*.

Uma das técnicas mais utilizadas nestas normas é a estimação e compensação de movimento. Esta técnica permite explorar a redundância temporal de uma sequência de imagens, constituindo um modo importante de melhorar o desempenho dos sistemas de compressão.

1.1 Motivação

Tradicionalmente, as técnicas de estimação e compensação de movimento utilizam apenas vectores de movimento de translação associados a blocos de pixels. Estes métodos assumem que:

- todos os pixels de cada bloco têm o mesmo movimento;
- os movimentos dos objectos são de translação;
- as condições de iluminação são estacionárias no espaço e no tempo;
- as situações de encobrimento entre os objectos e o aparecimento de novas áreas de fundo podem ser desprezadas.

Estas considerações nem sempre são verdadeiras. Para reduzir o efeito destes problemas, utilizam-se por vezes blocos de pequenas dimensões, de modo a melhorar a eficiência da estimação local de movimento. As técnicas baseadas em vectores de movimento de translação também não são capazes de compensar eficientemente situações mais complexas como sejam: rotações, zoom e outras deformações espaciais.

Por este motivo, algumas técnicas utilizam transformações geométricas na estimação e na compensação de movimento. Estes métodos permitem um melhor tratamento das situações referidas o que se traduz numa melhoria do desempenho global do sistema de codificação.

No entanto, a utilização de transformações geométricas espaciais apresenta também algumas dificuldades na compensação de fenómenos relacionados com:

- exposição de objectos que anteriormente se encontravam encobertos;

- aparecimento de novos objectos;
- mudanças causadas pela alteração das condições de iluminação.

A perda de eficiência das transformações geométricas nestes casos deve-se ao facto de actuarem unicamente sobre as coordenadas dos elementos de cada imagem. Deste modo têm dificuldade em compensar devidamente as alterações verificadas nos valores de luminância desses elementos.

1.2 Objectivos

Neste trabalho pretendemos estudar a aplicação de transformações geométricas à estimação e compensação de movimento. Pretendemos também desenvolver um novo método de estimação e compensação de movimento que utiliza novas transformações que são aplicadas no domínio da luminância do bloco a codificar.

Com as transformações de luminância propostas pretende-se melhorar a estimação das alterações de cada elemento da imagem a codificar, reduzindo assim o erro de predição, o que irá melhorar o desempenho global do sistema de codificação.

As novas transformações desenvolvidas neste trabalho serão testadas em aplicações de codificação de sequências de imagens com baixos débitos. As sequências utilizadas nestes testes são do tipo *cabeça e ombros*, típicas em aplicações de videotelefonia e videoconferência.

Os resultados obtidos serão avaliados e comparados com outros métodos, nomeadamente, baseados em estimação e compensação de movimento com vectores de translação e com transformadas geométricas.

Finalmente, será estudada a aplicação das novas técnicas desenvolvidas na codificação de sequências estereoscópicas.

1.3 Contribuições da tese

O trabalho apresentado utiliza uma abordagem original ao problema da predição dos valores dos elementos de cada bloco da imagem a codificar. Verificou-se que associando à compensação de movimento a compensação dos valores da luminância dos elementos do bloco, é possível estimar melhor cada bloco da imagem a codificar, a partir dos valores de uma imagem de referência. Deste modo, é possível reduzir o erro residual da estimação e compensação de movimento, melhorando assim o desempenho do codificador.

Duas novas técnicas de estimação e compensação de movimento são propostas:

- a técnica *BMAI*[1] associa a transformação da luminância à estimação e compensação de movimento com vectores de translação;
- a técnica *BMGTI*[2] associa a transformação da luminância à estimação e compensação de movimento com transformações geométricas.

Os testes realizados com sequências normais e com sequências estereoscópicas, exibiram uma melhoria do desempenho do codificador quando as novas transformações da luminância foram utilizadas. Este desempenho foi medido em termos da qualidade objectiva (*PSNR*) da sequência codificada e da taxa de transmissão necessária para a sua codificação.

1.4 Organização da tese

Nesta introdução fazemos um enquadramento sumário do trabalho desenvolvido.

No capítulo 2 realizamos uma breve introdução ao problema genérico da codificação de sinais vídeo. Deste modo, são abordadas algumas das questões fundamentais a considerar neste tipo de aplicações, nomeadamente, a quantificação a codificação entrópica e a codificação de imagens estáticas. São também discutidas a estimação e a compensação de movimento, devido à sua importância na maioria dos métodos conhecidos, bem como para o trabalho realizado. Concluímos este capítulo com uma discussão das principais normas actuais de codificação de vídeo.

No capítulo 3 são discutidas as técnicas de estimação e compensação de movimento com transformações geométricas. Neste capítulo são apresentados os principais tipos de transformações geométricas utilizados na estimação e na compensação de movimento. São também descritos alguns exemplos de métodos que utilizam esta abordagem. Entre estes, será realçado o método *BMGT*, ou *Block Matching with Geometrical Transformation*, que será utilizado neste trabalho como referência em termos de comparação.

No capítulo 4 são desenvolvidas as novas transformações propostas nesta tese que utilizam a compensação da luminância dos elementos das imagens. Após uma descrição

[1] *Block Matching Algorithm Improved with pixel domain transformation*
[2] *Block Matching with Geometric Transform Improved with pixel domain transformation*

do método proposto são referidas as características principais do sistema de codificação utilizado.

Um estudo comparativo dos desempenhos das técnicas propostas, relativamente às técnicas tradicionais de compensação de movimento com vectores de translação e transformações geométricas, é apresentado. Para além dos desempenhos relativos, foram também estudados os efeitos da variação de alguns parâmetros de codificação, sendo estes resultados também apresentados e discutidos.

No capítulo 5 estudamos a codificação de sequências estereoscópicas. Neste capítulo são abordados os aspectos gerais da aquisição, visualização e codificação deste tipo de sequências. Foi realizado um estudo de aplicação das transformações estudadas à codificação de sequências estereoscópicas, usando quatro métodos de codificação estereoscópica diferentes. Este estudo é apresentado conjuntamente com uma discussão sobre os resultados obtidos.

O capítulo 6 reúne as principais conclusões extraídas do trabalho realizado, fazendo também uma avaliação geral das técnicas propostas. São também feitas algumas sugestões de trabalho futuro.

Capítulo 2

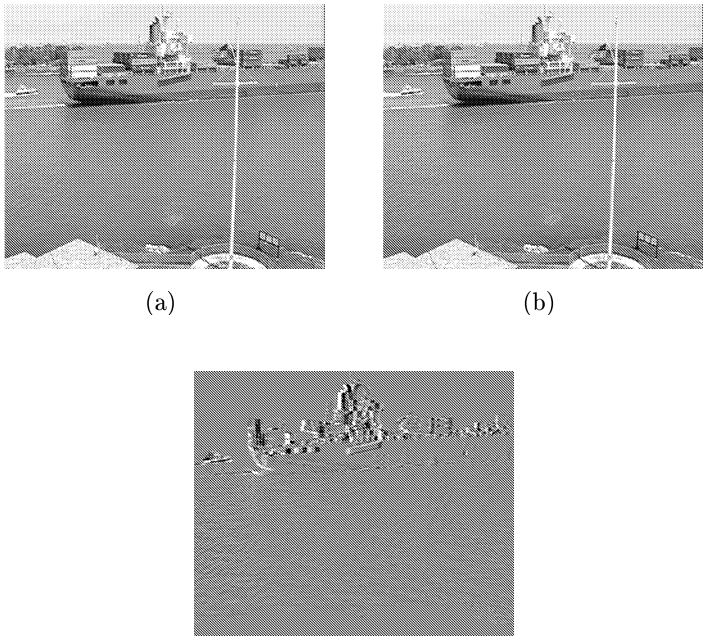
Codificação de Sinais Vídeo

Neste capítulo serão abordados alguns dos aspectos fundamentais da compressão e codificação de sinais vídeo. Uma sequência vídeo é uma série ordenada no tempo de imagens adquiridas em instantes separados por curtos intervalos de tempo. Ao registar em vídeo cenas em que objectos reais se movimentam ao longo do tempo, cada imagem da sequência vai consistir numa projecção num plano do movimento tridimensional dos objectos da cena.

Após uma breve introdução ao problema da codificação de sinais vídeo, serão abordados dois processos com grande relevância para a obtenção de compressão de um sinal vídeo: quantificação e codificação entrópica. Na secção 2.3 serão apresentados os aspectos principais da codificação de uma imagem fixa. Muitas das questões centrais da codificação de uma imagem abordadas nesta secção são igualmente importantes quando consideramos a codificação de uma sequência de imagens. No entanto, a codificação de uma sequência de imagens envolve um conjunto de outras técnicas, algumas das quais serão abordadas com mais detalhe na secção 2.4. Este capítulo termina com uma breve revisão das normas principais de codificação de sinais vídeo definidos actualmente, dos quais se destacam as normas $H26X$ e MPEG.

2.1 Introdução

Como é fácil de perceber pela análise de uma película cinematográfica, cada imagem que compõe um filme é normalmente muito semelhante à imagem anterior, ou seja, existe uma grande *redundância temporal* entre imagens consecutivas de uma sequência. Isto só não acontece em situações muito particulares, nomeadamente, quando ocorre uma mudança de cena.



(a) (b)

(c)

Figura 2.1: Imagens originais de uma sequência: **a)** *imagem(t-1)* e **b)** *imagem(t)*, e **c)** a imagem diferença, resultante da subtracção *imagem(t)-imagem(t-1)*.

Por outro lado, elementos de uma mesma região da imagem tenderão a ter valores muito semelhantes entre si, o que se designa por *redundância espacial*. Assim, a probabilidade de um elemento de imagem (pixel) ter na sua vizinhança elementos de valor semelhante é elevada.

Estes dois tipos de redundância, entre imagens consecutivas de uma sequência e entre elementos adjacentes de uma mesma imagem, estão na base dos dois tipos de técnicas de compressão de vídeo mais utilizadas:

- compressão *interimagem*, que utiliza várias imagens da sequência;
- compressão *intraimagem*, que utiliza a informação de apenas uma imagem;

Uma forma simples de explorar a redundância temporal em sequências vídeo é a construção da imagem diferença entre imagens consecutivas. Consideremos a figura 2.1 onde se representam duas imagens consecutivas de uma dada sequência, designadas *imagem(t-1)* e *imagem(t)*. Representamos na mesma figura a imagem diferença entre as duas imagens referidas. Nesta imagem, os níveis de diferença zero aparecem representados num tom de cinzento neutro, os pontos de diferenças negativas tendem a ser mais escuros, enquanto que os pontos de diferença positiva aparecem representados por zonas mais claras.

Normalmente uma imagem com pouca variação entre os seus elementos, tal como a imagem diferença apresentada na figura 2.1, dá origem a uma compressão mais eficiente, quando utilizamos técnicas de codificação que exploram a redundância espacial. Assim sendo, a transmissão da imagem diferença pode ser muito mais eficiente do que transmitir a imagem original, *imagem(t)*. Estando disponível no descodificador a *imagem(t-1)*, facilmente se reconstrói a imagem actual utilizando a imagem diferença. De um modo geral, a eficiência deste processo depende da eficácia da codificação da imagem de erro.

Nesta perspectiva, quanto menor for a diferença entre imagens consecutivas, mais eficiente será a compressão da imagem de erro. Num caso extremo em que não há movimento entre imagens consecutivas, a imagem de erro não contém qualquer informação, ou seja, o descodificador pode reconstruir a imagem actual por repetição da imagem anterior. Esta é a situação ideal, em que se consegue uma máxima compressão do sinal.

Antes de abordar directamente as principais técnicas de codificação de uma imagem estática, será feita na secção seguinte, uma introdução a dois conceitos fundamentais para a compressão de imagens paradas e sequências de imagens: quantificação e codificação entrópica. Após a discussão do problema da codificação de uma imagem, será abordado o problema mais genérico da codificação de sequências de imagens, onde serão introduzidos outros pontos relevantes, como as técnicas de estimação e compensação de movimento.

2.2 Quantificação e codificação entrópica

A digitalização de uma imagem envolve dois processos fundamentais: a *amostragem* e a *quantificação* [1]. A amostragem produz uma matriz de pixels, que representam pontos uniformemente espaçados da imagem original. O processo de quantificação limita o conjunto de valores possíveis para a representação de cada uma destas amostras.

Se a amostragem for realizada de acordo com o teorema de Nyquist para sinais bidimensionais (2D), é possível recontruir sem erro a imagem original a partir da versão discreta [2]. Isto significa que, nestas condições, a amostragem é um processo reversível.

Com a quantificação, pretende-se converter uma variável amostrada de valor real, numa variável discreta, cujos valores pertencem a um conjunto finito, de modo a reduzir o número de bits necessários para representar o seu valor. Esta operação pode ser representada graficamente por uma função em escada, como a da figura 2.2.

Esta função transforma o valor da variável de entrada t, com um valor $t_k \leq t < t_{k+1}$

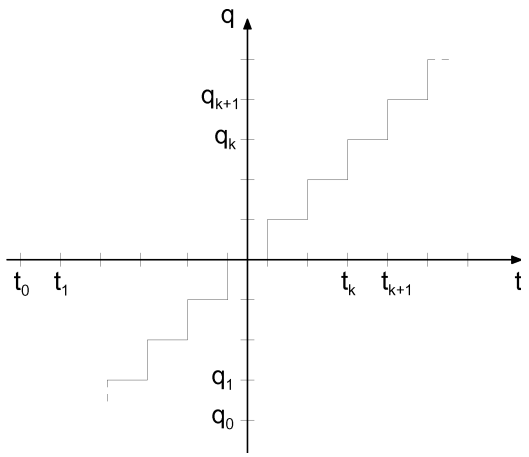


Figura 2.2: Função em escada que realiza a quantificação escalar da variável contínua t.

num valor q_k, onde $k = 0, 1, ..., N-1$. Neste processo vemos que, enquanto o valor da entrada varia de uma forma contínua, o conjunto de valores que a saída pode tomar é limitado e discreto. Quanto maiores forem os intervalos de decisão (dados por $t_{k+1} - t_k$), menor será a precisão da variável quantificada, ou seja, maior será o erro potencialmente introduzido pela operação de quantificação.

No processo de descodificação da variável quantificada, o valor correspondente, q_k, é utilizado para obter o valor de reconstrução. Isto significa que, ao contrário do que acontecia com a amostragem (desde que se respeite o teorema de Nyquist), a quantificação é um processo que introduz um erro irreversível, designado *erro de quantificação*. Este erro é uma fonte de distorção do processo de codificação/descodificação de sinais.

Os processos de amostragem e quantificação são tratados com detalhe em [2], onde nomeadamente são abordados os aspectros da quantificação óptima e da quantificação não-uniforme.

2.2.1 Codificação entrópica

O resultado da compressão, quer intra, quer interimagem, pode ser codificado utilizando códigos de comprimento variável, de forma a explorar a *redundância estatística* dos símbolos produzidos. Este processo designa-se normalmente codificação entrópica. A ideia principal associada a estes métodos é a atribuição de códigos de comprimento reduzido a símbolos que ocorram frequentemente, codificando os símbolos que ocorrem menos vezes com códigos de comprimentos progressivamente superiores.

O modelo probabilístico de ocorrência dos vários símbolos da fonte pode ser obtido por

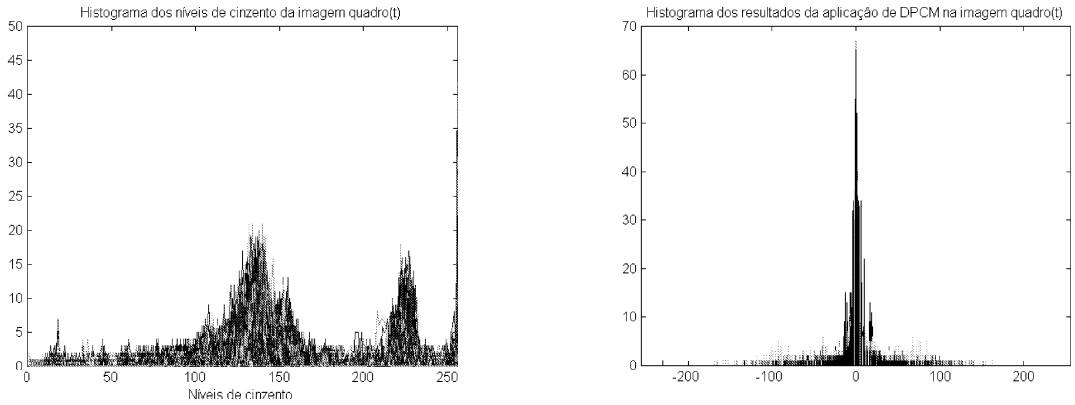
uma análise de uma amostra significativa de dados gerados, ou pode ser definido *a priori*. A taxa de compressão obtida depende directamente da precisão com que o modelo de probabilidade utilizado representa a taxa de ocorrência real para cada símbolo do alfabeto fonte.

Codificação diferencial

Uma das técnicas de processamento de sinal utilizada nas normas de codificação de imagem e vídeo é a *codificação diferencial*, também designada por *codificação preditiva*. Com este processo pretende-se alterar a probabilidade de ocorrência dos símbolos da fonte, de modo a reduzir a sua entropia. Mais uma vez, este processo explora a redundância existente entre elementos vizinhos de uma imagem.

Consideremos uma sequência de pixels a transmitir com N elementos: $p_1, p_2, ..., p_N$. Utilizando codificação diferencial, em vez de codificar directamente cada um destes valores, vamos codificar o erro de predição, dado por $d_k = p_k - p_{k-1}$ (predição de 1ª. ordem), com $k = 1, 2, ..., N$ e $p_0 = 0$. Existindo à partida uma grande correlação entre os elementos vizinhos, esta técnica tem a capacidade de concentrar a função densidade de probabilidade do erro de predição, d_k, em torno de zero.

Na figura 2.3 representam-se os histogramas de ocorrência dos pixels originais, p_k, e do erro de predição d_k. Nesta figura é notório o efeito de transformação conseguido por esta técnica.



(a) (b)

Figura 2.3: Histogramas dos valores dos elementos **a)** da *imagem(t)* e **b)** do erro de predição.

Ao reduzir a entropia da fonte, a codificação diferencial permite aumentar a taxa de

compressão obtida pelas técnicas de codificação entrópica. Seguidamente serão apresentadas brevemente duas técnicas importantes de codificação por entropia, utilizadas nas normas de codificação de imagens e vídeo: a utilização de códigos binários de Huffman e a codificação aritmética.

Códigos binários de Huffman

Um teorema fundamental da teoria da informação, cujo trabalho pioneiro foi desenvolvido por C. Shannon na década de 1940, diz-nos que a quantidade de informação associada à transmissão do símbolo s é dada por [3]:

$$I_s = log_2 \left(\frac{1}{P_s} \right), \tag{2.1}$$

sendo P_s igual à probabilidade da ocorrência do símbolo s. O comprimento óptimo, em bits, para o código atribuído ao símbolo s é dado por:

$$L_s = \lceil I_s \rceil = \left\lceil log_2 \left(\frac{1}{P_s} \right) \right\rceil, \tag{2.2}$$

A entropia da fonte é o número médio de bits por símbolo para todo o alfabeto (conjunto de símbolos) da fonte, e é dada pela expressão:

$$H = \sum_s P_s log_2 \left(\frac{1}{P_s} \right) = \sum_s P_s I_s. \tag{2.3}$$

O valor da entropia de uma fonte dá o limite teórico mínimo do comprimento médio dos códigos atribuídos aos símbolos do alfabeto fonte.

D. A. Huffman [4] desenvolveu um método de codificação que, dado um alfabeto fonte e as respectivas probabilidades de ocorrência, produz um código compacto, ou seja com um comprimento médio mínimo. Este comprimento atinge o limite teórico de Shannon se todas as probabilidades de ocorrência dos vários símbolos da fonte forem potências exactas de 1/2. Genericamente, o comprimento médio dos códigos binários gerados pelo método de Huffman, L_H, respeita os limites:

$$H \leq L_H < H + 1. \tag{2.4}$$

Esta expressão é válida para a codificação de símbolos isolados (sem extensão) do alfabeto fonte e demonstra a eficiência da codificação de Huffman.

O óptimo desta expressão só é atingido se as probabilidades dos símbolos do alfabeto fonte forem exactamente conhecidas e respeitarem a restrição já referida. Na prática, o conjunto de probabilidades de ocorrência de cada um dos símbolos da fonte é normalmente

estimado *a priori*, através da análise de um número significativo de mensagens de teste, sendo depois utilizado na construção do respectivo código de Huffman. A utilização deste código na codificação de uma mensagem particular poderá ser eventualmente sub-óptima, devido a um possível desajuste das distribuições probabilísticas. No entanto, os códigos de Huffman são largamente utilizados nas normas de codificação quer de imagens estáticas quer de sinais vídeo, bem como noutros tipos de sinais digitais, sendo uma das técnicas de codificação entrópica mais importantes.

Codificação aritmética

A codificação aritmética permite a representação em simultâneo de um conjunto de símbolos, tratados como um bloco de dados ao qual é associado um código, representado por um número real pertencente ao intervalo $[0, 1)$ [5]. Na codificação de uma mensagem, cada símbolo do alfabeto fonte é associado a um subintervalo definido pela sua probabilidade de ocorrência, de forma que a união destes subintervalos seja o intervalo $[0, 1)$.

Consideremos por exemplo uma fonte com um alfabeto de 4 símbolos, cujas probabilidades de ocorrência e intervalos de codificação são apresentados na tabela 2.1. Como se pode observar, o intervalo de codificação de um símbolo tem como limite inferior, a probabilidade acumulada de todos os símbolos precedentes, e como limite superior, a probabilidade acumulada de todos os símbolos até ele próprio.

Símbolo	Probabilidade	Intervalo de codificação
α	0.1	[0, 0.1)
β	0.4	[0.1, 0.5)
χ	0.2	[0.5, 0.7)
δ	0.3	[0.7, 1)

Tabela 2.1: Exemplo de alfabeto de uma fonte, probabilidades dos vários símbolos e respectivos intervalos de codificação.

O processo de codificação de uma mensagem é descrito na figura 2.4. A mensagem que pretendemos codificar neste exemplo é: $[\chi\ \alpha\ \delta\ \alpha\ \chi\ \delta\ \beta]$. Para isso começamos por considerar o intervalo inicial $[0, 1)$. Após considerarmos o primeiro símbolo, "χ", tomamos em consideração o subintervalo que lhe corresponde e este passa a ser o novo intervalo de codificação da mensagem. Após cada actualização do intervalo de codificação, todos os intervalos do alfabeto são redimensionados de modo a ajustá-los a um intervalo global

diferente do original. No exemplo apresentado, depois da codificação do primeiro símbolo, passamos a ter o intervalo $[0.5, 0.7)$, subdividido nos intervalos representados.

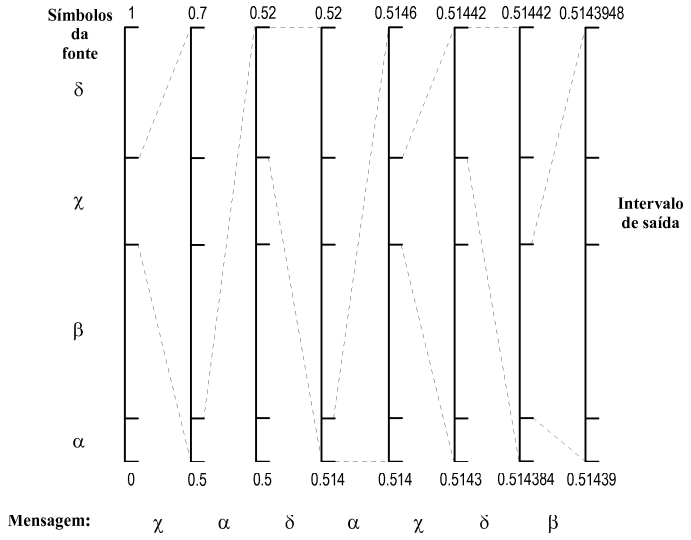


Figura 2.4: Representação da codificação aritmética da mensagem de teste.

Para cada um dos restantes símbolos da mensagem, repetimos o processo de actualização do intervalo de codificação e do redimensionamento dos intervalos de cada símbolo, até atingirmos o último símbolo da mensagem. O resultado da codificação da mensagem é então um subintervalo do intervalo original, neste caso o intervalo $[0.51439, 0.5143948)$. A transmissão da mensagem considerada não requer a transmissão de ambos os limites deste intervalo. Basta enviar para o descodificador um valor real que lhe pertença, por exemplo o limite inferior do intervalo, 0.51439. Este valor identifica rigorosamente um único subintervalo que representa (codifica) a mensagem, o que permite a sua correcta descodificação.

No processo de descodificação começamos por considerar uma vez mais o intervalo original $[0, 1)$ e em cada iteração:

1. escolhemos o subintervalo que contém o valor transmitido (neste caso, 0.51439);

2. escolhemos como símbolo descodificado nesta iteração, o símbolo correspondente a esse intervalo;

3. redimensionamos todos os subintervalos dos vários símbolos, de modo a estarem contidos no novo intervalo de descodificação e repete-se o processo utilizando este novo intervalo.

A tabela 2.2 representa de forma resumida o processo de descodificação, da mensagem considerada no exemplo anterior.

Iteração	Novo intervalo	Símbolo descodificado
1	[0.5, 0.7)	χ
2	[0.5, 0.52)	α
3	[0.514, 0.52)	δ
4	[0.514, 0.5146)	α
5	[0.5143, 0.51442)	χ
6	[0.514384, 0.51442)	δ
7	[0.51439, 0.5143948)	β

Tabela 2.2: Resumo do processo de descodificação da mensagem exemplo.

Note-se que no exemplo apresentado, se assumiu que o comprimento da mensagem a transmitir é conhecido pelo descodificador, de modo que este sabe quando terminar a descodificação da mensagem. Este problema pode ser resolvido se incluírmos no alfabeto fonte um símbolo especial para sinalizar o fim de cada mensagem.

Como a codificação aritmética envolve a transmissão de um valor fracionário, este processo depende muito da precisão da máquina utilizada. Esta precisão está associada ao número de bits utilizados na representação de um número, o que limita o número de símbolos que é possível codificar em simultâneo. Por outro lado, como cada mensagem é representada apenas por um número, só podemos dar início à sua descodificação após a recepção desse valor, o que impede uma descodificação progressiva.

Até agora assumimos sempre um conhecimento prévio das probabilidades de ocorrência dos símbolos da fonte e que as probabilidades não variam ao longo do tempo. Este é o caso nos codificadores aritméticos *estáticos*. No entanto, na codificação aritmética *adaptativa*, as probabilidades de ocorrência dos símbolos da fonte são actualizadas dinamicamente à medida que se avança na codificação da mensagem, num processo designado *modelação*. Este tipo de codificação aritmética permite obter melhores resultados, ao permitir ajustar os intervalos probabilísticos de cada símbolo, em função de cada mensagem. Os principais aspectos da codificação aritmética são abordados de uma forma detalhada em [5].

2.3 Codificação de imagens

Como já foi referido, um dos pontos de grande importância para a compressão de vídeo, é a técnica utilizada na compressão da imagem de erro, processo que é similar à compressão genérica imagens digitais. Nesta secção serão abordados alguns métodos para a codificação de imagens estáticas.

Algumas das técnicas de compressão de imagens estáticas são baseadas na partição da imagem em blocos disjuntos. Estes blocos são depois codificados separadamente utilizando, por exemplo, técnicas baseadas em tranformadas ou *Quantificação Vectorial (QV)*.

Na secção seguinte serão abordados alguns tópicos essenciais da codificação no domínio da transformada. Após esta introdução serão abordadas a transformada discreta de coseno, a partição do sinal em sub-bandas, transformada dicreta de *wavelet* (ou ôndula) e a codificação com fractais. Estas são algumas das técnicas mais divulgadas de codificação de imagens com transformadas. No final desta secção serão discutidos alguns dos aspectos fundamentais da quantificação vectorial.

2.3.1 Codificação com transformadas

A codificação isolada do valor de cada elemento de uma imagem é ineficiente, porque não explora a redundância existente entre elementos vizinhos. A utilização de transformadas é uma técnica eficiente de reduzir a redundância espacial. Se utilizarmos uma transformada adequada, a informação será concentrada num conjunto restrito de coeficientes significativos, menos correlacionados do que os pixels originais.

Outro factor importante que joga a favor da utilização de transformadas são as características do sistema visual humano. A sensibilidade visual de um observador humano varia com as frequências temporal e espacial do sinal observado, seguindo um modelo normalmente aproximado por um filtro passa-banda [6].

Estas características são exploradas nos sistemas de codificação de sinais visuais através de um conjunto de técnicas, designadas normalmente por técnicas de *codificação perceptual*. Um exemplo destas técnicas é a *quantificação perceptual*, que utiliza quantificação não uniforme para codificar de um modo menos preciso as componentes do erro a que somos menos sensíveis.

A figura 2.5 representa resumidamente um esquema de codificação de imagens utilizando transformadas. A imagem original é dividida em blocos rectangulares disjuntos, aos quais se aplica uma transformada bidimensional. O resultado desta operação é num

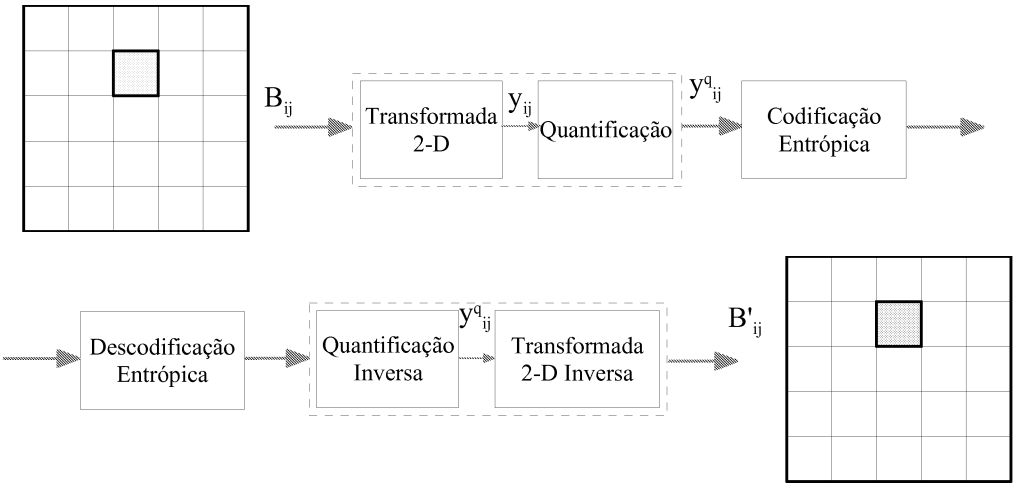


Figura 2.5: Diagramas de blocos de um sistema de codificação/descodificação de imagens com transformadas.

conjunto de coeficientes, y_{ij}, que são depois quantificados, codificados e transmitidos. No descodificador, os coeficientes são recuperados através de descodificação entrópica e da quantificação inversa. A imagem é depois reconstruída utilizando uma transformação inversa da utilizada no codificador.

A escolha da transformada a utilizar neste processo é obviamente muito importante. Como já foi referido, uma transformada adequada deve produzir um conjunto restrito de coeficientes, de forma a obter uma compressão eficaz da informação, utilizando quantificação perceptual e codificação entrópica.

A importância da escolha do quantificador a utilizar neste processo torna-se evidente se nos lembrarmos que é através dele que se consegue grande parte da compressão do sinal, e que este é o único factor de introdução de distorção no bloco reconstruído.

Existem várias transformadas que podem ser utilizadas no processo anteriormente descrito. A transformada ideal é aquela que minimiza a distorção visual entre a imagem original e a imagem reconstruída, para um mesmo número de bits utilizados na codificação.

Algumas das transformadas mais usadas são: a transformada discreta de Fourier (DFT), a transformada discreta de coseno (DCT), a transformada discreta de seno (DST), a transformada discreta de Hadamard (DHT) e a transformada de Karhunen-Loeve (KLT).

A transformada KLT é considerada óptima no sentido em que [7]:

- concentra o máximo de energia no menor número de coeficientes significativos;
- minimiza a entropia total do bloco;

- elimina a correlação entre os elementos do bloco original;

No entanto, a KLT tem vários problemas de implementação, nomeadamente: depende dos dados a codificar (o que não acontece com as outras transformadas) e não existe nenhum algoritmo rápido de cálculo [2].

De entre as outras transformas referidas, a transformada DCT foi adoptada pelas normas de codificação de imagens estáticas e de vídeo, pela sua capacidade de descorrelação dos dados e por existirem vários algoritmos para a sua computação rápida. Os aspectos principais desta transformada e a sua utilização na compressão de imagens serão abordados na secção seguinte.

Uma descrição detalhada de todas as transformadas já referidas (entre outras) é feita em [2].

DCT

A aplicação da transformada discreta de coseno (DCT), a cada um dos blocos disjuntos de uma imagem, é um método utilizado na generalidade das normas de codificação quer de imagens, quer de vídeo. O sistema de codificação utiliza a transformada DCT directa para transformar um bloco de pixels da imagem original, normalmente 8×8, num conjunto de coeficientes. O descodificador reconstrói a imagem original pela aplicação da transformada DCT inversa sobre esses coeficientes. Este processo tem uma capacidade de recontrução perfeita, sendo o erro introduzido pelo processo de quantificação dos coeficientes. Os coeficientes quantificados poderão ser sujeitos a uma codificação entrópica, de forma a reduzir a taxa de transmissão necessária. Este método segue o modelo representado na figura 2.5.

A dimensão usual dos blocos utilizados é 8×8, pois representa o melhor compromisso entre a simplicidade computacional, os requisitos de memória e a eficiência da compressão. A expressão usual da DCT de um bloco P, 8×8, é um novo bloco 8×8, Y, cujos elementos são os coeficientes y_{kl} da transformada, definidos por:

$$y_{kl} = \frac{c(k)c(l)}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} p_{ij}\, cos\left(\frac{(2i+1)k\pi}{16}\right)\, cos\left(\frac{(2j+1)l\pi}{16}\right), \tag{2.5}$$

$$\text{com } k,l = 0,1,...,7 \text{ e } c(k) = \begin{cases} \frac{1}{\sqrt{2}} & se \quad k = 0 \\ 1 & se \quad k \neq 0 \end{cases}.$$

Esta transformada pode ser interpretada como a decomposição do bloco P numa soma pesada de funções de base bidimensionais, que representam frequências espaciais diferentes. As funções de base da transformada DCT, para o caso de blocos 8×8, estão representadas na figura 2.6.

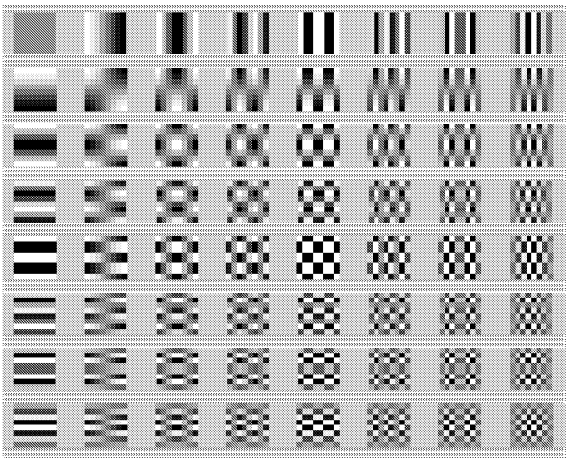


Figura 2.6: Funções de base da transformada DCT para blocos 8×8.

A transformada DCT inversa (IDCT) permite obter os elementos do bloco P a partir dos coeficientes determinados em (2.5), e tem a forma:

$$p_{ij} = \sum_{k=0}^{7} \sum_{l=0}^{7} y_{kl} \, \frac{c(k)c(l)}{4} \, cos\left(\frac{(2i+1)k\pi}{16}\right) \, cos\left(\frac{(2j+1)l\pi}{16}\right), \tag{2.6}$$

com $i, j = 0, 1, ..., 7$.

A razão da adopção da DCT pela maioria das normas está relacionada com as seguintes vantagens [7]:

- se os elementos da imagem original forem muito correlacionados, então a eficiência de compactação da DCT é próxima do óptimo, que é conseguido com a KLT;
- o conjunto das funções base da DCT é independente da imagem a codificar;
- a DCT é uma transformada ortogonal e separável, o que significa que os cálculos dos coeficientes podem ser feitos linha a linha e coluna a coluna;
- como os coeficientes da transformada representam diferentes frequências espaciais, as características do sistema visual humano podem ser facilmente exploradas pela quantificação perceptual de coeficientes y_{kl} da DCT;

- existem vários algoritmos que permitem realizar a computação rápida da DCT e DCT. Alguns destes algoritmos são descritos em [8], juntamente com os diferentes aspectos da utilização da DCT na norma de codificação de imagens estáticas JPEG.

Após o processo de quantificação, a maioria dos coeficientes significativos vai estar concentrada junto da posição y_{00}, que corresponde ao coeficiente de frequência nula, ou valor DC. Por outro lado, o processo de quantificação irá anular a maioria dos coeficientes, dando origem a uma matriz de coeficientes esparsa. Esta matriz pode ser serializada seguindo um varrimento predefinido. A posição e o valor dos coeficientes não-nulos são codificados utilizando uma técnica de *run-lenght coding* e codificação entrópica [8, 9].

Informação mais detalhada sobre a DCT e a sua aplicação na codificação de imagens pode ser obtida em [8, 10].

Codificação em sub-bandas e *wavelets*

Como o próprio nome indica, o principio da codificação de imagens em sub-bandas é a decomposição da imagem num conjunto de diferentes bandas de frequência. Como cada banda tem características distintas, é possível obter uma maior qualidade subjectiva utilizando técnicas que valorizem as bandas mais importantes, em relação às menos relevantes do ponto de vista perceptual.

A maioria dos sistemas de codificação por sub-bandas utiliza bancos de *filtros em quadratura*, ou *QMF's - Quadrature Mirror Filters*, para realizar a decomposição do sinal de entrada. Estes bancos de filtros permitem, na ausência de erros, a reconstrução quase perfeita do sinal original, sem a ocorrência de sobreposição, ou *aliasing* [11]. Um sistema de análise e de síntese com um banco de filtros em quadratura é apresentado na figura 2.7.

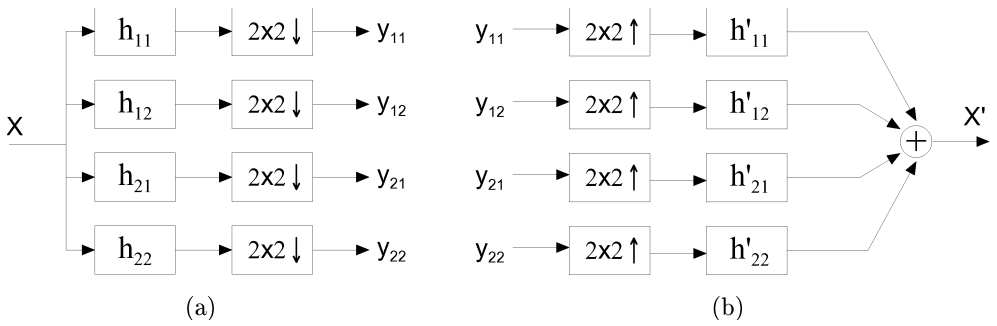


Figura 2.7: Diagrama de blocos de um sistema de processamento de um sinal bidimensional utilizando um banco de filtros de quatro bandas: **a)** sistema de análise **b)** sistema de síntese.

No sistema de análise, cada sub-banda resultante da filtragem inicial é decimada por um factor de dois (utiliza-se uma em cada duas amostras) em cada direcção. Este processo permite que cada banda utilize apenas um quarto dos pixels, o que resulta numa decomposição não expansiva. No processo de síntese, os sinais de cada sub-banda são interpolados por um factor de 2 (um zero é introduzido entre cada duas amostras) segundo cada direcção, antes de serem filtrados e somados, de forma a reconstruir o sinal original. Este processo é ilustrado na figura 2.8, onde se representam os espectros em cada etapa do sistema, para um sinal 1D.

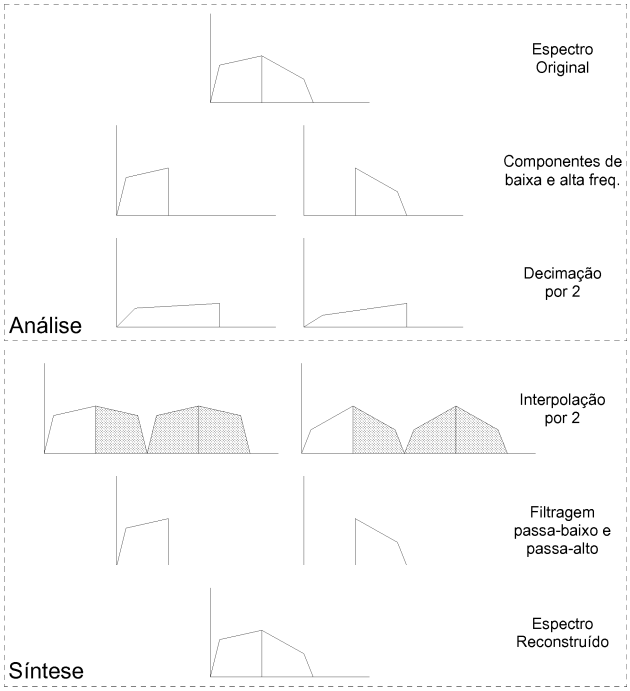


Figura 2.8: Evolução dos espectros de um sinal de entrada ao longo do processo de codificação por sub-bandas (exemplo de sinal 1D).

A implementação do banco de filtros a utilizar na codificação de imagens pode ser simplificada se considerarmos filtros bidimensionais separáveis. Isto significa que, para um filtro 2D com resposta impulsional $h(m,n)$, se verifica a igualdade: $h(m,n) = h_1(m)h_2(n)$, onde $h_1(m)$ e $h_2(n)$ são filtros 1D. Utilizando esta propriedade, o processo de filtragem de uma imagem pode ser feito linha a linha e coluna a coluna, utilizando um banco de filtros 1D formado por um filtro passa-baixo (h) e por um filtro passa-alto (g). Este método é ilustrado na figura 2.9.

Tal como acontecia na codificação utilizando transformadas, a compressão do sinal é conseguida pela codificação perceptual das várias sub-bandas, o que permite favorecer

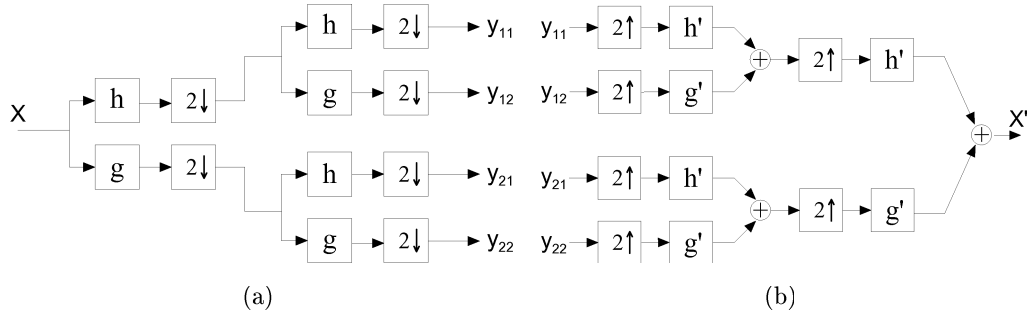


Figura 2.9: Bancos de filtros de **a)** análise e **b)** síntese, para codificação por sub-bandas utilizando a composição de filtros 1D.

os aspectos visualmente mais relevantes. Por outro lado, como o processo de filtragem do sinal original é feito sobre toda a imagem, evitamos os efeitos de blocos presentes em alguns sistemas de codificação que dividem a imagem em blocos.

Wavelets

Um caso particular e interessante de codificação em sub-bandas é a utilização da transformada discreta de *wavelets*[1]. As *wavelets* são famílias de funções de base obtidas a partir de uma função designada *wavelet mãe*, através de dilatações e translações. Para o caso unidimensional, teremos:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\,\psi\left(\frac{t-b}{a}\right), \tag{2.7}$$

onde a representa o factor de dilatação e b o factor de translacção da *wavelet* mãe ψ. Funções de *wavelet* de alta frequência correspondem a valores de $a < 1$, enquanto que as *wavelets* de baixa frequência correspondem a valores de $a > 1$.

A ideia base das transformadas de *wavelets* é a representação de uma função de energia finita, $f(t)$, à custa de uma combinação linear das funções de base $\psi_{a,b}(t)$. Na prática utiliza-se então o somatório de um conjunto discreto de *wavelets*, conseguidas pela alteração de (2.7) para a forma:

$$\psi_{m,n}(t) = {a_0}^{-\frac{m}{2}}\,\psi(a_0^{-m}t - nb_0). \tag{2.8}$$

Um caso particularmente interessante ocorre quando $a_0 = 2$ e $b_0 = 1$, o que torna a família de *wavelets* numa base ortonormal, permitindo a representação de uma função $f(t)$

[1] A expressão *ôndulas* tem ganho relevância como tradução deste termo. No entanto, a utilização da designação original está perfeitamente generalizada e será também por nós utilizada.

através de:

$$f(t) = \sum_m \sum_n c_{mn}\psi_{m,n}(t) \qquad \text{com} \qquad \mathrm{c_{mn}} = \int \mathrm{f(t)}\psi_{\mathrm{m,n}}(\mathrm{t})\mathrm{dt}. \tag{2.9}$$

Na análise com *wavelets* é utilizada uma *representação multiresolução*. Neste caso, o termo a_0^{-m} está associado à ampliação da onda e o termo nb_0 à sua localização. Se $a_0 = 2$, então a análise do sinal é feita octava a octava, e neste caso pode definir-se, para um determinado nível de decomposição m, uma familia de *funções de escalamento*:

$$\phi_{m,n}(t) = 2^{-\frac{m}{2}}\ \phi(2^{-m}t - n), \tag{2.10}$$

que forma também uma base ortonormal. Para $f_m(t)$ igual ao valor da função $f(t)$ no nível de decomposição m temos que:

$$f_m(t) = \sum_n d_{(m+1)n}\phi_{(m+1)n} + \sum_n c_{(m+1)n}\psi_{(m+1)n}. \tag{2.11}$$

Em 1989, Mallat demonstrou que a decomposição multiresolução (2.11) pode ser obtida a partir de um banco de filtros em quadratura, desde que as funções de transferência destes filtros respeitassem certas condições [12]. Esta descoberta simplificou a aplicação deste método e permitiu o desenvolvimento da sua aplicação à codificação genérica de sinais, e de imagens em particular. A decomposição multiresolução baseada em bancos de filtros é representada na figura 2.10, para o caso de um sinal com uma dimensão.

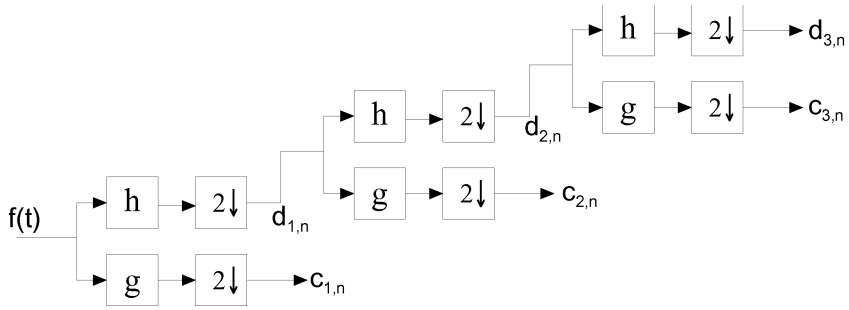
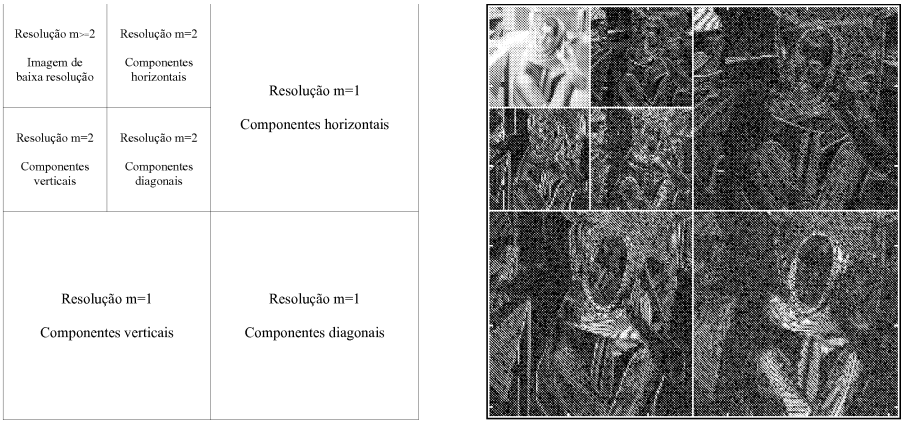


Figura 2.10: Banco de filtros para a implementação da transformada discreta de *wavelet*.

A aplicação desta transformada a uma imagem pode ser interpretada como a decomposição desta num conjunto de bandas com diferentes frequências espaciais. As quatro imagens resultantes da aplicação de um andar de filtragem contêm: as componentes de alta frequência horizontal e vertical, para além das componentes com as baixas e altas frequências segundo ambas as direcções. Estas componentes são normalmente ordenadas



(a) (b)

Figura 2.11: Sub-bandas resultantes da aplicação de uma transformada discreta de *wavelet*.

numa estrutura designada por *pirâmide multiresolução*, cuja organização se encontra representada na figura 2.11a). Na figura 2.11b) é mostrado o caso real de uma pirâmide multiresolução, resultante da aplicação de uma transformada de *wavelet* a uma imagem.

Em [12] é abordada a aplicação das transformadas de *wavelet* na codificação de imagens, sendo feita uma discussão da utilização de várias famílias de *wavelets* para a codificação de imagens. Em [13] é feita uma apresentação da teoria das *wavelets* e da sua relação com os bancos de filtros, focando os aspectos fundamentais a ter em conta na construção destes. Em [11] é apresentado um estudo detalhado das características dos bancos de filtros para codificação com sub-bandas, juntamente com uma discussão dos pormenores fundamentais da teoria dos bancos de filtros em quadratura. Esta discussão é complementada com alguns resultados da codificação de imagens com sub-bandas, obtidos com bancos de filtros de diferentes famílias, incluindo QMF's e *wavelets*.

Codificação com fractais

Um fractal é definido como uma imagem cujos pormenores são idênticos a ela própria, quando representada numa definição inferior. Estes pormenores podem ser construídos a partir de transformações sucessivas realizadas sobre a imagem original, ou parte dela. Isto significa que esta imagem pode ser representada com um grau de precisão infinito, e que pode ser definida a partir de um grupo de transformações e de um conjunto reduzido de parâmetros [14].

A figura 2.12 apresenta algumas iterações do processo de construção de um fractal. O "floco de neve" foi construído originalmente pelo matemático sueco Helge Von Koch,

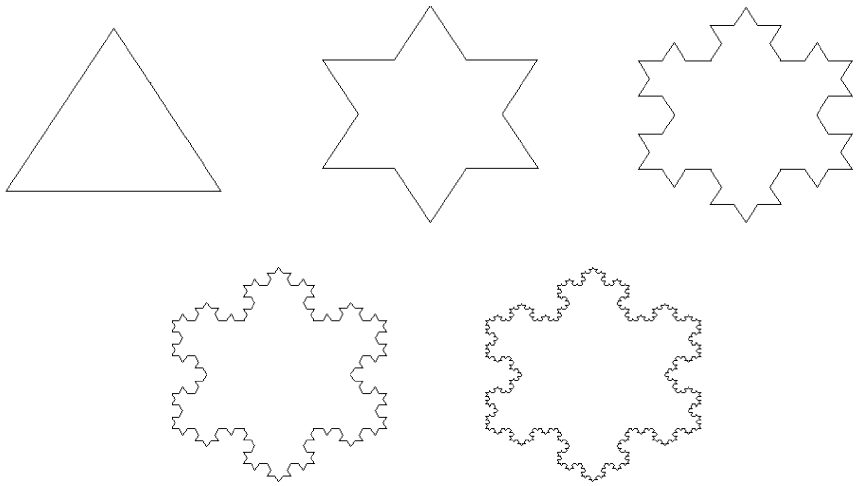


Figura 2.12: Processo de construção do fractal "floco de neve" de Koch. Com o avanço das iterações, o detalhe da representação tende para infinito e a imagem tende para uma imagem final, designada por *atractor*.

em 1904. Partindo de um triângulo equilátero, cada iteração do processo consiste na divisão em três partes iguais de cada um dos seus lados, sendo colocado um novo triângulo equilátero sobre o segmento central, cuja base é eliminada. Como se pode observar, após um número conveniente de iterações, a imagem tende para uma figura final com um nível de detalhe tão grande como se deseje.

Uma das bases matemáticas para a teoria dos fractais é a definição de *transformada contractiva*. Sejam P_1 e P_2 dois pontos da imagem original. Diz-se que τ é uma transformada contractiva se a distância entre P_1 e P_2 na imagem transformada é menor do que a distância original, ou seja:

$$d(\tau(P_1), \tau(P_2)) \leq s_\tau \times d(P_1, P_2), \tag{2.12}$$

com $0 < s_\tau < 1$, onde s_τ é designado por *contractividade* de τ. Esta propriedade indica que a imagem resultante da aplicação sucessiva de uma destas transformadas, produz uma imagem cada vez com mais detalhe, ou seja um fractal. Pretende-se que esta imagem, designada por *atractor*, seja o mais próxima possível da imagem a codificar.

Uma técnica muito popular, que utiliza funções contractivas, designa-se por *sistemas de funções iteradas* (*IFS - Iterated Function Systems*) [15]. Este método pesquisa um conjunto de fractais que possam ser utilizados para descrever e representar a imagem original. Estas técnicas utilizam transformações *afins*, que têm a capacidade de efectuar rotações, translações, mudanças de escala e inclinações da imagem original segundo um

dos seus eixos[2]. A determinação dos parâmetros das funções a utilizar é garantida por um resultado matemático, conhecido como o *teorema de Collage* [15].

A aplicação de fractais na codificação da imagens, pode também ser feita segundo um outro modelo [14], que consiste em três passos:

1. divisão da imagem em blocos disjuntos (eventualmente de dimensões diferentes) que serão codificados individualmente, designados originalmente por *range blocks*;

2. definição de um grupo de blocos, designados por *blocos de domínio* (*domain blocks*), que servirão de base para as transformações fractais;

3. escolha do bloco de domínio e do conjunto de transformações que iteradas sobre esse bloco, produzem uma aproximação de um bloco original (*range block*);

Esta técnica é descrita em [14], juntamente com alguns melhoramentos propostos por vários autores.

A codificação com fractais não é equilibrada em termos da complexidade computacional dos processos de codificação e de descodificação. O processo de codificação, que requer a pesquisa sistemática das funções de transformação a utilizar em cada bloco, é muito mais complexo do que a descodificação, que envolve apenas a aplicação das transformações ao bloco de partida.

Apesar da complexidade do sistema, alguns dos valores de compressão apresentados são impressionantes: aproximadamente 5000:1[3] [16], o que originou o aparecimento de um grande número de publicações nesta área, no início da década de 90. No entanto, estes métodos não se impuseram ao nível das normas de codificação de imagens paradas, que continuam a utilizar principalmente a DCT e *Wavelets*.

2.3.2 Quantificação vectorial

A técnica de quantificação vectorial (QV) processa a imagem no domínio espacial, dividindo-a em blocos de pixels, que são codificados conjuntamente. O fundamento desta técnica surgiu no campo da teoria da informação, onde se provou que a codificação de vectores é sempre mais eficiente do que a codificação de escalares [3]. A principal vantagem da QV é a simplicidade de descodificação, o que permite a visualização dos dados em sistemas

[2]Estas transformadas serão discutidas noutro contexto, na secção 3.1.1

[3]Estes valores são conseguidos apenas para casos (muito particulares) em que a imagem é reconstruída com uma dimensão muito superior à original.

com baixa capacidade computacional. Pelo contrário, o sistema de codificação é muito exigente do ponto de vista computacional, devido ao elevado número de operações que tem de realizar.

A figura 2.13 mostra um sistema de codificação e descodificação de imagem utilizando quantificação vectorial. Nela são representados dois componentes importantes do sistema de codificação: um dicionário de vectores (*codebook*, no original) e um sistema de comparação do vector a codificar com os vectores presentes no dicionário.

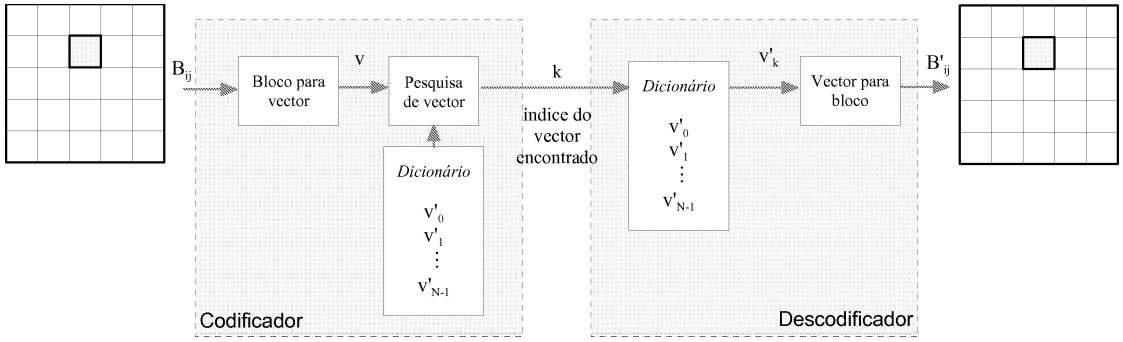


Figura 2.13: Diagrama de blocos de um sistema de codificação com quantificação vectorial.

O princípio fundamental da QV é o estabelecimento de uma correspondência entre um vector de entrada, v, e um vector, v', pertencente ao dicionário e identificado por um índice, que será transmitido. O vector v' é escolhido de forma a minimizar a distância $d(v, v')$, entre os dois vectores. As medidas de distância mais frequentes são o *erro médio quadrático* e o *erro médio absoluto*[4]. Nesta técnica o factor de compressão é obtido à custa da transmissão de um valor inteiro (o índice do dicionário) em vez de um vector com M elementos da imagem.

O processo de descodificação é extremamente simples, correspondendo apenas à consulta do vector do dicionário associado ao índice recebido. A construção deste dicionário é normalmente feita *a priori*, a partir de um conjunto representativo de imagens. O algoritmo mais utilizado para a contrução destes dicionários é o *algoritmo generalizado de Lloyd*, também denominado algoritmo *LGB* [17, 18]. Este algoritmo é um método iterativo, que refina o conjunto de vectores do dicionário de modo a que estes sejam estatisticamente representativos dos vectores a codificar. Verifica-se que a distorção média decresce em cada iteração do método, que é repetido até à obtenção de um conjunto de vectores que permita uma qualidade média desejada.

A qualidade alcançada é no entanto limitada pelo número de vectores do dicionário.

[4]Estas medidas serão discutidas mais detalhadamente na secção 2.4.2

Em princípio, quanto maior é este número, mais fielmente se pode esperar que o bloco de entrada seja representado. No entanto, um número elevado de vectores de codificação faz com que a complexidade do sistema aumente exponencialmente. Esta complexidade depende principalmente de dois factores: a medida de distorção utilizada e o método de pesquisa empregue na determinação do melhor vector do dicionário a utilizar em cada caso.

Uma descrição dos principais algoritmos de pesquisa pode ser encontrada em [17], juntamente com uma apresentação dos aspectos principais da quantificação vectorial. A utilização da QV na compressão de imagens é apresentada também em [18]. Outros exemplos da aplicação de métodos de QV a várias áreas de processamento de sinal são apresentados em [9].

2.4 Compressão de sequências de imagens

As técnicas de codificação de imagens paradas exploram principalmente a correlação existente entre os elementos da imagem. As técnicas de compressão de sequências de imagens tiram partido não só desta correlação, mas também da redundância existente entre imagens consecutivas. Estas técnicas dividem-se em dois grupos fundamentais: *compressão híbrida espacial e temporal* e *compressão espaço-temporal*, representadas esquematicamente na figura 2.14 [19].

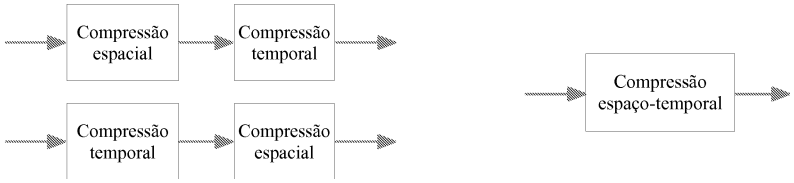


Figura 2.14: **a)** Compressão híbrida espacial e temporal e **b)** compressão espaço-temporal.

Seguidamente serão discutidos os aspectos fundamentais de cada um destes grupos de técnicas, com particular ênfase nos sistemas de codificação híbridos, devido à sua muito maior utilização entre os sistemas de codificação de vídeo e porque o alvo do trabalho de investigação apresentado se enquadra neste grupo.

Os métodos de estimação e compensação de movimento são aspectos essenciais para o sucesso dos esquemas híbridos de compressão e serão apresentados na secção 2.4.2.

Na secção 2.4.3 será apresentada uma técnica alternativa de codificação de imagens

em movimento, a codificação baseada em modelos, que é actualmente alvo de um grande esforço de investigação, tendo sido incorporada na mais recente norma de grupo MPEG.

No final deste capítulo, serão brevemente apresentadas algumas das normas de codificação de vídeo definidas actualmente.

2.4.1 Esquemas de compressão de sequências de imagens

Compressão espaço-temporal

Nestas técnicas de compressão, o espaço tridimensional (x, y, t) da sequência é codificado utilizando transformadas 3D, o que permite a exploração em simultâneo das redundâncias espacial e temporal. Esta é a razão da designação deste tipo de técnicas de compressão.

Mais uma vez a escolha da transformada a utilizar é fundamental. A transformada mais utilizada é a DCT tridimensional. Recentemente foram propostos novos esquemas de codificação que utilizam transformadas de *wavelets* 3D, codificação por sub-bandas 3D e codificação tridimensional com fractais [19].

A eficácia destas técnicas depende do nível de movimento entre as várias imagens da sequência. Se existir um grande movimento, a correlação temporal decresce, o que reduz a taxa de compressão conseguida pela transformada. Os métodos de estimação e compensação de movimento, que abordaremos em secções seguintes permitem resolver em parte este problema. No entanto, são de difícil aplicação em codificadores do tipo espaço-temporal, devido ao facto da informação do movimento existente na cena não ser por eles explicitamente explorada.

Neste aspecto, os esquemas de codificação híbridos conseguem um melhor desempenho, porque exploram independentemente as correlações espacial e temporal. Este tipo de técnicas será discutido na secção seguinte.

Compressão híbrida espacial e temporal

Como já referimos, a simples utilização da imagem diferença permite a exploração da redundância temporal de uma sequência. Por outro lado, a codificação desta imagem utilizando uma das técnicas descritas na secção anterior, permite explorar a correlação espacial existente nesta imagem. A utilização de um codificador entrópico, associado a este sistema, permite explorar a redundância probabilística dos símbolos gerados, aumentando a taxa de compressão global.

A exploração separada de cada uma destas redundâncias é o princípio dos codificadores híbridos. A figura 2.15 representa o diagrama de blocos de um codificador híbrido típico.

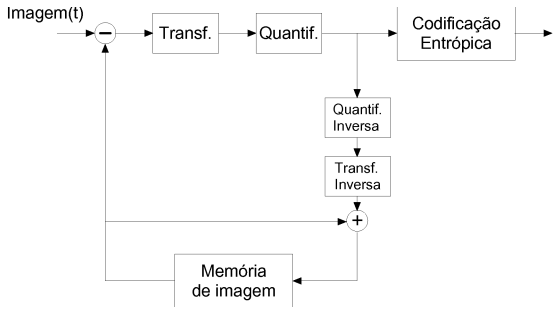


Figura 2.15: Esquema de compressão híbrida espacial e temporal [19].

Associado a este tipo de codificadores, utilizam-se normalmente processos que visam melhorar a previsão da imagem actual. Este facto permite a diminuição do erro residual, o que aumenta a eficácia do sistema. Estes processos designam-se por *estimação e compensação de movimento* e são apresentados seguidamente.

2.4.2 Estimação e compensação de movimento

Numa sequência de imagens sem movimento, seria apenas necessário enviar para o descodificador a primeira imagem da sequência e a indicação que todas as imagens seguintes seriam idênticas. Obviamente, esta não é uma situação comum.

De um modo geral, existem diferenças entre imagens consecutivas, devido ao movimento dos objectos da cena, a movimentos da câmara, a alterações nas condições de iluminação e ao ruído introduzido pelo sitema de aquisição de imagem. O movimento de uma cena pode ser estimado e representado por um conjunto de vectores de movimento, associados a cada região da imagem. O processo de *estimação de movimento*, procura determinar estes vectores de movimento.

A *compensação de movimento* considera um bloco da imagem de referência deslocado segundo as coordenadas do vector de movimento correspondente. Se o bloco deslocado for perfeitamente idêntico ao bloco da imagem original, então o vector de movimento é suficiente para a sua codificação. Neste caso, o ganho em compressão conseguido é enorme.

Em imagens naturais com movimento, raramente acontece que o erro entre os blocos seja completamente anulado pela compensação de movimento de uma imagem para a outra. De um modo geral, a compensação de movimento permite reduzir o erro entre o bloco original e o bloco de referência. Esta redução possibilita uma codificação mais

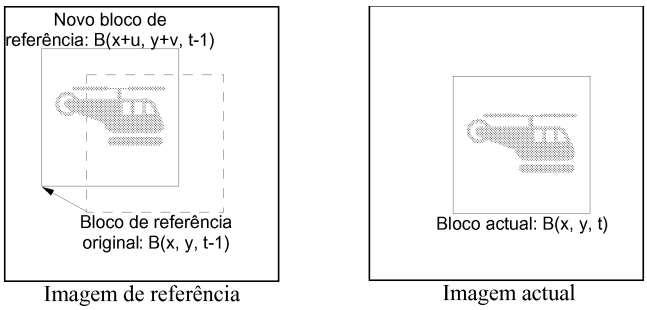


Figura 2.16: Estimação e compensação de movimento de translação.

eficiente do erro de predição, mesmo considerando a informação suplementar associada à transmissão do vector de movimento.

O erro de predição, após o processo de compensação de movimento, pode ser escrito como:

$$e(x, y, t) = B(x, y, t) - B(x - u, y - v, t - 1), \tag{2.13}$$

onde $B(x, y, t)$ representa os valores dos elementos do bloco B da $imagem(t)$ de coordenadas (x, y), e $B(x - u, y - v, t - 1)$ são os valores dos elementos do bloco correspondente na $imagem(t - 1)$ (imagem de referência), de coordenadas $(x - u, y - v)$.

A imagem de referência não é necessariamente a imagem anterior, como veremos mais adiante. Tem no entanto de ser uma imagem já codificada e transmitida, de modo a estar disponível no descodificador.

O vector (u, v) é o resultado da estimação de movimento para o bloco $B(x, y)$ da imagem original.

A figura 2.17 representa o esquema de um sistema de codificação e descodificação que utiliza estimação e compensação de movimento e a transmissão da imagem de erro residual. Neste tipo de sistemas, o processo de compensação de movimento é necessário tanto no codificador como no descodificador. No entanto, a estimação de movimento é feita apenas na codificação.

Os vectores de movimento (u, v), resultantes da estimação de movimento, são codificados e transmitidos juntamente com o erro de predição e são utilizados na reconstrução da imagem actual. No descodificador, a imagem compensada em movimento é somada à imagem de erro, de modo a obter a imagem reconstruída.

O codificador reconstrói também a imagem actual, de modo a obter uma cópia da imagem presente no descodificador, que será utilizada na predição da imagem seguinte da

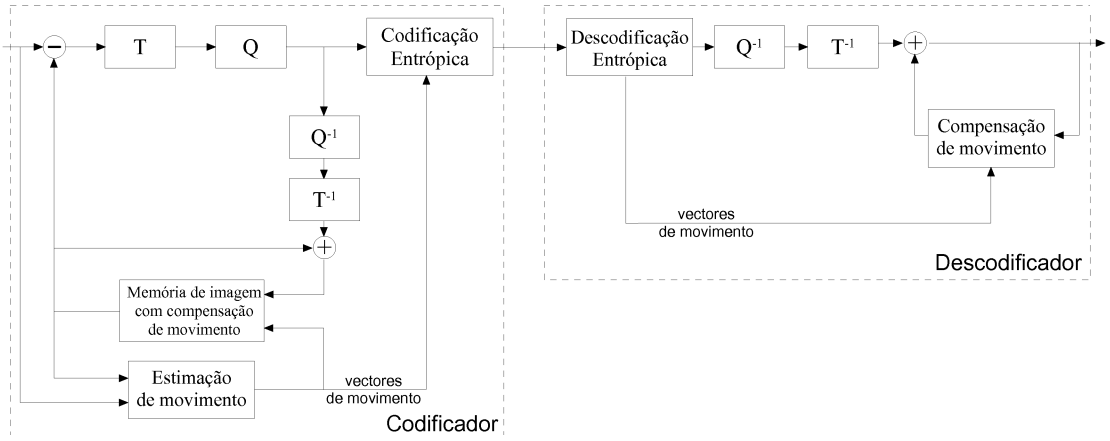


Figura 2.17: Sistema de codificação híbrido com estimação e compensação de movimento.

sequência.

Nas secções seguintes serão discutidos os aspectos principais dos esquemas de estimação e compensação de movimento de translação, usados nas normas de codificação de sinais vídeo, devido à sua simplicidade e eficiência. Outras técnicas, que utilizam, por exemplo, transformadas geométricas para representar as alterações verificadas nas imagens, são um dos temas centrais deste trabalho, sendo a sua abordagem realizada em capítulos posteriores.

Critérios de correspondência entre blocos

De um modo geral, o vector de movimento associado a um bloco da imagem a codificar é escolhido de modo a maximizar a correlação ou minimizar o erro entre este bloco e o bloco que lhe corresponde numa imagem de referência. Devido à sua maior complexidade computacional, a correlação entre blocos é normalmente menos utilizada do que as medidas de erro. As medidas de erro entre blocos mais utilizadas são o *erro médio absoluto* (*MAE* - *Mean Absolut Error*) e o *erro médio quadrático* (*MSE* - *Mean Squared Error*).

Seja B a representação de um bloco da imagem actual de dimensão $M \times N$ e R o bloco que lhe corresponde na imagem de referência. A função de erro médio absoluto, para um bloco com posição inicial (x, y), MAE, é definida pela seguinte expressão:

$$MAE(u,v) = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |B(x+i, y+j) - R(x+i+u, y+j+v)| \,. \qquad (2.14)$$

Nesta equação, os valores de u e v pertencem ao intervalo $[-s, s]$ definido pela janela de pesquisa utilizada (de dimensão $(2s+1) \times (2s+1)$). O vector de movimento (u, v) é

escolhido de modo a minimizar o valor da função de erro, $MAE(u, v)$. Outra possibilidade é a utilização da medida do erro médio quadrático *MSE*, dada pela seguinte expressão:

$$MSE(u,v) = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left(B(x+i, y+j) - R(x+i+u, y+j+v)\right)^2 . \qquad (2.15)$$

Em vez do MAE, utiliza-se por vezes o valor do *erro absoluto acumulado*, dado por:

$$MCE(u,v) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |B(x+i, y+j) - R(x+i+u, y+j+v)|, \qquad (2.16)$$

porque permite operar com números inteiros.

Idealmente o critério de correspondência entre os dois blocos devia ser baseado não no erro entre eles, mas na quantidade de dados gerados pelo processo de codificação. No entanto, não se utiliza normalmente esta solução devido à sua elevada complexidade computacional. Para além disso, verificou-se a existência de uma relação linear aproximada entre o valor do MAE e a quantidade de bits gerados, para a codificação do bloco utilizando técnicas MPEG [10]. Este facto mostra bem a validade da utilização deste critério de erro.

Algoritmos para estimação de movimento

Uma vez definido o critério de erro a utilizar, por exemplo o erro médio absoluto, podemos determinar o vector de movimento associado a cada bloco, calculando o valor $MAE(u, v)$ para todos os vectores de movimento admissíveis, e escolhendo o vector que minimiza esse valor. Este processo é designado por *pesquisa exaustiva*, e exige o cálculo do MAE em todas as $(2s+1)^2$ posições da janela de pesquisa considerada. É óbvio que a complexidade computacional deste método é bastante elevada. No entanto, a determinação do vector óptimo é garantida.

Devido às exigências de processamento impostas pela pesquisa exaustiva, foram desenvolvidos métodos alternativos que permitem realizar uma estimação de movimento com menor complexidade computacional. No entanto, estes métodos são sub-óptimos, no sentido em que não garantem a determinação do vector que minimiza a medida de erro. Em contrapartida permitem a obtenção de resultados muitas vezes satisfatórios.

Existem duas soluções principais para reduzir a complexidade do processo de estimação de movimento. A primeira restringe a pesquisa dos vectores de movimento a um subconjunto dos pontos da janela considerada, enquanto que a segunda determina o erro

entre os blocos utilizando apenas parte dos seus elementos. Podem também utilizar-se combinações destas duas técnicas.

Do primeiro grupo salientam-se as técnicas de pesquisa logarítmica. A ideia base, subjacente a estas técnicas, é a utilização de apenas um conjunto pré-definido de pontos, pertencentes a uma região da janela de pesquisa. A região considerada em cada iteração é centrada em torno do mínimo determinado na iteração anterior, e as suas dimensões são reduzidas, concentrando assim a pesquisa numa área mais reduzida.

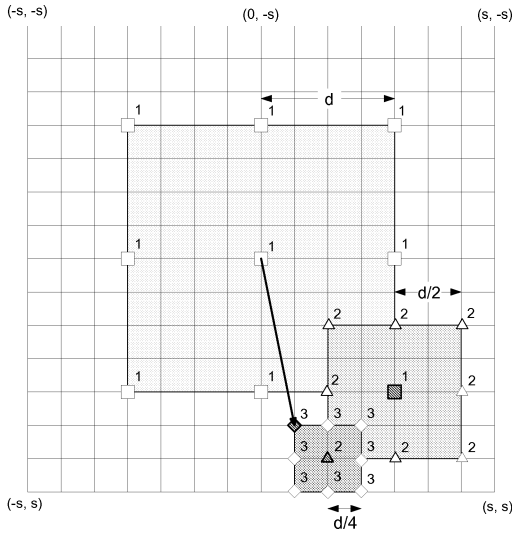


Figura 2.18: Algoritmo logarítmico de pesquisa em três passos.

Na figura 2.18 resentamos esquematicamente um método de pesquisa logarítmica bidimensional denominado *pesquisa em três passos* [20, 7]. Deste modo, a estimação do movimento é feita unicamente em $8\lceil log_2 s \rceil + 1$ pontos, reduzindo desta forma a complexidade da estimação de movimento.

Outro tipo de pesquisa logarítmica é a *pesquisa ortogonal* [21], que é representada na figura 2.19. Esta pesquisa é realizada segundo as direcções horizontal e vertical, em cada iteração. O melhor dos 3 pontos pesquisados horizontalmente é escolhido, servindo de base para a pesquisa vertical (ou vice-versa). Após cada iteração, a região considerada é reduzida para metade e centrada em torno do ponto determinado. Este método requer apenas o cálculo da função de erro em $4\lceil log_2 s \rceil + 1$ pontos.

A principal limitação dos métodos de pesquisa não exaustiva é a possível convergência para um mínimo local da função erro, diferente do mínimo global. Em contrapartida, estes métodos são uma boa opção para a redução da complexidade computacional do processo da estimação de movimento.

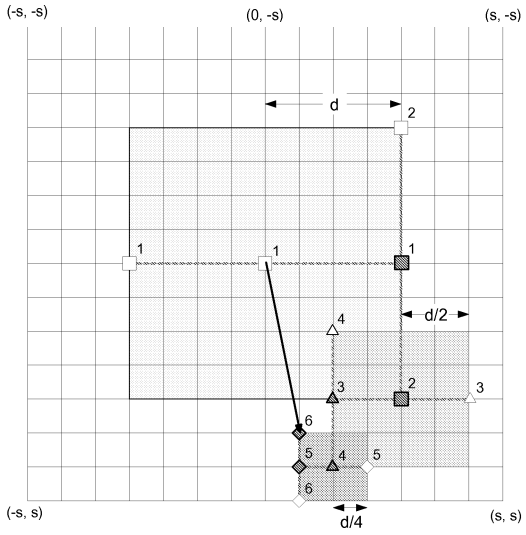


Figura 2.19: Algoritmo logarítmico de pesquisa ortogonal.

Como já foi referido, outra técnica utilizada na redução da complexidade é a utilização de uma aproximação da função de erro entre blocos, que utiliza apenas um subconjunto dos pontos. Uma das justificações possíveis para a utilização deste método baseia-se no seguinte facto: ao considerarmos que o movimento é homogéneo para todos os elementos do bloco, esse movimento pode ser determinado apenas a partir de alguns desses elementos. A escolha dos elementos a utilizar deve, no entanto, ser feita de uma forma criteriosa [7].

Em [10] são apresentadas outras técnicas de estimação do vector de movimento de translação. Para além disso, outros aspectos interessantes deste processo são discutidos.

Estimação de movimento com precisão sub-pixel

Não é natural esperar que o movimento de um objecto na imagem esteja perfeitamente relacionado com a grelha definida pelas posições dos pixels na imagem. No entanto, todos os vectores de movimento referidos até agora têm componentes inteiras. Diz-se por isso que têm uma precisão inteira, ou de um pixel. Para se conseguirem vectores de movimento que representem melhor os deslocamentos reais dos objectos da imagem, a estimação de movimento deve ser feita à custa de uma maior resolução, nomeadamente, utilizando precisões de fracções de pixel, ou precisões sub-pixel.

Obviamente, o aumento da resolução dos vectores de movimento implica uma maior complexidade do processo de estimação de movimento. No entanto, a generalidade das normas de codificação vídeo actuais permitem a utilização de resoluções de meio pixel para os vectores de movimento.

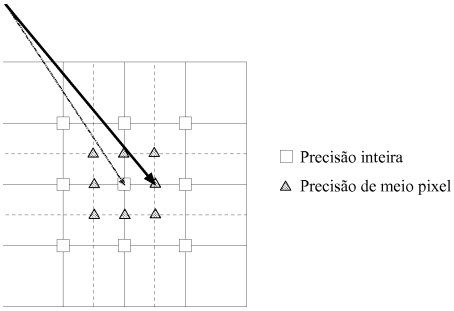


Figura 2.20: Estimação de movimento com precisão de meio pixel.

O aumento da resolução dos vectores de movimento pode ser conseguido por interpolação das imagens original e de referência. Utilizando blocos interpolados, podemos usar qualquer um dos métodos de estimação de movimento já discutidos. Esta solução aumenta consideravelmente a complexidade do processo de estimação de movimento, devido ao aumento da dimensão dos blocos.

Uma alternativa possível consiste em realizar a estimação de movimento em dois passos. No primeiro passo determina-se o vector de movimento com precisão de um pixel, utilizando, por exemplo, um dos métodos descritos anteriormente. No passo seguinte, refina-se o resultado obtido, considerando também os pontos interpolados que existem em torno do pixel com as coordenadas determinadas no passo anterior (ver figura 2.20).

A utilização de vectores de movimento com precisão de fracções de pixel exige sempre a interpolação das imagens consideradas, de modo a determinar a imagem de erro.

Estimação bidireccional de movimento

A estimação bidireccional de movimento consiste na utilização de duas imagens de referência, uma anterior e outra posterior ao instante t, correpondente à imagem que se pretende codificar. Na figura 2.21 representamos este processo. O vector (u_a, v_a) representa o movimento estimado para o bloco B, utilizando o bloco A como referência. O vector (u_c, v_c) representa o vector de movimento para o bloco B, considerando a $imagem(t+n)$ como referência. Estes vectores podem ser determinados utilizando, por exemplo, as técnicas de estimação de movimento já referidas.

A reconstrução do bloco B pode ser feita à custa dos elementos dos blocos A ou C compensados em movimento, ou da média dos valores dos elementos destes blocos. Neste caso, o valor do elemento do bloco reconstruído será calculado utilizando os elementos dos

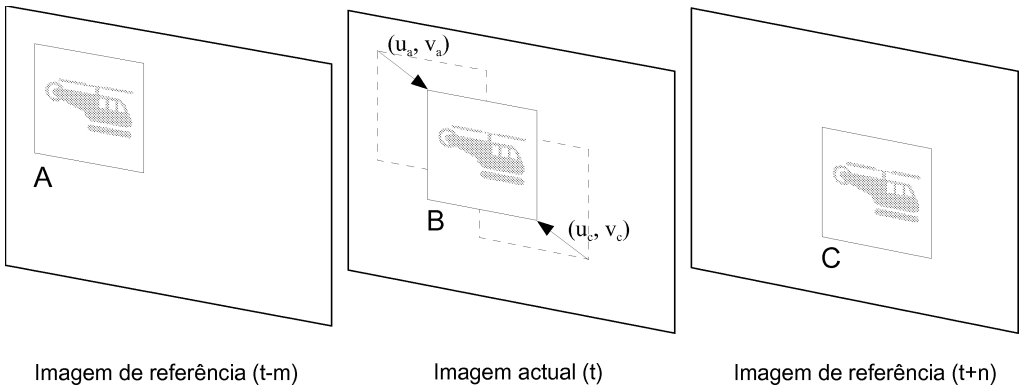


Figura 2.21: Estimação bidireccional de movimento.

blocos A e C com compensação de movimento, pela fórmula:

$$B(i,j) = \left\lceil \frac{A(i+u_a, j+v_a) + C(i+u_c, j+v_c)}{2} \right\rceil, \tag{2.17}$$

para todos os pontos (i,j) do bloco B. Normalmente é escolhido o tipo de compensação que minimiza o erro do bloco.

Este processo exige que o descodificador possua não só a imagem anterior $(t-m)$, mas também a imagem $(t+n)$, posterior à imagem actual. Isto implica um reordenamento das imagens transmitidas pelo codificador. O descodificador deve ser capaz de armazenar e reordenar as imagens reconstruídas, de modo a respeitar a ordem original da sequência, no momento da sua apresentação.

Árvores quaternárias

A divisão da imagem original em blocos pretende obter zonas homogéneas, às quais será atribuído um só vector de movimento, o que poderá não ser adequado se um bloco abranger vários objectos com movimentos distintos. A divisão dos blocos originais em blocos mais pequenos permite reduzir este fenómeno, o que diminui o erro da estimação de movimento.

Variar o tamanho dos blocos utilizados na estimação do movimento permite ajustá-los melhor ao tamanho dos objectos que se movem na imagem. Deste modo, a compensação de movimento de uma cena com objectos de vários tamanhos e movimentos distintos, será feita utilizando blocos ajustados ao tamanho e movimento dos objectos da imagem.

A segmentação dos blocos originais pode ser feita pela sua partição em quatro novos blocos, como é apresentado na figura 2.22. A decisão de dividir cada bloco é baseada normalmente num teste de homogeneidade. Vários testes foram propostos para este efeito [22, 23, 24].

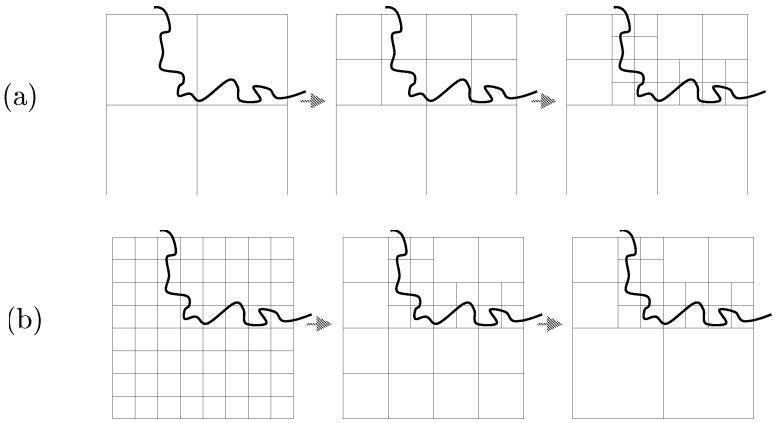


Figura 2.22: Construção de uma árvore quaternária, segundo os métodos de **a)** *top-down* e **b)** *bottom-up* [22].

O processo de segmentação é repetido até atingirmos uma dimensão mínima para os blocos, ou até que não seja necessária qualquer divisão, de acordo com o critério de homogeneidade utilizado. Este modo de segmentação é normalmente designado por *top-down* (ver figura 2.22a)), por partir inicialmente de blocos de grande dimensão, que são divididos em blocos progressivamente mais pequenos. Um método alternativo de segmentação parte de blocos com a dimensão mínima, que depois são agrupados (ou não), em blocos progressivamente maiores. Este método é denominado *bottom-up* e está representado na figura 2.22b).

A representação da imagem segmentada deste modo é feita através de uma árvore quaternária[5]. Nesta representação cada bloco homogéneo dá origem a uma folha, enquanto que um bloco a dividir dá origem a um nó com quatro novos ramos. Cada posição da árvore identifica perfeitamente um bloco quanto à sua dimensão e localização.

Para além disso, uma árvore quaternária pode ser codificada de um modo extremamente eficiente. A figura 2.23 representa a árvore quaternária resultante do processo de segmentação ilustrado na figura 2.22.

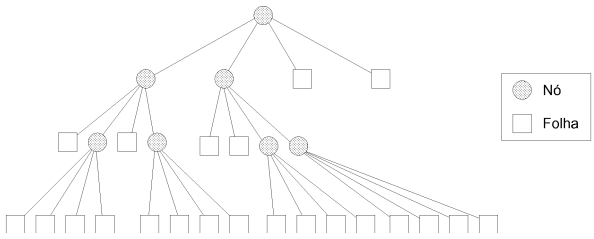


Figura 2.23: Árvore quaternária que representa a segmentação efectuada na figura anterior.

[5]Tradução do inglês *quadtree*.

A aplicação de árvores quaternárias foi bastante utilizada no desenvolvimento de sistemas de compressão de imagem por transformadas e quantificação vectorial [25, 24].

Como já foi referido, a aplicação de árvores quaternárias é também muito utilizada em métodos de estimação e compensação de movimento [22, 26]. Esta utilização é vantajosa porque permite utilizar blocos de dimensão variável, que agrupam elementos com o mesmo tipo de movimento. Enquanto que a codificação de blocos de grandes dimensões permite reduzir o número de vectores de movimento, a utilização de blocos de menores dimensões permite obter uma melhor qualidade da imagem reconstruída.

2.4.3 Técnicas de codificação baseadas em modelos

As técnicas de codificação abordadas até este momento utilizam de alguma forma conceitos desenvolvidos na área da teoria da informação. De facto, os métodos de codificação preditiva, codificação com transformadas e quantificação vectorial consideram as imagens (ou sequências de imagens) como sinais aleatórios, que são comprimidos por redução das suas redundâncias.

As técnicas de codificação baseadas em modelos funcionam segundo um princípio diferente. Ao construirmos um modelo de um objecto presente na cena, podemos analisar e reconstruir as alterações desse objecto com base em transformações deste modelo. Isto permitirá uma redução da informação necessária para a codificação da sequência.

Um esquema básico de um sistema de codificação baseado em modelos é apresentado na figura 2.24.

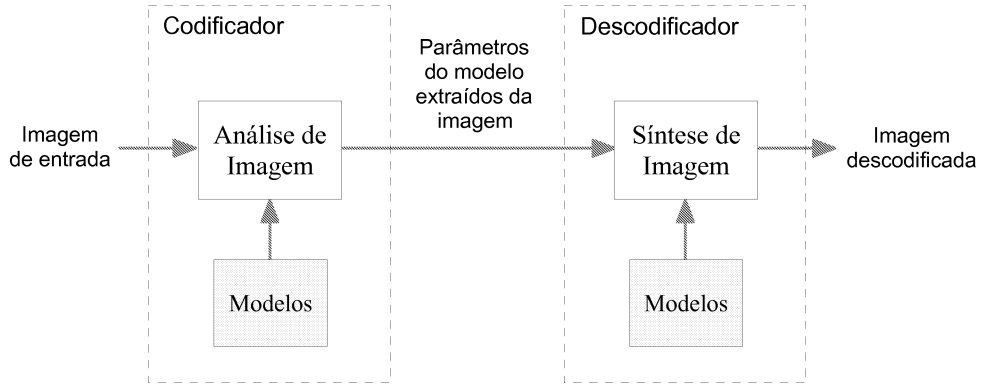
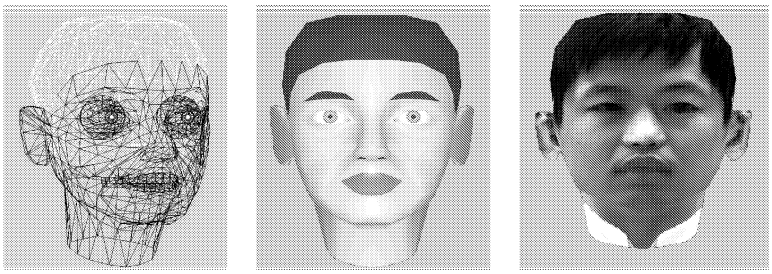


Figura 2.24: Esquema geral de um sistema de codificação baseado em modelos.

Os componentes fundamentais deste sistema de codificação são: o *modelo*, utilizado para representar objectos da cena e os processos de *análise* e *síntese* da imagem, que permitem respectivamente, a representação e a reconstrução de uma imagem com base no

modelo utilizado.

Os modelos utilizados são normalmente *superfícies 3D* ou *volumes 3D*, gerados a partir de medidas efectuadas *a priori*. Estes modelos consistem normalmente num "esqueleto" do objecto, definido por regiões planares triangulares, designado por *modelo de arame*[6]. Podem no entanto utilizar-se também representações *paramétricas* dos modelos tridimensionais, definidos neste caso por primitivas geométricas.



(a) (b) (c)

Figura 2.25: Modelo tridimensional de uma face: **a)** modelo de arame; **b)** modelo anterior preenchido com textura sintética; **c)** modelo adaptado à representação de uma face real (com textura natural) [27].

Existem dois esquemas para a codificação de sequências de imagens baseadas em modelos [19]:

- *codificação semântica*, que utiliza modelos explícitos de objectos para efectuar a análise e a síntese das imagens em movimento[7];
- *codificação orientada para objectos*, que não utiliza modelos explícitos de objectos.

Exemplos de codificação semântica são comuns na codificação de sequências vídeo do tipo *cabeça e ombros*, muito utilizadas em aplicações de videotelefonia e videoconferência [27, 28]. Neste caso, um modelo de arame de uma face humana é adaptado a cada imagem da sequência real. A codificação da sequência é feita à custa de um conjunto de parâmetros, que definem a deformação desse modelo de uma imagem para a seguinte. Este processo está ainda longe de ser trivial, pois envolve a análise da expressão facial do sujeito da imagem e a extracção dos seus pontos, que permitem a adaptação correcta do modelo.

Depois da adaptação do modelo de arame, em cada triângulo da região planar é aplicada uma textura correspondente à informação da luminância e crominância da imagem

[6]Tradução do inglês *wireframe*.

[7]Estes codificadores são por vezes designados também por *codificadores baseados no conhecimento*.

original.

Em [19] são apresentados os aspectos fundamentais da codificação de sequências do tipo *cabeça e ombros*. Para além disso, são apresentados alguns dos modelos mais conhecidos, utilizados neste tipo de aplicações.

Em [29, 30] é explicado o princípio da *codificação orientada para objectos*. Segundo esta técnica, cada imagem da sequência é dividida em objectos com movimento. Estes objectos são descritos por três tipos de parâmetros:

- parâmetros de *movimento* (nos casos apresentados, o movimento considerado é apenas de translação);
- parâmetros de *forma* (por exemplo o desenho a preto e branco do contorno do objecto);
- parâmetros de *cor* (informação da luminância e crominância dos pixels do objecto);

Estes modelos de objectos podem ser definidos como superfícies rígidas e planares, ou como objectos tridimensionais, construídos de forma a que a projecção do modelo seja idêntica à imagem real. Estes objectos são definidos utilizando um algoritmo que analisa as imagens da sequência e constrói o modelo de arame de cada objecto identificado.

Por não ser necessário um modelo prévio de cada tipo de objecto da cena, estas técnicas permitem a codificação de um grupo mais alargado de sequências. Esta exigência limita as técnicas de codificação semântica, capazes de codificar eficientemente apenas os objectos para os quais existe um modelo pré-definido.

No entanto, uma cena não pode normalmente ser descrita completamente por objectos, havendo sempre áreas da imagem não identificadas como tal. Um dos métodos mais aconselhados é a combinação de técnicas de codificação baseadas em modelos com técnicas de codificação tradicionais, capazes de processar estas áreas da imagem [19].

Em [30] são comparados os resultados obtidos por um codificador orientado para objectos e um codificador híbrido baseado em blocos, na codificação de uma sequência do tipo *cabeça e ombros*. O autor deste trabalho conclui que, apesar da maior qualidade objectiva conseguida pelo codificador híbrido, o codificador baseado em objectos consegue, para a mesma taxa de transmissão, uma maior qualidade subjectiva.

Concluiu também que os codificadores baseados em modelos são principalmente adequados para a codificação de cenas em que:

- são poucos os objectos em movimento;

- o movimento dos objectos é moderado e domina sobre o resto das alterações da cena;
- os objectos cobrem cerca de 40 a 60% da área da imagem;
- não existe movimento da câmara (a cena tem um fundo estático).

Estas características predominam em aplicações típicas de videoconferência e videotelefonia.

Com o início do trabalho na norma MPEG-4[8], o paradigma da integração de objectos sintéticos em cenas naturais, que é um dos pilares do desenvolvimento desta norma, gerou um interesse renovado na codificação baseada em modelos. Foi então criado um grupo de trabalho, designado SNHC - *Synthetic and Natural Hybrid Coding* - dedicado ao desenvolvimento deste tipo de sistemas de codificação. Do trabalho deste grupo resultou um conjunto de ferramentas destinadas à codificação de objectos sintéticos e à sua integração em cenas com outros tipos de conteúdos.

Em [27] é apresentada uma descrição das ferramentas de animação facial MPEG-4. Seguindo a filosofia das várias normas MPEG, que normalizam apenas os pontos essenciais de cada sistema, não foi definido um modelo específico de face. A ferramentas definidas permitem: configurar o modelo 3D utilizado (FDP's - *Facial Definition Parameters*); definir os parâmetros de animação desse modelo (FAP's - *Facial Animation Parameters*) e definir o modo de interpolar os parâmetros de animação necessários (FIT - *FAP Interpolation Table*).

Para além de apresentar genericamente a tecnologia de animação facial MPEG-4, em [27] é descrito um sistema de animação facial próprio (representado anteriormente na figura 2.25).

2.5 Normas de codificação vídeo

Nesta secção pretendem-se apresentar de forma breve algumas das normas de codificação de sinais vídeo digitais. Não sendo a intenção deste trabalho estudar detalhadamente todas estas normas, este capítulo introdutório não ficaria completo sem uma referência, ainda que breve, às principais normas de codificação actuais.

Neste breve resumo, serão referidas algumas aplicações típicas de cada uma das normas e as características técnicas que consideramos mais interessantes no enquadramento deste trabalho.

[8] Os aspectos principais desta norma serão apresentados na secção 2.5.3.

2.5.1 H.261

A recomendação H.261 [31], foi desenvolvida pelo CCITT - *Consultative Commitee for International Telephony and Telegraphy*[9] - no âmbito da recomendação H.320 [32] que define uma família de normas destinadas a aplicações de videotelefonia e videoconferência em tempo real, em redes RDIS (*Rede Digital com Integração de Serviços*). Por esta razão, esta norma é também designada por $p\times 64$, devido ao facto de operar com taxas de transmissão múltiplas de 64 kbps, com p entre 1 e 30.

Esta norma usa codificação intra e codificação inter utilizando como referência a imagem anterior. Em ambos os casos utiliza-se DCT aplicada sobre blocos de 8×8 elementos. Os coeficientes resultantes da aplicação desta transformada são quantificados e codificados com códigos de comprimento variável.

A norma utiliza apenas 2 formatos de imagens não interlaçadas: o formato CIF (*Commom Intermediate Format*), com 288 linhas com 352 elementos cada e o formato QCIF (*Quarter CIF*), com 144×176 elementos. O sistema de cor utilizado é o formato YUV 4:2:0 [1]. A frequência temporal usada é de 30 imagens por segundo.

A representação de uma imagem é feita segundo uma estrutura hierárquica com 3 camadas: bloco (de dimensão fixa 8×8); macrobloco (MB) e grupo de blocos (GOB - *Group Of Blocks*). Cada macrobloco é constituído por 4 blocos de luminância e os dois de crominância. Cada GOB é constituído por 3 linhas de 11 macroblocos, ou seja, uma imagem CIF tem 12 GOB's, e uma imagem QCIF 3 GOB's.

Os processos de estimação e compensação de movimento utilizam blocos de luminância com 16×16 elementos, ou seja um macrobloco, sendo no entanto opcionais. A norma não define qualquer método de estimação de movimento, restringindo, no entanto, a janela de pesquisa utilizada a uma dimensão máxima de 15×15 pixels.

Cada vector de movimento é determinado utilizando uma precisão de um pixel. O vector de movimento é único para cada macrobloco, sendo utilizado um vector cujas componentes têm metade do tamanho original para a compensação de movimento dos blocos de crominância.

[9] Actualmente designado por ITU-T, sector de normas de comunicações da *International Telecommunications Union*

2.5.2 H.263

A norma H.263 [33] foi desenvolvida pela ITU-T, a partir da experiência adquirida no H.261 e na norma MPEG-2. Esta norma destina-se ao mesmo tipo de aplicações do H.261, mas para taxas de transmissão ainda mais baixas, cobrindo aplicações de videotelefonia na *Internet*, em redes de dados locais ou de área alargada (*LAN's* e *WAN's*), ou mesmo na rede telefónica comutada.

Sendo uma evolução do H.261, a norma H.263 partilha com este um conjunto de características tais como a utilização de vídeo não interlaçado e o formato de cor YUV 4:2:0. No entanto, esta nova norma permite a utilização de três novos formatos de imagens, resumida na tabela 2.3.

Formato	**Nº linhas**	**Pixels por linha**	**H.261**	**H.263**
sub-QCIF	96	128		Obrigatório
QCIF	144	176	Obrigatório	Obrigatório
CIF	288	352	Opcional	Opcional
4CIF	576	704		Opcional
16CIF	1152	1408		Opcional

Tabela 2.3: Formatos de imagem suportados pelas normas H.261 e H.263.

Tal como no caso do H.261, o controlo do débito gerado pelo codificador é feito principalmente à custa da variação do passo de quantificação dos coeficientes da DCT.

Relativamente aos processos de estimação e compensação de movimento, mantém-se a utilização de vectores de movimento de translação, sendo agora possível utilizar uma precisão de meio-pixel. Outra característica importante é a utilização de estimação e compensação bidireccional de movimento.

2.5.3 As normas MPEG

O grupo MPEG (*Moving Pictures Expert Group*), originalmente conhecido como ISO-IEC/JTC1 SC29/WG11, foi criado pela ISO (*International Organization for Standardisation*) em 1988, para proceder ao desenvolvimento de uma norma que uniformizasse a codificação de sinais de vídeo e áudio, para aplicações multimédia.

O primeiro resultado do trabalho deste grupo ficou concluído em 1991, com a publicação da norma ISO/IEC 11172, designada por *Coding of moving pictures and associated audio for digital storage media at up to 1.5 Mbit/s*. Esta norma ficou conhecida como

MPEG-1, e foi desenvolvida para a codificação de sinais audiovisuais digitais com débitos até aos 1.5 Mbps.

A norma *MPEG-2*, também conhecida por ISO/IEC 13818 ou ITU-T H.262, desenvolveu as capacidades da norma *MPEG-1*, de modo a estender o seu domínio de aplicação, que vai desde o armazenamento digital até à difusão de televisão digital de alta definição.

A mais recente norma do grupo, denominada *MPEG-4* foi finalizada recentemente, tendo sido definida de modo a suportar um grande número de aplicações para transmissão, acesso e manipulação de sinais audiovisuais digitais. Uma das inovações principais desta norma é a introdução de interactividade e controlo de acesso aos conteúdos por parte do utilizador, para além de proporcionar elevadas taxas de compressão.

Todos as normas MPEG têm em comum o facto de especificarem apenas a estrutura dos dados utilizados na codificação e o processo de descodificação, definidos independentemente da aplicação alvo. Este facto, faz com que cada fabricante seja livre de tomar as opções de implementação que entender, no sentido de melhorar o desempenho do sistema.

Todos estas normas têm também em comum o facto da sua especificacação conter quatro partes fundamentais:

- 1ª Parte - Sistema: define o modo de integrar num só fluxo de bits vários sinais de vídeo, áudio e controlo;
- 2ª Parte - Vídeo: define as técnicas aplicáveis à codificação de sequências de imagens;
- 3ª Parte - Áudio: define as técnicas aplicáveis à codificação de sinais áudio e voz;
- 4ª Parte - Testes de conformidade: define um conjunto de testes de verificação do sistema.

Seguidamente serão abordados brevemente os aspectos principais dos sistemas de codificação de vídeo da cada uma das normas referidas.

A norma MPEG-1

A norma MPEG-1 [34, 10] foi desenvolvida para codificação de TV e áudio, de forma a permitir a implementação das funcionalidades normais de um videogravador: reprodução, acesso aleatório, avanço e recuo rápidos, entre outros.

Estes requisitos condicionam os métodos utilizados na codificação. O sistema mais adequado para permitir o acesso aleatório a imagens de uma sequência seria a utilização

apenas de codificação intra. No entanto, um sistema deste tipo seria pouco eficite, porque não explora a redundância temporal da sequência.

Esta norma define três tipos de codificação para as imagens da sequência: codificação intra (imagens I), codificação inter com predição causal (imagens P) e imagens B, que utilizam estimação e compensação bidireccional de movimento, como explicado na secção 2.4.2. As imagens B apresentam a maior eficiência de codificação, não sendo, no entanto, utilizadas como referência nos processos de estimação de movimento. A figura 2.26 representa exemplos de imagens dos três tipos e as relações de predição temporal existentes entre elas.

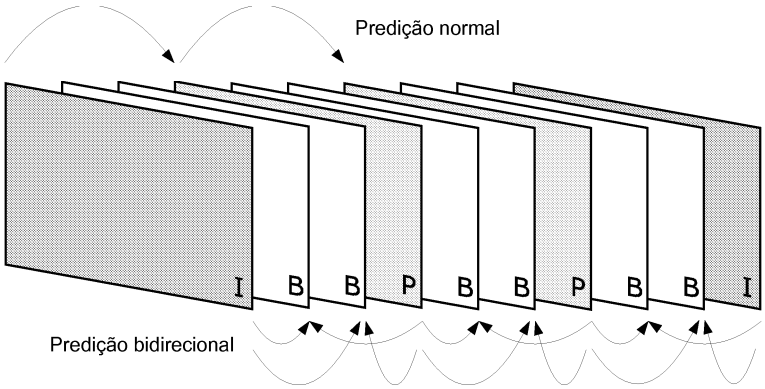


Figura 2.26: Exemplo dos tipos de imagem definidos na norma MPEG-1 e relações de predição temporal.

Como já referimos, a utilização de imagens do tipo B implica um reordenamento na transmissão das imagens da sequência. A norma define um quarto tipo de imagens, raramente utilizadas, denominadas imagens D. Estas imagens correspodem a uma representação de baixa resolução e não podem ser utilizadas simultaneamente com imagens dos outros tipos.

As imagens utilizadas podem ter uma máximo de 4095×4095 elementos. Cada imagem é dividida em macroblocos, numerados da esquerda para a direita e de cima para baixo. A estrutura de cada macrobloco é idêntica à do H.261, mantendo-se também o formato de cor YUV 4:2:0.

Para cada macrobloco da imagem é realizada a estimação de movimento, de acordo com o tipo de imagem a ser codificada. A cada macrobloco de uma imagem P é associado um vector de movimento, sendo utilizados dois vectores de movimento para cada bloco de uma imagem do tipo B. Os vectores de movimento utilizados têm uma precisão máxima de meio pixel.

As imagens de erro e as imagens intra são codificadas com DCT utilizando blocos de 8×8 elementos. Os coeficientes da tranformada são quantificados e serializados utilizando um varrimento em zig-zag. O valor e posição dos coeficientes não nulos são codificados entropicamente.

Uma diferença importante em relação à norma H.261 é a utilização de uma matriz de passos de quantificação, definida em função do tipo de bloco a codificar e da importância perceptiva de cada coeficiente.

A norma MPEG-2

A norma MPEG-2 [35, 10, 36] foi desenvolvida tendo em vista estender a gama de aplicações do MPEG-1, nomeadamente permitindo a utlização de débitos binários muito superiores, por exemplo 80 ou 100Mbps, necessários na codificação de HDTV (*High-definition TV* ou televisão de alta definição).

Para um conjunto tão grande de aplicações suportadas, não era possível definir uma só especificação adequada a todas elas. Por outro lado, para uma norma definida de um modo abrangente, a maioria das aplicações não utilizaria todas as características nela definidas. Deste modo, a norma MPEG-2 foi definida como um conjunto de ferramentas que podem ou não ser utilizadas em função de cada aplicação particular.

Para além de manter a compatibilidade com a norma MPEG-1, o MPEG-2 introduz novos níveis de representação hierárquica, nomeadamente o conceito de escalamento dos dados e mecanismos que aumentam a imunidade aos erros de transmissão.

Escalamento dos dados possibilita a utilização de apenas parte da informação, na reconstrução de uma versão de menor qualidade da sequência original. Um descodificador MEPG-2 pode neste caso, descodificar apenas a camada base do sinal ou utilizar as camadas superiores para aumentar a qualidade do sinal reconstruído.

A necessidade de uma maior imunidade aos erros deve-se ao facto de as aplicações MPEG-2 não se restringirem principalmente ao armazenamento e reprodução de sequências a partir de suporte digital, mas incluirem transmissão de sinais de TV e videotelefonia sobre um grande número de meios de transmissão, desde as redes de dados à comunicação por satélite.

A base da norma MPEG-2 segue um esquema de codificação semelhante ao utilizado pela norma MPEG-1, baseada em codificação preditiva, estimação de movimento causal ou bidireccional, codificação de imagem com DCT e utilização de códigos de comprimento

variável. No entanto existem algumas diferenças:

- a definição de um conjunto de perfis e níveis de utilização;
- o suporte de escalamento dos dados;
- o suporte de vídeo interlaçado bem como novos esquemas de estimação e compensação de movimento apropriados.

Os formatos suportados pela norma foram divididos em vários perfis com quatro níveis de utilização. Um perfil é definido de modo a utilizar um conjunto particular de algoritmos definidos na norma, o que limita a sintaxe dos dados.

Para um perfil, cada nível define a gama de valores de um conjunto de parâmetros: como sejam a resolução das imagens utilizadas, a frequência e o débito máximo. A tabela 2.4 representa um resumo de alguns dos valores destes parâmetros para cada um dos quatro níveis definidos na norma.

Nível	**Valores máximos**
High	Imagens com 1152×1920 elementos; 60 imagens/s; 80Mbps
High 1440	Imagens com 1152×1440 elementos; 60 imagens/s; 60Mbps
Main	Imagens com 576×720 elementos; 30 imagens/s; 15Mbps
Low	Imagens com 288×352 elementos; 30 imagens/s; 4Mbps

Tabela 2.4: Valores máximos dos parâmetros para os níveis MPEG-2.

Os perfis *Simple* e *Main* não são escaláveis, enquanto que os restantes perfis permitem o escalamento dos dados gerados. O perfil *Main* contém as funcionalidades do MPEG-1, permitindo ainda a codificação de vídeo interlaçado. O perfil *Simple* pretende ser uma versão reduzida do perfil anterior, que exclui, por exemplo, a estimação bidireccional de movimento. A tabela 2.5 representa um resumo das principais características dos cinco perfis definidos.

O primeiro, *SNR Scalable* permite apenas escalamento ao nível da relação sinal ruído (*SNR*), para dois níveis de qualidade, mantendo a mesma resolução e formato de imagem.

Perfil	**Características**
Simple	Inclui todas as funcionalidades do perfil *Main*, excepto o suporte de predição bidireccional; representação YUV 4:2:0
Main	Funcionalidades básicas da norma MPEG-1, com suporte para vídeo interlaçado; representação YUV 4:2:0
SNR Scalable	Todas as funcionalidades do perfil *Main*; define duas camadas de dados com escalamento de SNR; representação YUV 4:2:0
Spacial Scalable	Todas as funcionalidades do perfil *SNR Scalable*; suporta uma nova camada com escalamento de resolução espacial; representação YUV 4:2:0
High	Todas as funcionalidades do perfil *Spacial Scalable*; suporta três camadas com escalamento de SNR e de resolução espacial; representação YUV 4:2:2

Tabela 2.5: Características típicas dos vários perfis MPEG-2.

O perfil *Spacial Scalable* utiliza três camadas de dados, que permitem um aumento da resolução espacial, para além do escalamento da SNR já descrita. Finalmente, o perfil *High* é equivalente ao perfil anterior, mas suporta a codificação de imagens com um formato YUV 4:2:2.

O escalamento dos dados permite que descodificadores com diferentes capacidades sejam capazes de representar versões de qualidade diferente, a partir do mesmo sinal codificado. Esta propriedade é importante, nomeadamente, em aplicações de difusão e de pesquisa em bases de dados multimédia. Além disso, podemos definir camadas que contêm dados com diferentes prioridades, o que pode ser útil na transmissão em redes de dados.

A utilização de níveis e perfis permite a especificação, num só documento, de técnicas e parâmetros correspondentes a um grande número de aplicações. É importante realçar que nem todos os níveis estão definidos para os vários perfis de funcionamento. Por outro lado, os valores dos parâmetros de cada nível variam com o perfil utilizado. Uma discussão profunda deste assunto pode ser encontrada em [36], para além do texto da norma [35].

Como foi referido anteriormente, uma das novas características do MPEG-2 é o suporte de vídeo interlaçado, conforme se representa na figura 2.27.

Para este formato de vídeo foram definidos dois tipos de predição: predição a nível de imagem e a predição a nível de campo. No primeiro caso, constrói-se uma imagem completa

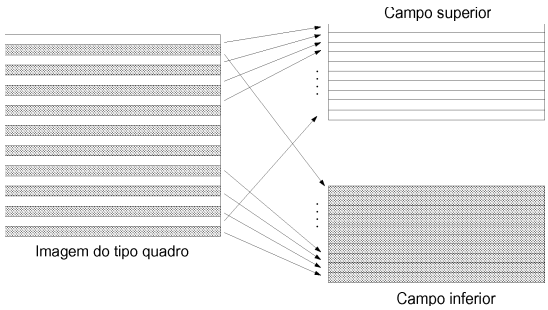


Figura 2.27: Relação entre uma imagem de vídeo interlaçado e os seus campos.

a partir dos seus campos, sendo a estimação de movimento feita do modo habitual.

No segundo caso, consideram-se os dois campos de cada imagem separadamente. Para cada campo, a estimação de movimento pode utilizar como referência o outro campo da mesma imagem, ou um campo de uma outra imagem da sequência. A predição de campo é eficaz para a codificação de sequências de vídeo interlaçado, não sendo necessária para sequências de vídeo normais.

A norma MPEG-4

A mais recente norma do grupo MPEG, denominada MPEG-4 [37], representa a introdução de uma nova filosofia nos sistemas de codificação de sinais audiovisuais. Com a crescente convergência e integração de serviços nas áreas das telecomunicações, dos computadores e das empresas de televisão e entretenimento, o novo standard pretende viabilizar essa convergência [38].

Três grandes ideias emergem nesta norma: o aumento da importância dos conteúdos audiovisuais em todos os tipos de redes de comunicações, o aumento da mobilidade e o aumento da interactividade. Mais uma vez estes requisitos condicionam as necessidades em termos dos algoritmos desenvolvidos. No caso do MPEG-4, optou-se por um esquema de codificação baseada em *objectos audiovisuais* independentes, cujas relações no espaço e no tempo compõem a cena codificada. Isto representa uma inovação em relação às normas anteriores, que procuravam unicamente a eficácia da compressão, ignorando o seu conteúdo.

A figura 2.28 representa um diagrama de blocos de um sistema MPEG-4. Diferentes tipos de objectos podem ser utilizados. Para além dos sinais audiovisuais clássicos, a norma suporta vários tipos de objectos sintéticos, de som e imagem, para além de objectos de texto. Cada um destes diferentes tipos de objectos, é codificado de um modo

independente, adequado a cada caso.

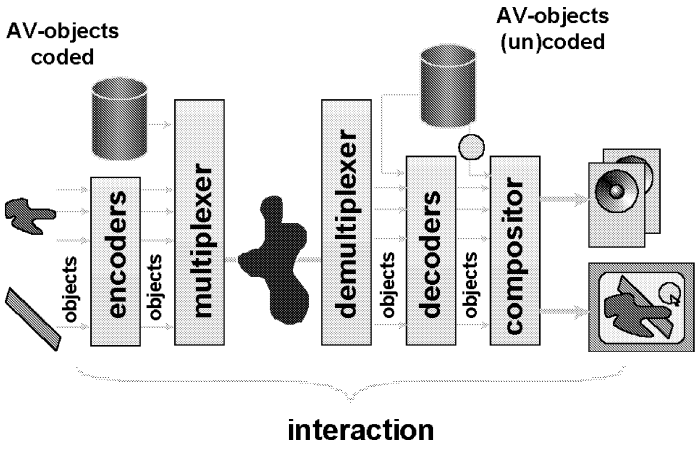


Figura 2.28: Esquema simplificado de um sistema MPEG-4 [39].

O esquema apresentado permite a composição dos diferentes tipos de objectos numa só cena, realizada num bloco final dedicado a esta tarefa. Isto permite uma interacção do utilizador com os vários objectos codificados, dando-lhe possibilidade de controlar e manipular os conteúdos representados. Por outro lado, o utilizador pode criar ele próprio os seus próprios conteúdos, pela associação de vários objectos armazenados [39].

Para além disso, a divisão da cena em vários objectos permite um escalamento natural, por exemplo ao nível da qualidade/resolução pretendida para cada componente específico da cena.

O aumento da mobilidade é uma consequência do desenvolvimento das redes móveis e dos respectivos equipamentos terminais. Este facto permitiu a introdução de serviços de *internet* e aplicações audiovisuais nestas redes.

Esta possibilidade foi considerada pelos elementos do grupo MPEG, que desde 1995 previram a necessidade do funcionamento de certas aplicações MPEG-4 sobre redes móveis. As características particulares deste tipo de redes, nomeadamente os baixos débitos e as elevadas taxas de erro que apresentam, levaram à introdução de métodos de detecção e correcção de erros muito mais elaborados que os utilizados nas normas anteriores.

Tal como já acontecia com as restantes normas MPEG, o MPEG-4 é definido como um conjunto de ferramentas, utilizadas de acordo com as características de cada aplicação. A definição de um conjunto de perfis e níveis define uma estrutura hierárquica de representação da informação, que garante a compatibilidade de diferentes aplicações.

Entre as novas ferramentas para a codificação de objectos visuais, destacam-se a uti-

lização de modelos para a codificação de faces, a utilização de grelhas 2D triangulares e a utilização de grelhas 3D genéricas. Juntam-se a estas um conjunto de ferramentas de codificação de sinais vídeo (compatíveis com as normas anteriores), que permitem também a codificação eficaz de objectos visuais com formas arbitrárias. Uma descrição genérica destas ferramentas é apresentada em [40].

A norma MPEG-7

A norma MPEG-7 representa a mais recente evolução na área da codificação de sinais audiovisuais, por parte do grupo MPEG. De modo a facilitar o acesso aos conteúdos audiovisuais pretendidos pelo utilizador, a norma MPEG-7 pretende desenvolver métodos de descrição dos vários tipos de informação audiovisual, incluindo imagens estáticas, vídeo, voz, áudio natural e sintético, gráficos e modelos 3D, independentemente do seu formato (analógico ou digital), e do seu meio de armazenagem (papel, filme, suportes magnéticos, etc.) [41].

Ao contrário das normas anteriores, orientados para a reprodução de conteúdos, a nova norma destina-se fundamentalmente à identificação e descrição desses conteúdos. Esta identificação deve ser feita independentemente do modo como se encontra armazenada a informação e do tipo de codificação utilizado (MPEG-X ou outros).

O MPEG-7 manterá as capacidades de codificação por objectos, conjuntamente com as técnicas de codificação convencionais. Este modelo permitirá a identificação e pesquisa de objectos específicos, entre os conteúdos utilizados, bem como a sua manipulação independente e a utilização de diferentes graus de detalhe na sua descrição.

Tal como nas normas anteriores do grupo MPEG, o MPEG-7 não será definido para nenhuma aplicação específica, o que implica a normalização apenas da descrição dos objectos audiovisuais, não sendo definidos os processos de extracção, codificação e pesquisa desses objectos. Uma apresentação genérica deste standard pode ser encontrada em [41]. Em [42] é feita uma descrição das tecnologias envolvidas nesta norma e das suas aplicações típicas.

2.5.4 Outras fontes de informação

Apesar de imperativa, a apresentação das várias normas de codificação de sinais vídeo, realizada nesta secção, não podia deixar de ser breve pois ultrapassa o âmbito central deste trabalho. Procurou-se, no entanto, dar uma ideia das características principais de

cada uma das normas, relativamente aos seus objectivos e métodos.

A norma *JPEG*, para a codificação de imagens estáticas, foi a fonte de muitas das soluções utilizadas pelas normas aqui apresentadas. Uma descrição detalhada desta norma pode ser encontrada em [8].

As publicações que se referem às normas 1 e 2 do grupo MPEG são abundantes. Entre elas podemos salientar os livros [10] e [36] que apresentam descrições detalhadas destas normas.

Para além disso, vasta documentação referente às actividades do grupo MPEG, e a todas as normas do grupo, podem ser encontrados na página oficial do grupo MPEG, no endereço http://www.cselt.it/mpeg/. A página http://www.mpeg.org/, *"MPEG Pointers and Resources"*, contém uma lista de ponteiros para recursos MPEG, incluindo companhias envolvidas no seu desenvolvimento, produtos comerciais disponíveis, sequências de teste codificadas e *software* relacionado, para além de informação diversa referente às normas MPEG-1 e MPEG-2.

Capítulo 3

Estimação e Compensação de Movimento utilizando Transformações Geométricas

Como vimos no capítulo anterior, os métodos de estimação e compensação de movimento permitem melhorar a eficiência da codificação de sequências de imagens. As técnicas de compensação de movimento mais utilizadas baseiam-se na correspondência entre blocos, associados a vectores de movimento de translação. Estas técnicas, designam-se genericamente por *BMA - Block Matching Algorithms.*

Estas técnicas realizam a estimação e compensação de movimento assumindo que [43]:

- os movimentos dos objectos são de translação;
- as condições de iluminação são constantes no espaço e no tempo;
- as situações de encobrimento entre os objectos e o aparecimento de novas áreas do fundo da cena podem ser desprezadas;

Estes presupostos não se verificam na grande maioria das situações, o que torna este modelo desadequado, por exemplo, para a representação de movimentos de rotação e em situações de *zoom.* A utilização de transformações geométricas na estimação e compensação do movimento permite a representação de um número mais elevado de alterações entre as imagens da sequência.

As técnicas *BMA* utilizam o vector de movimento que minimiza o erro de predição entre os blocos. Na compensação de movimento com transformações espaciais, utilizamos

a *transformação geométrica* que minimiza o valor do erro entre o bloco a estimar e o bloco de referência transformado [19]. Como nas técnicas convencionais, o critério de erro utilizado é normalmente o *erro médio absoluto* ou o *erro médio quadrático.*

Seguidamente apresentamos as transformações geométricas mais utilizadas para a estimação e compensação de movimento. As transformações *afins* e *perspectivas* permitem uma aplicação planar de um bloco original num bloco transformado arbitrário, respectivamente para blocos triangulares ou quadrangulares. Serão também discutidas as transformações *bilineares*, que permitem correspondências mais complexas de quadriláteros em quadriláteros.

A aplicação destas transformações no processo de compensação de movimento será também discutido. Será dada particular relevância ao método descrito em [26, 44], designado por *BMGT* (*Block Matching with Geometrical Transform*). Este método será utilizado como referência para os métodos tradicionais de compensação de movimento com transformações geométricas.

No final do capítulo será feita uma breve descrição de alguns métodos de codificação de sequências que utilizam transformações geométricas para a estimação e compensação de movimento.

3.1 Transformações geométricas

Uma *transformação espacial* define uma aplicação de um sistema de coordenadas noutro, estabelecendo uma correspondência entre os pontos das imagens de partida e de chegada. A função genérica de transformação definida na forma directa, relaciona o sistema de coordenadas de saída com o sistema de coordenadas de entrada, ou vice-versa, na sua forma inversa.

Considerando o vector de coordenadas $[u, v]$, do sistema de entrada e o vector de coordenadas $[x, y]$, do sistema de saída, as formas directa e inversa de uma transformação espacial podem ser dadas pelas seguintes expressões:

$$\begin{aligned} [x, y] &= [X(u, v), Y(u, v)] \\ [u, v] &= [U(x, y), V(x, y)]. \end{aligned} \tag{3.1}$$

Nestas expressões, X, Y, U e V são funções arbitrárias que definem as transformações espaciais das coordenadas.

X e Y, por converterem as coordenadas de entrada nas coordenadas de saída, definem

a transformação *directa*. Do mesmo modo, as funções U e V definem a transformação *inversa*.

Sendo X e Y funções reais, elas transformam um conjunto de pixels, dispostos numa grelha discreta com coordenadas inteiras e uniformemente espaçadas, num espaço contínuo de coordenadas reais. No entanto, pretendemos uma imagem de saída representada por um conjunto de pixels dispostos numa grelha discreta.

A utilização de transformações reais ponto-a-ponto pode originar dois fenómenos indesejáveis: a existência de "buracos" na grelha da imagem de saída, aos quais não corresponde nenhum pixel da imagem de entrada; ou a existência de sobreposição de dois ou mais pixels da imagem de entrada num só elemento da grelha da imagem de saída. Para evitar estes fenómenos, blocos quadrangulares são normalmente transformados pela aplicação dos seus quatro cantos, como representamos na figura 3.1.

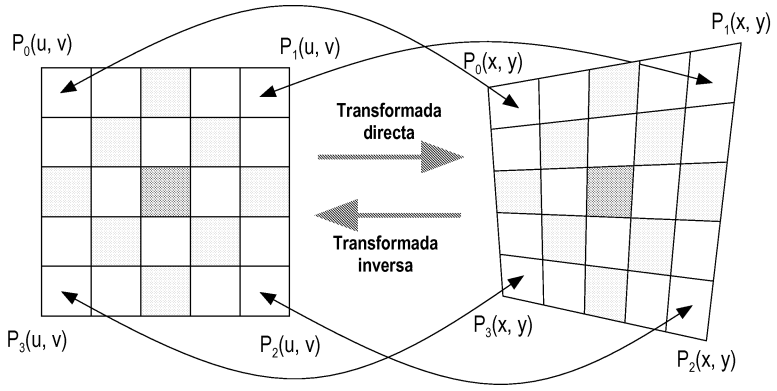


Figura 3.1: Transformações geométricas directa e inversa de um quadrilátero.

Após a transformação do bloco original, procede-se a uma re-amostragem nos pontos da grelha da imagem de saída. Esta nova amostragem utiliza uma interpolação que determina o valor a atribuir a cada elemento da grelha de saída, a partir dos valores dos pontos transformados. Este processo é extremamente importante na aplicação de transformações geométricas a imagens e será descrito com maior detalhe oportunamente neste capítulo.

Ao contrário do que acontece com a transformação directa, a transformação inversa utiliza uma imagem de saída completa, podendo dar origem a uma imagem original com buracos e sobreposições. Neste caso, a interpolação é realizada no espaço de entrada e não no espaço de saída, o que torna este tipo de solução mais vantajosa para algumas aplicações.

3.1.1 Transformações geométricas mais comuns

As funções genéricas de transformação, definidas nas equações (3.1), podem tomar qualquer forma arbitrária. As mais simples, como as transformações *afins* e *perspectivas*, utilizam uma matriz de transformação homogénea genérica. Transformações mais complexas podem ser conseguidas utilizando transformações *bilineares* ou *polinomiais*. Outras transformações, que não podem ser descritas por funções analíticas, são definidas utilizando matrizes esparsas de pontos de controlo, para os quais a correspondência é conhecida, e interpolando os restantes pontos.

Neste trabalho, as transformações geométricas serão utilizadas para a estimação e compensação de movimento, o que implica a transmissão dos parâmetros que as definem. Para cada tipo de transformação utilizada, há que considerar um compromisso entre a sua complexidade e os resultados obtidos na correspondência entre blocos.

Existem exemplos de aplicações de diferentes tipos de transformações geométricas para a estimação e compensação de movimento. De entre elas, as mais usadas são as transformações *afins*, *perspectivas* e *bilineares*.

As transformações afins e perspectivas podem ser definidas através de uma matriz de transformação genérica, T_1, 3×3, que opera sobre dois *sistemas de coordenadas homogéneas*. Um vector $[u, v]$ com coordenadas bidimensionais, é representado num sistema de coordenadas homogéneas com mais um elemento, w', designado por coordenada homogénea, tomando a forma $[u', v', w']$. Genericamente, a utilização de coordenadas homogéneas adiciona um elemento aos vectores de um sistema de coordenadas, verificando-se as relações: $u = u'/w'$ e $v = v'/w'$.

Isto implica que a representação de um ponto em coordenadas homogéneas não é única. No entanto, as coordenadas homogéneas $[u, v, 1]$ representam sempre um único ponto $[u, v]$ do sistema bidimensional, pelo que o factor de escala w' pode ser ignorado sem perda de generalidade, se estivermos apenas interessados na transformação bidimensional.

Deste modo, a representação genérica de uma transformação definida pela matriz T_1 tem a forma:

$$[x, y, 1] = [u, v, 1] \times T_1 = [u, v, 1] \times \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}. \tag{3.2}$$

Uma transformação definida desta forma é capaz de executar translações, rotações, mudanças de escala, reflexões, inclinações segundo um dos eixos e mudanças de perspec-

tiva de sistemas de coordenadas bidimensionais. A forma como isso é conseguido pode ser facilmente compreendida se dividirmos a matriz T_1 em quatro submatrizes, como se apresenta de seguida:

$$T_1 = \begin{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} & \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix} \\ \begin{bmatrix} a_{31} & a_{32} \end{bmatrix} & a_{33} \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & a_{33} \end{bmatrix}. \tag{3.3}$$

A matrix T_{11} define uma transformação linear que executa rotações, mudanças de escala e alterações da inclinação. A matrix T_{12} produz uma transformação perspectiva e a matrix T_{21} realiza uma translação, enquanto que o elemento a_{33} representa um factor de escala global. Estas transformações e a sua combinação na tranformação genérica conseguida pela matriz T_1 são explicadas em [45].

Transformações afins

Uma transformação *afim* pode ser definida por uma matriz com a forma:

$$[x, y, 1] = [u, v, 1] \times \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & 1 \end{bmatrix}. \tag{3.4}$$

Nesta matriz, o elemento a_{33} é unitário (o que evita a operação de escala da coordenada homogénea) e a última coluna é igual a $[0\ 0\ 1]^T$. Isto resulta numa transformação *ortográfica*, ou seja, que preserva os pontos equidistantes da imagem de entrada (embora as distâncias correspondentes na imagem de saída possam ser diferentes). Do mesmo modo, as transformações afins preservam o paralelismo entre linhas, o que restringe as transformações entre quadriláteros.

Genericamente, uma transformação afim transforma um triângulo de entrada num qualquer triângulo desejado, mas não tem capacidade para transformar um quadrilátero noutro quadrilátero arbitrário.

Da formulação da transformação com a matriz T_1, definida em (3.4), tiram-se as funções de transformação directa:

$$\begin{aligned} x &= a_{11}u + a_{21}v + a_{31} \\ y &= a_{12}u + a_{22}v + a_{32}. \end{aligned} \tag{3.5}$$

Os seis parâmetros necessários para definir uma transformação afim podem ser determinados utilizando três pontos, cuja transformação é conhecida, o que permite a definição do sistema de seis equações a seis incógnitas.

Transformações perspectivas

Genericamente, uma transformação perspectiva tem a forma:

$$[x', y', w'] = [u, v, 1] \times \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \tag{3.6}$$

com $x = x'/w'$ e $y = y'/w'$ e os coeficientes a_{13} e a_{23} diferentes de zero. Deste modo, as funções de transformação perspectivas são dadas por:

$$\begin{aligned} x &= \frac{x'}{w'} = \frac{a_{11}u + a_{21}v + a_{31}}{a_{13}u + a_{23}v + a_{33}} \\ y &= \frac{y'}{w'} = \frac{a_{12}u + a_{22}v + a_{32}}{a_{13}u + a_{23}v + a_{33}}. \end{aligned} \tag{3.7}$$

Nestas equações é fácil observar que as transformações afins são um caso particular das transformações perspectivas, em que $a_{13} = a_{23} = 0$, ou seja, onde a coordenada homogénea, w', é constante para todos os pontos. Pelo contrário, numa transformação perspectiva, w' varia em função de cada ponto.

Isto faz com que esta transformação preserve o paralelismo entre duas linhas apenas quando estas são paralelas ao plano de projecção. Todas as outras linhas transformadas convergem para um ponto de fuga. As transformações perspectivas mantêm, no entanto, a propriedade de preservar as linhas segundo todas as orientações.

Tendo oito graus de liberdade (a matriz de transformação pode ser normalizada de modo a que $a_{33} = 1$, sem perda de generalidade) uma transformação perspectiva permite definir transformações planares de quadriláteros em quadriláteros, ao contrário do que acontecia com as transformações afins (que com seis parâmetros definem apenas transformações de triângulos).

Em [45] é apresentado um estudo da aplicação destas transformações em aplicações planares genéricas de um quadrado num quadrilátero, de um quadrilátero num quadrado e de um quadrilátero num quadrilátero.

Transformações bilineares

As transformações bilineares têm a propriedade de transformar quadriláteros em quadriláteros não planares. Genericamente, uma transformação bilinear tem a forma:

$$[x, y] = [uv, u, v, 1] \times \begin{bmatrix} a_3 & b_3 \\ a_2 & b_2 \\ a_1 & b_1 \\ a_0 & b_0 \end{bmatrix}. \tag{3.8}$$

As funções que definem a transformação directa são dadas por:

$$\begin{aligned} x &= a_3uv + a_2u + a_1v + a_0 \\ y &= b_3uv + b_2u + b_1v + b_0. \end{aligned} \tag{3.9}$$

Uma transformação bilinear preserva as linhas verticais e horizontais da imagem original. Devido a esta propriedade, pontos equidistantes segundo estas duas direcções permanecem equidistantes. Pelo contrário, as linhas da imagem original não orientadas segundo uma destas direcções, não são preservadas, dando origem a curvas quadráticas na imagem de saída.

Dada uma aplicação genérica de quatro pontos de entrada, (u_0, v_0), (u_1, v_1), (u_2, v_2) e (u_3, v_3) em quatro pontos de saída, (x_0, y_0), (x_1, y_1), (x_2, y_2) e (x_3, y_3), os parâmetros a_i da transformação podem ser determinados pela resolução do sistema:

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 & u_0 & v_0 & u_0v_0 \\ 1 & u_1 & v_1 & u_1v_1 \\ 1 & u_2 & v_2 & u_2v_2 \\ 1 & u_3 & v_3 & u_3v_3 \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}. \tag{3.10}$$

Os parâmetros b_i podem ser determinados de forma análoga, pela expansão da equação (3.8) para o conjuntos de quatro pontos transformados. As coordenadas do resultado da transformação de um ponto genérico podem assim ser determinadas pelas equações (3.9).

Consideremos o caso representado na figura 3.2, que ilustra a transformação bilinear de um bloco quadrado num quadrilátero genérico, cujas coordenadas dos quatro vértices são conhecidas.

Se substituirmos na equação (3.10) o valor das coordenadas dos pontos de entrada e de saída, dadas por:

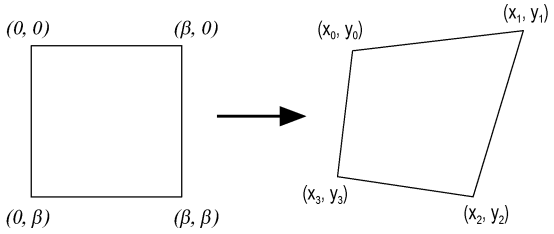


Figura 3.2: Transformação de um quadrado num quadrilátero genérico.

$$
\begin{aligned}
(0,0) &\rightarrow (x_0, y_0) \\
(\beta,0) &\rightarrow (x_1, y_1) \\
(\beta,\beta) &\rightarrow (x_2, y_2) \\
(0,\beta) &\rightarrow (x_3, y_3).
\end{aligned}
$$

obtêm-se as equações dos vários valores da função $X(u,v)$ para os quatro pontos considerados:

$$
\begin{aligned}
x_0 &= a_0 \\
x_1 &= a_0 + \beta a_1 \\
x_2 &= a_0 + \beta a_1 + \beta a_2 + \beta^2 a_3 \\
x_3 &= a_0 + \beta a_2.
\end{aligned}
\tag{3.11}
$$

Resolvendo as equações (3.11) em ordem aos parâmetros a_i, obtêm-se as seguintes expressões analíticas:

$$
\begin{aligned}
a_0 &= x_0 \\
a_1 &= \frac{x_1 - x_0}{\beta} \\
a_2 &= \frac{x_3 - x_0}{\beta} \\
a_3 &= \frac{x_2 + x_0 - x_1 - x_3}{\beta^2}.
\end{aligned}
\tag{3.12}
$$

Se determinarmos de forma análoga os valores dos parâmetros b_i, e substituirmos estes valores nas equações (3.9), obtemos directamente as equações das funções da transformação bilinear:

$$
\begin{aligned}
X(u',v') &= x_0 + (\frac{x_1 - x_0}{\beta})u' + (\frac{x_3 - x_0}{\beta})v' + (\frac{x_2 + x_0 - x_1 - x_3}{\beta^2})u'v' \\
Y(u',v') &= y_0 + (\frac{y_1 - y_0}{\beta})u' + (\frac{y_3 - y_0}{\beta})v' + (\frac{y_2 + y_0 - y_1 - y_3}{\beta^2})u'v'.
\end{aligned}
\tag{3.13}
$$

Estas funções transformam um quadrado, definido pelas coordenas normalizadas u' e $v' \in [0, \beta]$, num quadrilátero arbitrário, definido pelas coordenadas dos seus quatro vértices (x_i, y_i).

Como foi já referido, estas novas coordenadas terão valores reais, que podem não coincidir com a grelha de coordenadas inteiras. Neste caso, torna-se necessário calcular os valores dos pontos da grelha usando interpolação.

3.1.2 Interpolação bilinear

Com o processo de interpolação pretendemos determinar o valor de um ponto de uma grelha, dados os valores do pontos resultantes de uma transformação geométrica, dispostos numa grelha que não coincide com a primeira.

A abordagem mais simples de interpolação é a utilização do valor do *vizinho mais próximo*, para aproximar o valor pretendido. Outras técnicas utilizam vários valores conhecidos na vizinhança do ponto a determinar para cálcular o valor interpolado.

Uma delas é a *interpolação bilinear* que determina o valor a atribuir ao ponto P através de uma soma pesada dos valores dos seus quatro vizinhos mais próximos. Considere-se a figura 3.3, que representa o ponto a interpolar e os seus quatro vizinhos mais próximos, com valores P_a, P_b, P_c e P_d.

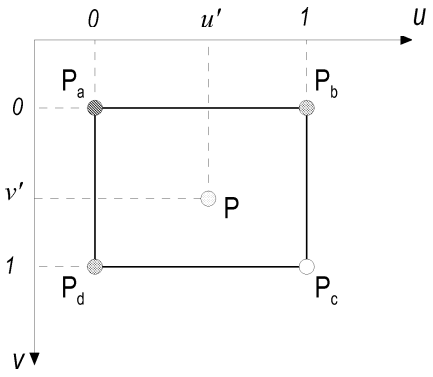


Figura 3.3: Interpolação bilinear.

O valor a atribuir ao ponto P usando interpolação bilinear é dado pela equação:

$$\begin{aligned} P &= (1-u')(1-v')P_a + u'(1-v')P_b + u'v'P_c + (1-u')v'P_d \\ &= (P_b - P_a)u' + (P_d - P_a)v' + (P_a - P_b + P_c - P_d)u'v' + P_a, \end{aligned} \tag{3.14}$$

onde u' e v' são os valores normalizados das coordenadas de P $(0 \leq (u', v') \leq 1)$.

De modo a reduzir a complexidade computacional do cálculo de uma interpolação, foram propostas algumas variantes desta técnica. Numa delas, designada por *interpolação bilinear simplificada* [46], o valor a atribuir ao ponto P é determinado a partir dos seus

três vizinhos mais próximos, através da expressão:

$$P = (P_b - P_a)u' + (P_d - P_a)v' + P_a. \tag{3.15}$$

A expressão (3.15) difere da anterior apenas pela supressão do termo $(P_a - P_b + P_c - P_d)u'v'$. Este termo representa a multiplicação de dois valores à partida reduzidos (u' e $v' \in [0,1]$), pelo que seria de esperar que o seu valor fosse pequeno e eventualmente desprezável.

De facto, esta técnica, apesar de mais simples, obtém resultados muito semelhantes aos da técnica original e bastante superiores aos da interpolação utilizando o vizinho mais próximo [26].

3.2 Estimação e compensação de movimento com transformações geométricas

Como referimos no inicío deste capítulo, o facto dos métodos *BMA* permitirem apenas a estimação de movimentos de translação, torna-os pouco adequados para a compensação de outros tipos de movimento. Esta limitação pode provocar um aumento do erro de predição, comprometendo a eficência da codificação. Por este facto, os esquemas de estimação e compensação de movimento que utilizam transformações geométricas têm vindo a ganhar relevância, ao ponto de terem sido integrados na recente norma de codificação de vídeo MPEG-4 [37].

Na secção seguinte apresentamos brevemente alguns dos aspectos principais dos métodos de estimação e compensação de movimento com transformações geométricas. O método *BMGT* será utilizado neste trabalho como referência destes métodos, na comparação de resultados, pelo que será descrito na secção 3.2.2. Este capítulo terminará com um breve resumo de alguns métodos de codificação de vídeo que utilizam transformações geométricas.

3.2.1 Estimação e compensação de movimento com transformações geométricas

O resultado da estimação de movimento com transformações geométricas, é a determinação da transformação que representa melhor as alterações entre o bloco da imagem de referência e o bloco a codificar. Com esta técnica, os parâmetros de estimação e compensação de movimento deixam de ser apenas as componentes ortogonais do vector de movi-

mento de translação, como no caso das técnicas BMA. Utilizam-se agora os parâmetros das funções de transformação que relacionam o bloco da imagem actual com o bloco de referência. Este processo encontra-se representado esquematicamente na figura 3.4.

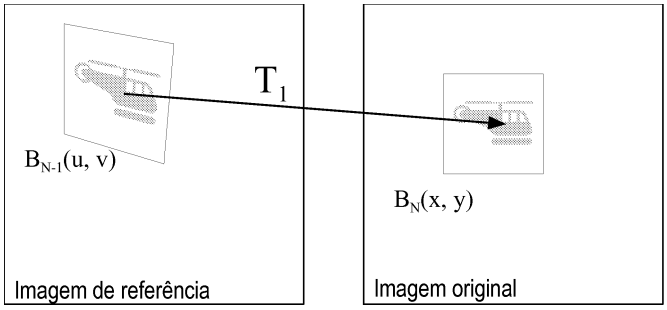


Figura 3.4: Estimação de movimento com transformações geométricas. Neste exemplo, T_1 é a transformação utilizada sobre o bloco de referência $B_{N-1}(u, v)$, de modo a obter o bloco actual, $B_N(x, y)$.

O processo de estimação de movimento com transformações geométricas consiste na transformação de um bloco de referência e na avaliação do erro entre este bloco transformado e o bloco da imagem actual, tal como acontecia no caso das técnicas BMA. Este processo pode ser descrito pelos seguintes passos [44]:

1. Seja B_N o bloco da imagem N que se pretende codificar. Para isso será utilizado o bloco B_{N-1}, com a mesma posição mas pertencente à imagem de referência (neste caso a imagem anterior);

2. De modo a podermos aplicar diversas transformações ao bloco B_{N-1}, utilizamos uma grelha adicional, designada G_{trans}. A figura 3.5a) representa a grelha original de pixels, G_{orig} (a tracejado) e a grelha G_{trans} (traço contínuo), que coincide inicialmente com os pixels. Uma transformação do bloco é executada movimentando cada um dos vértices de G_{trans} de forma independente, dentro de uma área restrita em torno da sua posição inicial, correspondente à janela de pesquisa;

3. De modo a avaliar os efeitos de uma transformação particular, os elementos do bloco B_N são transferidos da sua grelha original para a nova grelha, como representado na figura 3.5b). Os pixels da grelha transformada que não coincidam com elementos do bloco original são calculados a partir dos seus vizinhos mais próximos, utilizando um processo de *interpolação bilinear*;

4. Finalmente, a grelha G_{trans} é transformada de modo a voltar a coincidir com a grelha

original (figura 3.5c)). Esta transformação inversa dá origem a um novo bloco B'_N, correspondente à estimação do bloco original;

5. Para cada bloco reconstruído, resultante da aplicação de uma transformação particular ao bloco de referência, determina-se o valor do erro de predição utilizando um dos critérios de erro habituais. O resultado da estimação de movimento para o bloco considerado, é a transformação que minimiza o valor deste erro.

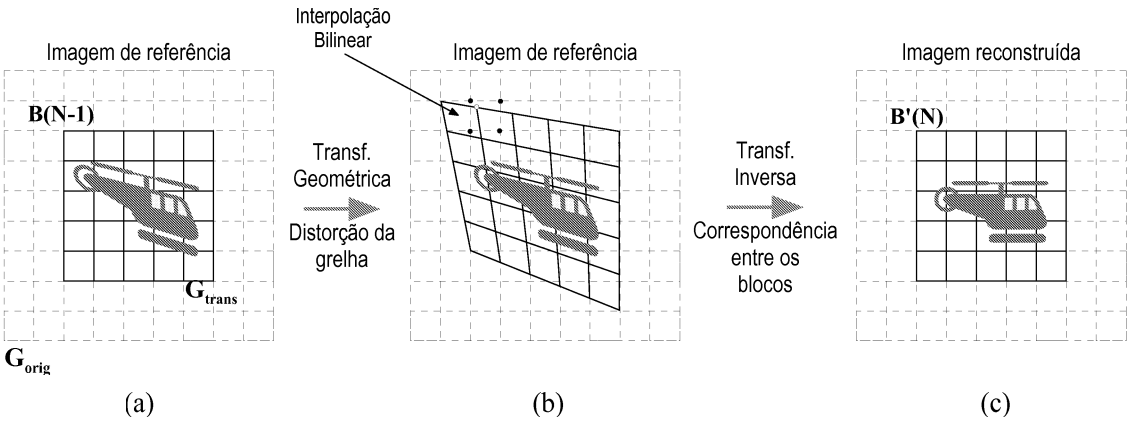


Figura 3.5: **a)** Bloco $B(N-1)$ da imagem de referência, sobre a grelha normalizada original. **b)** Aplicação da transformação geométrica e distorção da grelha. **c)** transformação inversa, reconstrução do bloco e correspondência com o bloco original.

Uma vez determinada a transformação que minimiza o erro entre os blocos, os seus parâmetros são calculados através das coordenadas dos vértices do bloco de referência e do bloco original. Estes parâmetros são então codificados e transmitidos, de modo que, no descodificador, o bloco B_N possa ser estimado pela aplicação desta transformação ao bloco da imagem de referência, no processo de compensação de movimento.

Deste modo, os parâmetros de estimação e compensação de movimento podem ser:

- os coeficientes, a_i e b_i, da transformação espacial utilizada;
- os vectores que relacionam as coordenadas dos vértices do bloco original, B_N, com os vértices do bloco transformado, B_{N-1} (ver figura 3.4).

A transmissão dos parâmetros da transformação permite a sua obtenção automática no descodificador. Se transmitirmos as coordenadas dos vectores de deslocamento dos vértices, os cálculos dos parâmetros da transformação têm de ser efectuados no descodificador. No entanto, enquanto que os parâmetros da transformação são valores reais, as

componentes vectoriais são números inteiros, permitindo uma codificação mais eficiente. Para além disso, a quantificação dos coeficientes da transformação poderia introduzir um novo erro no sistema. Por estas razões, transmitem-se normalmente os vectores que definem a transformação utilizada na compensação de movimento.

3.2.2 A utilização de transformações bilineares e o método *BMGT*

Todas as transformações geométricas apresentadas até agora (afins, perspectivas e bilineares) podem ser utilizadas na estimação e compensação de movimento com benefícios sobre a técnica *BMA* [43, 47, 44]. No entanto, devido às diferentes complexidades computacionais e transformações conseguidas, os resultados obtidos pela sua aplicação na estimação e compensação de movimento são diferentes.

Como já referimos, enquanto que as transformações afins conseguem executar apenas translações, rotações e inclinações dos eixos no plano da imagem, as transformações perspectivas e bilineares são capazes de compensar alterações mais complexas na imagem, como aproximações (*zoom*) e outros movimentos. Em [44], os três tipos de transformações são aplicados na estimação e compensação de movimento de sequências vídeo e os diferentes resultados foram analisados em termos da qualidade obtida e da taxa de informação produzida. Nesse trabalho foi demonstrada a vantagem da utilização da transformação bilinear face às transformações afins e perspectivas.

Este estudo levou à proposta de um método de estimação e compensação de movimento que utiliza transformações bilineares, designado originalmente *BMGT* (*Block Matching with Geometric Transformation*), apresentado em detalhe em [44, 48]. A técnica *BMGT* é um exemplo típico da aplicação de transformações geométricas à codificação de sinais vídeo. Seguidamente, apresentamos algumas das principais características desta técnica.

A estimação de movimento descrita na secção anterior implica a pesquisa da transformação particular que minimiza o erro de predição entre o bloco original e o bloco de referência transformado. Como já vimos, uma pesquisa exaustiva utilizando um deslocamento máximo s exige o cálculo do erro para um conjunto de $N = (2s + 1)^2$ pontos. Na determinação da transformação de um bloco quadrangular (com quatro vértices que podem ser deslocados independentemente), o número de pontos a considerar é $N^4 = (2s+1)^8$. Para cada um destes pontos deve ser determinado o bloco reconstruído e o valor do erro de predição.

De modo a reduzir a complexidade computacional deste processo, a técnica *BMGT*

utiliza um *algoritmo de pesquisa ortogonal* (ver figura 2.19). Este método foi escolhido após a comparação da complexidade computacional e dos resultados obtidos, relativamente à pesquisa exausitva e a outros métodos de pesquisa logarítmica [44].

Do mesmo modo, a realização de um estudo comparativo entre vários métodos possíveis de interpolação, levou à utilização da interpolação bilinear simplificada, em vez do vizinho mais próximo ou da interpolação bilinear convencional [44].

Os resultados da aplicação do método *BMGT* provaram a sua validade face às técnicas *BMA*, por permitir uma maior qualidade de reconstrução para taxas de informação similares. A sua utilização permitiu a codificação de sequências de imagens utilizando apenas os dados da estimação de movimento, isto é, sem a transmissão da imagem de erro residual, mantendo um nível de qualidade razoável. Estes resultados provam que a utilização de transformações geométricas para a estimação e compensação de movimento é vantajosa face aos métodos tradicionais de correspondência entre blocos com estimação de movimento de translação.

Várias técnicas foram propostas para a utilização de transformações geométricas na estimação e compensação de movimento. A secção seguinte pretende ser um resumo de algumas dessas técnicas, que introduzem aspectos interessantes à configuração do método base.

3.3 Exemplos de técnicas de compensação de movimento com transformações geométricas

Como já foi referido, as transformações afins são particularmente adequadas para a conversão de triângulos em triângulos. As técnicas de compensação de movimento que as utilizam, recorrem pois a grelhas de blocos triangulares, que vão sendo distorcidos pela a aplicação de transformações afins, de modo a representar as alterações verificadas na imagem a reconstruir.

Em [49, 50] é descrito um destes métodos designado originalmente por *TBM* (*Triangle-Based Motion compensation*). Neste método, a imagem actual é dividida numa grelha uniforme de blocos triangulares, enquanto que na imagem de referência os blocos triangulares são deformados, como se representa na figura 3.6. Na compensação de movimento, cada triângulo da imagem de referência é transformado no triângulo correspondente da imagem actual, o que define a transformação afim dos pixels da primeira imagem no bloco

regular da imagem reconstruída.

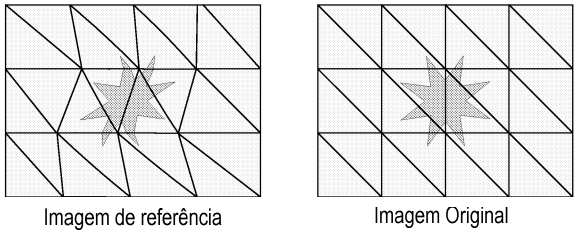


Figura 3.6: Compensação de movimento com transformações afins e uma grelha triangular contínua.

O processo de estimação de movimento determina a configuração dos blocos da imagem de referência. Este é um processo complexo, devido ao facto da deformação de um vértice da grelha triangular implicar alterações em muitos elementos da imagem.

Uma forma de estimar o vector de deslocamento de cada nó da grelha é utilizar *correspondência de hexágonos*[1]. Para cada ponto da grelha, considera-se o hexágono definido pelos seis blocos que o partilham. O vector de movimento deste ponto é determinado de forma a minimizar o erro de distorção, considerando os pontos da região hexagonal definida desta forma. Este processo pode ser repetido para toda a imagem, de forma a refinar os vectores determinados.

Noutro tipo de técnicas, a aplicação de transformações afins começa pela definição de uma grelha triangular sobre uma imagem de referência ou um objecto segmentado dessa imagem. Esta grelha de pontos[2] sofrerá uma deformação progressiva, de cada imagem da sequência para a seguinte, de modo a representar as alterações verificadas na sequência ao longo do tempo.

Este tipo de abordagem, representado na figura 3.7, foi incorporado na norma MPEG-4 [37, 40], onde as transformações da imagem são representadas por vectores de movimento associados aos nós da grelha. Ocasionalmente, podemos transmitir uma imagem *intra*, e voltar a utilizar uma grelha regular. Este método é adequado para a representação de campos de movimento contínuos e sem deformações muito acentuadas.

Outros métodos utilizam grelhas de blocos quadrangulares, associados, por exemplo, a transformações perspectivas ou bilineares. Um exemplo desta aplicação é o método *CGI* (*Control Grid Interpolation*), apresentado em [50, 51], que é uma extensão directa

[1]Este método é descrito em [50], onde se pode encontrar a descrição de outras técnicas com o mesmo objectivo.

[2]Nestas aplicações, a grelha designa-se normalmente por *mesh 2D*.

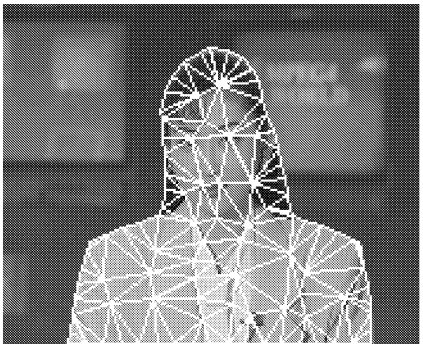


Figura 3.7: Aplicação de uma *mesh* 2D a um objecto da sequência *"Akiyo"* segundo o modelo definido na norma MPEG-4 [40].

do exemplo anterior para grelhas de blocos quadrados, deformados com transformações bilineares.

Outra técnica interessante foi proposta em [50] com o nome de *HGI* (*Hierarchical Grid Interpolation*). Neste método, a grelha quadrangular é definida não à custa de blocos homogéneos, mas sim como o resultado de um processo de segmentação baseado numa árvore quaternária. O critério de segmentação utilizado é a semelhança dos valores de um bloco, de uma imagem para a seguinte. Isto resulta na aplicação das transformações geométricas a um conjunto de blocos diSjuntos, de diferentes dimensões.

Uma característica comum a todos os métodos anteriores é a utilização de grelhas com blocos disjuntos. Estes métodos são normalmente designados por *métodos contínuos*, em contraponto aos *métodos descontínuos*. Nos métodos descontínuos os blocos da imagem de referência não têm de ser relacionados entre si, podendo existir sobreposições ou áreas não cobertas por nenhum bloco.

Um exemplo de um método descontínuo que utiliza blocos quadrangulares é o método *BMGT*, em que cada bloco da imagem de referência é definido independentemente de todos os outros, o que facilita a estimação de movimento da sequência. No entanto, a utilização de uma grelha contínua permite uma transmissão mais eficiente dos vectores utilizados na compensação de movimento, devido ao facto de cada vector ser partilhado por vários blocos.

Uma solução diferente das anteriores é proposta em [52], que define um método em que a aplicação de transformações geométricas é feita sobre blocos de formas arbitrárias. Estes blocos são definidos com base num processo de segmentação aplicado às imagens da sequência, que produz um conjunto de regiões com elementos semelhantes, diferenciadas

do segundo plano pela detecção do seu movimento.

A estimação e compensação de movimento de cada região é feita utilizando transformações perspectivas. Os parâmetros da transformação são calculados iterativamente, através do método de descida abrupta. De modo a melhorar a rapidez de convergência, a transformação a determinar é inicializada com os parâmetros correspondentes a um movimento de translação, determinado por um processo simplificado de *BMA*.

Como é possível observar neste pequeno resumo de métodos, a aplicação das transformações geométricas na estimação e compensação de movimento pode ser feita segundo diferentes métodos. Estas técnicas têm a característica comum de conseguirem uma representação mais fiel do movimento real dos objectos da cena.

Neste trabalho foi desenvolvido e estudado um novo método para a aplicação de transformações geométricas na estimação e compensação de movimento. Os detalhes deste método serão apresentados no capítulo seguinte, e os seus resultados serão discutidos e comparados com os obtidos por alguns dos métodos anteriormente apresentados.

Capítulo 4

Estimação e Compensação de Movimento com Transformações no Domínio da Luminância

Como vimos no capítulo anterior, a utilização de transformações geométricas na compensação de movimento permite ultrapassar as limitações dos métodos baseados na correspondência entre blocos (métodos *BMA*), que consideram apenas movimentos de translação dos objectos presentes na imagem. Os processos de estimação e compensação de movimento com transformações geométricas utilizam transformações espaciais de blocos da imagem de referência, para estimar os blocos da imagem a codificar.

Este método resulta sempre no aumento da quantidade de informação associada à estimação de movimento de cada bloco da imagem codificada, já que se substituem as coordenadas do vector de movimento de translação (utilizado nos métodos *BMA*), por um conjunto maior de valores que definem a transformação geométrica a utilizar. Podem ser utilizados os parâmetros reais da transformação ou os vectores de movimento associados a cada um dos vértices do bloco de referência.

Estas técnicas permitem uma melhoria da estimação da nova imagem da sequência e uma diminuição do erro de predição. Globalmente, os sistemas que utilizam transformações geométricas permitem reduzir o débito total gerado para cada imagem da sequência, ou para um mesmo débito, melhorar a qualidade da sequência descodificada [19, 43, 44, 48].

No entanto, apesar das melhorias significativas conseguidas com a utilização de transformações geométricas, subsistem ainda alguns problemas importantes, relacionados no-

meadamente com:

- a exposição de objectos da cena que anteriormente se encontravam encobertos;
- o aparecimento de novos objectos em cena;
- as mudanças causadas pelas alterações das condições de iluminação.

Estes fenómenos geram situações que mesmo as transformações geométricas não conseguem compensar convenientemente, apesar de conseguirem resultados superiores aos do método *BMA*. Esta perda de eficiência deve-se ao facto destas transformações espaciais, que actuam sobre as coordenadas dos elementos de cada imagem, não serem capazes de compensar devidamente as alterações verificadas nos valores de luminância desses elementos.

Um novo método de estimação e compensação de movimento foi desenvolvido no âmbito deste trabalho. Este método difere dos métodos anteriores pelo facto de utilizar novas transformações, realizadas no domínio da luminância do bloco a codificar.

Neste capítulo serão apresentadas estas novas transformações e serão discutidos os aspectos fundamentais da sua aplicação à estimação e compensação de movimento, num sistema de codificação de sequências de imagens. Os resultados obtidos por estas novas técnicas serão avaliados e comparados com outros métodos, nomeadamente com o método *BMA* e com as técnicas tradicionais de estimação e compensação de movimento com transformações geométricas, representadas pelo método *BMGT*.

4.1 Transformação de blocos no domínio da luminância

Tal como em todos os métodos utilizados na estimação de movimento discutidos anteriormente, pretende-se transformar um bloco de referência de modo minimizar a distorção entre o bloco resultante e o bloco original. Ao contrário das soluções anteriores que utilizavam deformações espaciais para aproximar os blocos, as novas transformações desenvolvidas neste trabalho operam sobre os valores dos elementos do bloco de referência, de modo a aproximá-los dos valores pretendidos, ou seja, dos valores do bloco original.

Seja P um bloco de elementos da imagem de referência, de dimensões $M \times N$ e Q o bloco correspondente na imagem original, que se pretende aproximar. Pretende-se encontrar uma transformação que minimize o erro resultante da estimação de movimento.

Esta transformação pode ser genericamente representada por:

$$q_{ij} = f(p_{ij}), \tag{4.1}$$

onde p_{ij} e q_{ij} representam os elementos i, j do bloco de referência e do bloco original, respectivamente. A função f pode ser uma função de variável real genérica, definida para cada par de blocos considerados.

No nosso caso foi utilizada uma função linear, com dois parâmetros c e b, definida por:

$$f(p_{ij}) = p_{ij} \times c + b. \tag{4.2}$$

Esta função foi escolhida com base num compromisso entre a sua complexidade e a quantidade de operações que é capaz de realizar sobre os valores dos elementos do bloco original. Um aumento na complexidade da função utilizada leva, por um lado, a um aumento dos dados relacionados com a compensação de movimento e, por outro lado, a um aumento da complexidade computacional dos processos de estimação e compensação de movimento.

A função apresentada em (4.2), permite a multiplicação do valor original de cada elemento do bloco por um factor de escala, c, e o deslocamento deste valor por uma constante dada por b, sendo c e b valores reais. Estas transformações são realizadas directamente sobre o valor dos elementos do bloco, ou seja, no *domínio da luminância.*

Sendo a função f dada por (4.2), a equação (4.1) tomará a forma:

$$q_{ij} = p_{ij} \times c + b, \qquad \forall_{q_{ij}} \in Q \wedge \forall_{p_{ij}} \in P. \tag{4.3}$$

A expressão anterior não permite a determinação dos parâmetros c e b, sendo necessários dois elementos de cada bloco para calcular estes valores, através da resolução de um sistema de duas equações a duas incógnitas.

No entanto, a utilização de blocos de dois pixels para a estimação e compensação de movimento utiliza um número exagerado de blocos, exigindo um conjunto proibitivo de parâmetros para a compensação de movimento. De modo a tornar este método aplicável, os blocos utilizados têm de ter, necessariamente, um número de elementos muito superior, devendo ser encontrada uma solução para a sua aplicação a blocos genéricos de dimensão $M \times N$.

Utilizar apenas os valores de dois elementos de cada um dos blocos $M \times N$, resolvendo assim um sistema de duas equações, não é claramente uma solução adequada. Isto acontece porque as alterações verificadas em cada bloco podem não ser homogéneas, quer

pela existência de vários objectos no bloco com movimentos diferentes, quer pelas razões referidas no início deste capítulo.

Torna-se pois evidente que a melhor estimativa para o valor das incógnitas c e b se obtém determinando os valores que minimizam uma função de erro para todos os elementos dos blocos. Isto leva-nos a considerar nos cálculos *todos* os elementos dos dois blocos $M \times N$, o que origina um sistema de $M \times N$ equações a duas incógnitas, ou seja um sistema *sobre-determinado.*

A solução deste sistema não poderá satisfazer simultaneamente todas as restrições impostas pelas equações correspondentes a cada um dos pontos dos blocos. Os valores dos parâmetros c e b devem então ser determinados de forma a minimizar o erro das várias equações do sistema.

A medida mais utilizada é a soma dos quadrados dos erros, dada por:

$$e = \sum_{i=1}^{M} \sum_{j=1}^{N} \quad [\, q_{ij} - (p_{ij} \times c + b) \,]^2 \tag{4.4}$$

O vector solução que minimiza este erro é denominado *solução dos mínimos quadráticos* do sistema. Note-se que a utilização deste critério é particularmente adequada para este caso pois a redução do valor obtido pela equação (4.4) implica uma diminuição do *MSE* entre os blocos.

Dependendo dos valores dos elementos dos dois blocos, que condicionam o erro mínimo conseguido por este processo, a aproximação conseguida com este método mostrou-se vantajosa face à utilização dos métodos tradicionais de compensação de movimento.

4.1.1 Determinação dos parâmetros de transformação

As $M \times N$ equações semelhantes, representadas em (4.3), correspondentes a cada par de pontos dos blocos P e Q, dão origem ao sistema:

$$\begin{cases} p_{11}.c + b = q_{11} \\ p_{21}.c + b = q_{21} \\ \qquad \vdots \\ p_{MN}.c + b = q_{MN} \end{cases} \tag{4.5}$$

Este sistema pode ser escrito na forma matricial:

$$A\mathbf{x} = \mathbf{b}. \tag{4.6}$$

onde:

- A é uma matriz com duas colunas e $M \times N$ linhas. A primeira coluna da matriz A é constituída pelos elementos do bloco P, ordenados coluna a coluna. A segunda coluna da matrix A é um vector de 1's;
- $\mathbf{x} = [\, c \quad b \,]^T$ é o vector das incógnitas;
- $\mathbf{b}$ é um vector com $M \times N$ elementos, obtidos coluna a coluna, da matriz Q dos elementos do bloco original;

O sistema matricial (4.6), construído da forma descrita, terá o seguinte aspecto:

$$\begin{bmatrix} p_{11} & 1 \\ p_{21} & 1 \\ \vdots & \vdots \\ p_{M1} & 1 \\ p_{12} & 1 \\ \vdots & \vdots \\ p_{MN} & 1 \end{bmatrix} \begin{bmatrix} c \\ b \end{bmatrix} = \begin{bmatrix} q_{11} \\ q_{21} \\ \vdots \\ q_{M1} \\ q_{12} \\ \vdots \\ q_{MN} \end{bmatrix} \tag{4.7}$$

A solução do sistema $A\mathbf{x} = \mathbf{b}$ que minimiza o erro médio quadrático de todas as suas equações é dada por $\mathbf{x} = A^+\mathbf{b}$, onde A^+ é a matriz *pseudo-inversa* da matriz A [53]. A matriz pseudo-inversa pode ser obtida a partir da matriz transposta de A, designada por A^T, da seguinte forma:

$$A\mathbf{x} = \mathbf{b} \tag{4.8}$$

$$A^T A\mathbf{x} = A^T\mathbf{b} \tag{4.8 a}$$

$$\mathbf{x} = (A^T A)^{-1} A^T\mathbf{b} \tag{4.8 b}$$

$$\mathbf{x} = A^+\mathbf{b}, \tag{4.8 c}$$

de onde se tira a expressão que define a matriz pseudo-inversa[1]:

$$A^+ = (A^T A)^{-1} A^T. \tag{4.9}$$

[1]Para os casos em que a matriz $(A^T A)$ é invertível.

A resolução da equação $\mathbf{x} = A^{+}\mathbf{b}$ dá-nos então a solução pretendida, ou seja, os valores dos parâmetros de transformação da luminância dos blocos, c e b. Considerando a equação (4.8 b) com as matrizes A e $\mathbf{b}$ representadas em função dos valores dos blocos P e Q das imagens da sequência como descrito na equação (4.7), obteremos:

$$\begin{bmatrix} c \\ b \end{bmatrix} = \left[\begin{bmatrix} p_{11} & \dots & p_{MN} \\ 1 & \dots & 1 \end{bmatrix} \times \begin{bmatrix} p_{11} & 1 \\ \vdots & \vdots \\ p_{MN} & 1 \end{bmatrix} \right]^{-1} \times \begin{bmatrix} p_{11} & \dots & p_{MN} \\ 1 & \dots & 1 \end{bmatrix} \times \begin{bmatrix} q_{11} \\ \vdots \\ q_{MN} \end{bmatrix}, \tag{4.10}$$

ou seja:

$$\begin{bmatrix} c \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij}^2 & \sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij} \\ \sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij} & M \times N \end{bmatrix}^{-1} \times \begin{bmatrix} \sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij} q_{ij} \\ \sum_{i=1}^{M}\sum_{j=1}^{N} q_{ij} \end{bmatrix}. \tag{4.11}$$

A resolução deste sistema em ordem às variáveis c e b resulta nas expressões:

$$c = \frac{MN \sum_{i=1}^{M}\sum_{j=1}^{N}(p_{ij}\, q_{ij}) - (\sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij})(\sum_{i=1}^{M}\sum_{j=1}^{N} q_{ij})}{MN \sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij}^2 - (\sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij})^2} \tag{4.12}$$

$$b = \frac{(\sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij}^2)(\sum_{i=1}^{M}\sum_{j=1}^{N} q_{ij}) - (\sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij})(\sum_{i=1}^{M}\sum_{j=1}^{N}(p_{ij}\, q_{ij}))}{MN \sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij}^2 - (\sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij})^2} \tag{4.13}$$

As equações (4.12) e (4.13) são utilizadas na determinação directa dos valores dos parâmetros pretendidos, dados os blocos P e Q. Os valores determinados desta forma poderão então ser utilizados na estimação de movimento. A secção seguinte apresenta as abordagens utilizadas para este efeito.

4.2 A utilização dos parâmetros de transformação da luminância na estimação de movimento

A utilização dos parâmetros de compensação da luminância na estimação do movimento entre duas imagens consecutivas de uma sequência, aproxima os valores dos elementos de dois blocos correspondentes dessas imagens.

Como foi referido, as alterações verificadas entre as imagens consecutivas de uma sequência podem dever-se a um conjunto diverso de factores, cujos efeitos se pretendem anular com a estimação e compensação de movimento. Deste modo, vimos que os métodos *BMA*, conseguem de alguma forma compensar os movimentos de translação dos objectos na cena. A utilização de transformações geométricas mais elaboradas permite a compensação de movimentos mais complexos, como sejam: rotações, efeitos de zoom e inclinações segundo um dos eixos da imagem.

Alguns dos fenómenos verificados são, no entanto, difíceis de compensar, por implicarem também alterações nos valores dos elementos da imagem. De modo a conseguir compensar também estes fenómenos, é necessário utilizar técnicas que combinem os efeitos dos métodos *"tradicionais"* de estimação e compensação de movimento, apresentados nos capítulos anteriores, com as novas transformações no domínio da luminância.

Deste modo, foram desenvolvidas neste trabalho duas novas técnicas de estimação e compensação de movimento, que associam à transformação dos valores dos elementos dos blocos, o método *BMA* original e a compensação de movimento com transformações geométricas, segundo o método *BMGT*[44]:

- o primeiro método foi designado por *BMAI* (*"Block Matching Algorithm Improved with pixel domain transformation"*)[2];

- do mesmo modo, o segundo método foi denominado *BMGTI* (*"Block Matching with Geometric Transform Improved with pixel domain transformation"*).

Estes métodos foram implementados, testados e comparados entre si e com os métodos originais. Nas secções seguintes serão descritos com detalhe, juntamente com as questões principais que se colocam na utilização de transformações da luminância dos blocos estimados. Após esta apresentação, os resultados obtidos com estas novas técnicas serão analisados e comparados com os resultados conseguidos pelos métodos anteriormente descritos.

[2] A razão da utilização de uma designação em língua inglesa deve-se ao facto destes métodos terem sido originalmente descritos e apresentados em artigos científicos de conferências internacionais.

4.3 A combinação da transformação da luminância com a técnica *BMA*

A estimação de movimento com a técnica *BMA* envolve, como já foi referido, a pesquisa numa janela da imagem de referência, do bloco que melhor aproxima o bloco a codificar na imagem actual. Neste processo é utilizada uma estratégia de pesquisa exaustiva, garantindo a obtenção do óptimo, ou são utilizadas técnicas de pesquisa logarítmicas, que reduzem a complexidade computacional. O resultado da aplicação destes métodos é um vector de movimento de translação, que identifica qual o bloco da imagem de referência que minimiza o erro de predição, em relação ao bloco da imagem actual.

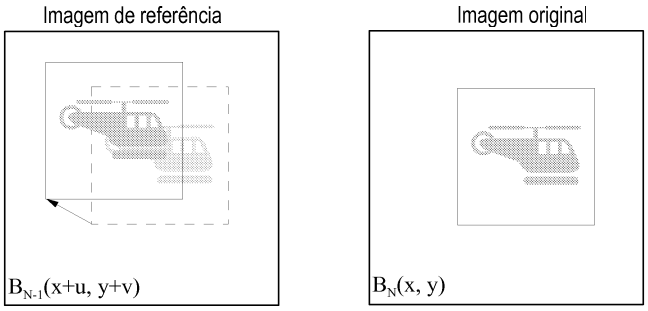


Figura 4.1: Pesquisa do vector de movimento utilizado na técnica *BMA*.

A figura 4.1 representa esquematicamente este processo, onde $B_{N-1}(x+u, y+v)$ identifica o bloco da imagem de referência obtido pela aplicação da técnica *BMA*. Este bloco foi determinado de modo a minimizar o *MAE* entre ele e o bloco original, $B_N(x, y)$.

A combinação da técnica *BMA* com a nova transformação da luminância dos pixels, designada por *BMAI*, associa a pesquisa espacial do vector de movimento com a estimação dos parâmetros da transformação de luminância.

Para cada um dos blocos de referência, $B_{N-1}(x+u, y+v)$, considerados, o processo de estimação de movimento *BMAI*, realiza as seguintes operações:

- utilizando o bloco $B_{N-1}(x+u, y+v)$ como bloco de referência, e o bloco original $B_N(x, y)$, estima os valores dos parâmtros de transformação de luminância dos blocos, c e b, utilizando as equações (4.12) e (4.13);

- após a aplicação da transformação de luminância sobre o bloco de referência, determina o *MAE* entre o bloco resultante desta transformação, designado $B'_{N-1}(x+u, y+v)$, e o bloco original $B_N(x, y)$.

É escolhido como resultado do processo de estimação de movimento, o conjunto de parâmetros que minimiza o erro médio absoluto. Estes parâmetros são as coordenadas (u, v) de deslocamento do bloco, e os valores $c_{(u,v)}$ e $b_{(u,v)}$, determinados para esse bloco de referência particular.

Este método sugere duas observações importantes:

- a utilização da técnica *BMAI* na estimação de movimento requer a transmissão de dois novos parâmetros;
- enquanto que as coordenadas (u, v) são valores inteiros, c e b são valores reais, pelo que a sua transmissão deverá envolver um processo de quantificação, que implica a introdução de erro.

A primeira observação indica que a utilização da nova técnica implica um aumento na quantidade de informação a transmitir para a compensação de movimento de cada bloco, o que é um factor penalizador deste processo.

De modo a ter em conta o erro de quantificação, referido no segundo ponto, na estimação de movimento, a quantificação dos valores de compensação é feita logo após a determinação destes. Os valores utilizados nos cálculos do *MAE* de predição, efectuados na pesquisa da transformação da luminância, são os valores resultantes da quantificação inversa dos parâmetros c e b originais, designados por c_q e b_q.

Os valores c_q e b_q são valores aproximados dos parâmetros originais. A sua utilização no processo de estimação de movimento, garante que os parâmetros resultantes permitem minimizar o *MAE* total, que inclui já o erro introduzido pela quantificação dos valores c e b.

4.4 A combinação da transformação da luminância com a técnica *BMGT*

De modo semelhante ao utilizado para a definição da técnica *BMAI*, foi estudada neste trabalho a combinação de técnicas de estimação de movimento com transformações geométricas com compensação dos valores de luminância dos elementos das imagens. O método utilizado no desenvolvimento da nova técnica foi o método *BMGT*, já apresentado no capítulo anterior. Este método utiliza transformações bilineares para a estimação de cada bloco B_N da imagem original, à custa da transformação de um bloco B_{N-1} da imagem de referência.

A nova técnica desenvolvida foi designada *BMGTI*. O processo de estimação de movimento utilizado nesta técnica combina um conjunto de passos do método original, com algumas novas operações, necessárias à utilização da nova transformação da luminância. O método *BMGTI*, representado esquematicamente na figura 4.2, pode resumir-se do seguinte modo:

- para o bloco actual, designado por $B_N(x, y)$, pesquisamos as transformações bilineares do bloco B_{N-1} da imagem de referência, que o permitem estimar;
- seja $B_N^{T_1}(x, y)$ o bloco resultante de utilização de uma transformação bilinear particular T_1, determinada no ponto anterior. Utilizando o bloco $B_N^{T_1}(x, y)$ como o bloco de referência, estimar os factores de compensação c e b que permitem minimizar o erro entre ele e o bloco B_N, utilizando as equações (4.12) e (4.13);
- após a quantificação dos valores de compensação calculados no ponto anterior, utilizar os novos valores c_q e b_q para a compensação do bloco $B_N^{T_1}(x, y)$, e determinar o *MAE* entre o novo bloco transformado, $B'_N(x, y)$, e o bloco original $B_N(x, y)$;
- repetir este processo para cada transformação considerada e escolher, como resultado da estimação de movimento, a transformação bilinear e a transformação de luminância correspondente, que permitem minimizar o *MAE* de predição, para o bloco $B_N(x, y)$.

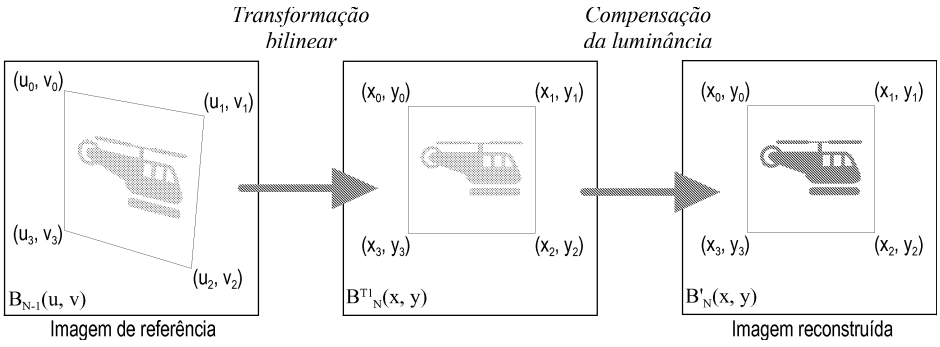


Figura 4.2: Estimação de movimento com a técnica *BMGTI*.

Os parâmetros de estimação de movimento associados à nova transformação *BMGTI* continuam a utilizar os oito valores das coordenadas dos vectores que definem a transformação espacial do bloco de referência. No entanto, é necessário transmitir também os parâmetros da transformação de luminância quantificados, c_q e b_q.

Após a explicação dos métodos base de cada uma das transformações propostas, que utilizam compensação da luminância, as secções seguintes explicarão os aspectos principais da sua aplicação no codificador de sequências utilizado.

4.5 Codificação de sequências de imagens com as técnicas *BMAI* e *BMGTI*

De modo a testar as novas técnicas de estimação e compensação de movimento, foi desenvolvido um sistema de codificação e descodificação de sequências de imagens. O desenvolvimento deste sistema teve por base o codificador descrito em [44].

O codificador desenvolvido teve como principal objectivo a codificação de uma sequência de imagens utilizando as técnicas de estimação de movimento abordadas neste trabalho: os métodos tradicionais *BMA* e *BMGT* e os métodos propostos *BMAI* e *BMGTI*, de modo a permitir a comparação dos resultados obtidos para cada um dos casos. Nesta secção serão abordadas as principais características do sistema de codificação utilizado, sendo apresentados na secção seguinte os resultados obtidos em alguns dos testes realizados.

Para além de receber a sequência de imagens a codificar, o programa de codificação recebe entre outros argumentos:

- a dimensão inicial de cada bloco quadrado utilizado no processo de estimação de movimento;
- o deslocamento máximo utilizado nas pesquisas a efectuar nos processos de estimação de movimento das várias técnicas (define a janela de pesquisa);
- o número de bits com que se pretende codificar, em média, cada imagem da sequência;
- quais as técnicas de estimação de movimento que devem ser utilizadas na codificação da sequência;
- os valores para os passos de quantificação escalar dos factores de compensação da luminância, associados às transformações *BMAI* e *BMGTI*.

O número de bits que se pretende utilizar na codificação de cada imagem é um valor meramente indicativo no início da codificação, mas vai influenciar decisivamente as opções

tomadas pelo sistema de codificação, que tentará manter a taxa de transmissão próxima do pretendido. Isto nem sempre é possível por razões que serão abordadas brevemente, verificando-se no entanto, para a grande maioria dos casos, que o débito de informação médio final se aproxima bastante do pretendido inicialmente.

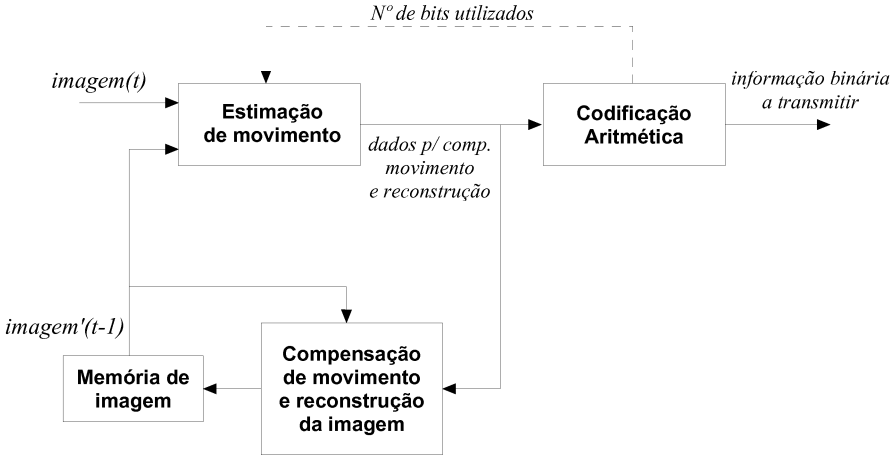


Figura 4.3: Diagrama de blocos com o esquema de funcionamento do codificador desenvolvido.

A figura 4.3 representa em diagrama de blocos o funcionamento resumido do programa desenvolvido, estando representadas as principais acções realizadas na codificação de uma sequência de entrada.

Para cada imagem de entrada, o codificador realiza a estimação de movimento utilizando a imagem de referência reconstruída. No processo de estimação de movimento é escolhida a técnica a utilizar para cada bloco, de entre as várias indicadas no início da codificação. Esta decisão é baseada na qualidade de reconstrução que cada técnica permite para o bloco considerado e no número de bits disponíveis para a codificação da imagem actual. Este processo será apresentado de uma forma detalhada na secção seguinte.

Como se pode observar pela figura, os dados resultantes do processo de estimação de movimento são comprimidos utilizando codificação aritmética, o que permite reduzir o número de bits gerado para cada imagem. Para além disso, estes dados são também utilizados para reconstruir uma réplica da imagem descodificada, utilizada como referência na estimação de movimento da próxima imagem a codificar.

Uma característica importante do sistema implementado é a não utilização da imagem de erro. Apesar de reduzir a quantidade de informação transmitida para cada imagem, este facto pode ser muito penalizante para a qualidade de codificação. A não transmissão da imagem de erro permite no entanto uma avaliação mais precisa da qualidade dos processos

de estimação e compensação de movimento. De modo a podermos utilizar este método, assume-se que a primeira imagem da sequência, codificada utilizando DCT com perdas, está disponível no descodificador, o que permite a reconstrução de toda a sequência a partir desta imagem inicial.

4.5.1 Estimação de movimento e escolha da técnica a utilizar para cada bloco

No caso de se pretender utilizar mais do que um tipo de transformação na estimação de movimento, torna-se necessária a escolha da técnica a aplicar para a codificação de cada bloco da imagem actual. Esta escolha é feita com base em dois factores importantes:

1. o erro médio absoluto conseguido para o bloco em questão por cada uma das técnicas consideradas;

2. a quantidade de informação gerada por cada uma dessas técnicas, relativamente às restantes.

Esta ponderação está relacionada com o facto de quanto mais complexa é uma transformação, maior é o conjunto de alterações que ela permite compensar, mas também maior é a quantidade de bits necessários para a codificação da informação de estimação de movimento que lhe está associada. Assim, ordenando da transformação mais simples para a que permite uma melhor estimação do movimento, e simultaneamente por ordem crescente da quantidade de parâmetros que lhe estão associados, podemos observar que:

- a técnica *BMA*, permite apenas a compensação de translações de objectos na cena da imagem de referência, mas implica a transmissão de apenas dois valores inteiros, correspondentes às coordenadas do vector de movimento do bloco;

- a técnica *BMAI*, permite a compensação de translações e de alterações na luminância dos elementos do bloco da imagem de referência, e implica a transmissão dos dois valores das coordenadas do vector de movimento e dos dois valores dos factores de compensação quantificados, c_q e b_q;

- a técnica *BMGT*, permite a compensação de transformações bilineares dos objectos na cena da imagem de referência e implica a transmissão dos oito valores das coordenadas dos vectores de deformação dos vértices do bloco de referência;

- a técnica *BMGTI*, permite a compensação de transformações bilineares e de alterações na luminância dos elementos do bloco da imagem de referência, mas implica a transmissão, para além dos oito valores da transformação bilinear, dos dois valores dos factores de compensação quantificados, c_q e b_q.

De modo a ter em conta estas diferenças, a escolha da técnica a utilizar para cada bloco é feita através da comparação dos valores do *MAE* obtidos para o bloco por cada uma das técnicas aplicadas.

Por exemplo, a técnica *BMGTI* só será escolhida para a codificação de um bloco se o ganho que permitir relativamente à técnica *BMA* for considerável, uma vez que implica um grande aumento na informação a transmitir. No entanto, para que o codificador escolha a técnica *BMAI* sobre a técnica *BMA* não é necessário um ganho tão grande ao nível do erro do bloco, já que o acréscimo de informação neste caso é muito menor.

Dados os valores dos erros para a codificação do bloco com cada uma das técnicas consideradas, a selecção da transformação a aplicar é feita pela comparação dos resultados obtidos, segundo o seguinte processo:

1. se as diferenças entre o *MAE* conseguido pela técnica *BMA* e os valores alcançados pelas restantes técnicas consideradas, não for superior a um conjunto de valores padrão predefinidos, correspondentes a cada uma destas comparações, então a técnica escolhida é a técnica *BMA* por conseguir a transmissão mais eficiente, sem uma penalização excessiva do erro do bloco;

2. entre as três técnicas restantes, é seleccionada a técnica *BMAI* apenas se o erro obtido não exceder os erros das técnicas *BMGT* e *BMGTI* por uma margem superior à estabelecida;

3. a técnica *BMGT* é seleccionada relativamente à técnica *BMGTI* se a diferença entre os erros respectivos for inferior a um valor predefinido;

4. se a qualidade obtida pela técnica *BMGTI* ultrapassa a dos restantes métodos por uma quantidade que justifique a transmissão de todos os seus parâmetros de compensação de movimento, então a técnica seleccionada é a técnica *BMGTI*.

Cada uma das comparações efectuadas em cada um dos pontos do processo de selecção descrito, envolve a utilização de um valor padrão preestabelecido. O conjunto de valores utilizados foi refinado em iterações heurísticas sucessivas, tendo-se chegado à conclusão

que estes valores deveriam depender do tamanho do bloco a codificar. O resultado foi um esquema de decisão que permite a selecção da transformação mais apropriada, após ponderada a relação entre a qualidade de reconstrução conseguida e a quantidade de informação gerada.

Uma situação que pode ocorrer, principalmente para os blocos que constituem um segundo plano fixo da cena, é a não variação do bloco da imagem de referência para a imagem a codificar. O sistema de decisão apresentado identifica estes blocos *a posteriori*, através dos blocos codificados com *BMA* utilizando vectores de movimento com componentes nulas. Estes blocos são então codificados apenas com um parâmetro que identifica o seu tipo, não sendo necessária a transmissão de qualquer valor para a compensação de movimento.

4.5.2 A segmentação dos blocos originais

Como foi referido nos capítulos introdutórios, a divisão do bloco originalmente utilizado em vários blocos de menores dimensões, permite a obtenção de melhores resultados nos processos de estimação e compensação de movimento. De modo a explorar esta situação, o codificador desenvolvido tem a capacidade de segmentar cada bloco a codificar, utilizando árvores quaternárias (ver secção 2.4.2).

Um bloco é dividido em quatro novos blocos sempre que o codificador considere que o *MAE* resultante do processo anterior não é satisfatório. Esta decisão é tomada comparando o *MAE* obtido com um valor limiar. Este valor é actualizado no final da codificação de cada imagem, considerando o número médio de bits que se pretende utilizar para cada imagem e o número de bits disponível para a imagem seguinte. Isto faz com que o número de blocos segmentados aumente quando o número de bits disponível para a codificação da imagem actual é elevado, e diminui o número de segmentações quando o codificador pretende poupar bits.

A segmentação é feita no máximo até ao segundo nível de profundidade na árvore quaternária, ou até que os blocos resultantes atinjam a dimensão mínima de 8×8. O processo completo de controlo da segmentação pode ser explicado nos seguintes passos:

1. se o *MAE* do bloco é maior que o valor limite considerado, dividir o bloco original em quatro blocos menores, desde que cada um destes blocos seja ainda maior ou igual a 8×8;

2. para cada um dos blocos resultantes da segmentação anterior, aplicar o processo de

estimação de movimento, e determinar o *MAE* associado a cada um deles;

2.1. se para um destes blocos o *MAE* resultante não for satisfatório, aplicar nova segmentação se os blocos resultantes forem não menores que 8×8;

2.2. para os blocos segmentados do segundo nível, verificar se a segmentação permite um ganho ao nível do erro face ao bloco segmentado do primeiro nível. Se sim, o novo *MAE* do bloco do primeiro nível é igual à soma dos *MAE*'s dos blocos do segundo nível, se não, a segmentação é anulada, e mantém-se o bloco original;

3. se o erro final obtido para o bloco original segmentado (determinado agora pela soma dos erros dos quatro blocos do primeiro nível) for suficientemente menor que o erro original, manter a segmentação do bloco. Se não, a segmentação deixa de ter efeito e utiliza-se o bloco com a dimensão original.

Mais uma vez, o objectivo de todo este processo é permitir ao codificador utilizar a segmentação de um bloco original sempre que a relação entre o ganho de qualidade alcançado e o acréscimo de bits utilizado seja favorável. Quer este processo, quer o processo de escolha do método de estimação de movimento a utilizar, foram desenvolvidos de modo a que o resultado final da codificação, em termos de débito binário médio por imagem, fossem o mais próximo possível do valor pretendido.

4.6 Resultados

O programa desenvolvido para a codificação de sequências, apresentado na secção anterior, foi utilizado na realização de alguns testes de codificação com as técnicas de estimação e compensação de movimento abordadas neste trabalho. De modo a ser possível analisar os resultados obtidos com as várias transformações abordadas, as sequências de teste foram codificadas quatro vezes, variando entre cada teste o conjunto de técnicas à disposição do codificador para a estimação de movimento.

A utilização da técnica *BMA* é comum a todos os testes, por permitir uma codificação muito eficiente para os blocos mais simples, cujo movimento pode ser representado por uma translação, ou que simplesmente não se movem de uma imagem para a seguinte. Por outro lado, as transformações conseguidas por esta técnica podem ser vistas como um subconjunto das transformações representadas por qualquer uma das outras três técnicas abordadas, sendo no entanto a sua codificação muito mais eficiente.

Assim, as combinações utilizadas foram:

1. *BMA+BMGT*;
2. *BMA+BMAI*;
3. *BMA+BMGTI*;
4. *BMA+BMAI+BMGT+BMGTI*.

Cada sequência foi codificada utilizando cada uma destas combinações, tendo sido os seus resultados avaliados em termos do valor da *relação sinal-ruído de pico* ou *PSNR* (*"Peak Signal to Noise Ratio"*), obtida para cada imagem codificada da sequência. Seja $I(x,y)$ a imagem original da sequência e $I'(x,y)$ a imagem reconstruída correspondente. Para imagens de dimensões $M \times N$, representadas com valores inteiros de níveis de cinzento (neste caso variando de 0 a 255) o valor da *PSNR* é dado por:

$$PSNR = 10\, log_{10}\, \frac{255^2}{\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (\, I(i,j) - I'(i,j)\,)^2} \tag{4.14}$$

As sequências de teste utilizadas são principalmente do tipo *cabeça e ombros*, por serem das mais comuns em aplicações de débitos baixos. Algumas das sequências utilizadas fogem, no entanto, um pouco ao que é normal neste tipo de aplicações, introduzindo novas dificuldades no processo de codificação. Testes com sequências de outros tipos foram também realizados, sendo os seus resultados apresentados nesta secção.

As sequências do tipo *cabeça e ombros* apresentam uma pessoa a falar, de frente para a câmara, quase sempre sem efectuar grandes movimentos e situada em frente de um fundo fixo. Esta configuração é particularmente adequada à codificação com baixos débitos por apresentar um grande número de zonas da imagem sem movimento, e não conter blocos com uma grande actividade de uma imagem para a seguinte.

A figura 4.4 representa algumas imagens da sequência *Sérgio* utilizada nos testes principais deste trabalho. Nesta sequência, um homem move a sua cabeça da esquerda para a direita, enquanto abre e fecha a boca e os olhos.

Esta situação provoca o aparecimento na imagem de áreas inicialmente encobertas, relativas à face esquerda e às zonas dos olhos e da boca. Esta sequência foi escolhida por duas razões: primeiro por apresentar todas as características de uma sequência do tipo *cabeça e ombros*, com a particularidade de exagerar os pormenores de difícil codificação, e em segundo lugar porque foi a sequência utilizada em [44] para a demonstração original da técnica *BMGT*.

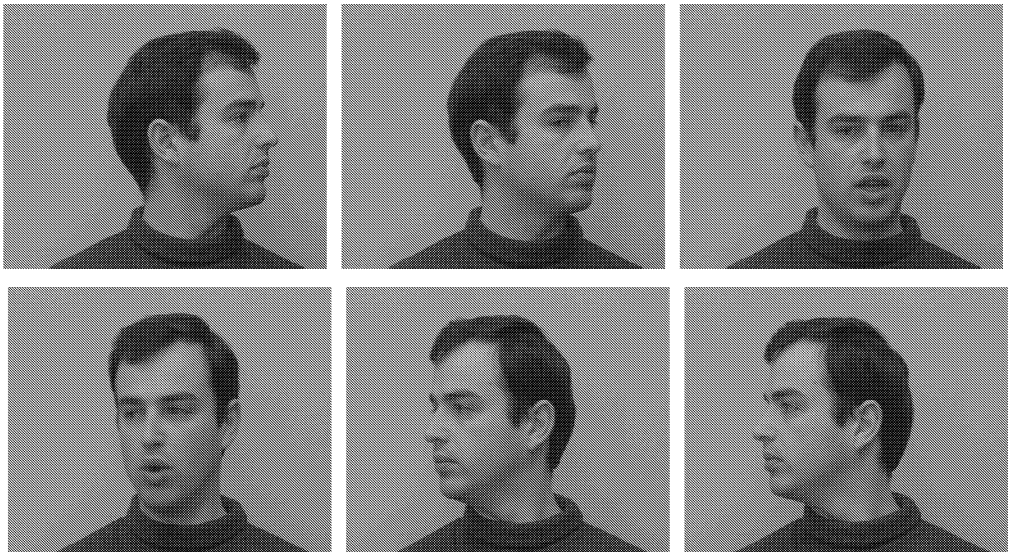


Figura 4.4: Resumo da sequência *Sérgio* utilizada nos principais testes de codificação.

A sequência é composta por 72 imagens, registadas originalmente a 25 imagens por segundo, cada uma com 352 × 288 elementos (dimensões *CIF*). Nos testes efectuados, apenas uma em cada duas imagens foi codificada, o que permite uma redução do débito binário utilizado, reduzindo a frequência temporal para 12,5 Hz. Esta vantagem ao nível da codificação provoca uma dificuldade acrescida na compensação de movimento, que tem agora o dobro da amplitude do existente na sequência composta por todas as imagens.

A sequência foi inicialmente codificada para cada uma das combinações apresentadas, utilizando blocos com uma dimensão inicial de 32 × 32 elementos, podendo estes blocos ser segmentados. A figura 4.5 mostra as curvas com a *PSNR*, obtidas para cada imagem da sequência *Sérgio*, para um débito total aproximadamente igual a 40 kbps.

Nesta figura podemos observar que, para a mesma taxa de transmissão, a utilização das novas transformações para a estimação e compensação de movimento permite sempre a obtenção de uma maior qualidade para o sinal reconstruído, face à codificação da sequência utilizando as técnicas tradicionais de compensação de movimento com transformações geométricas. Os ganhos conseguidos chegam aos 2 dB para as duas codificações com *BMGTI*, e 1 dB para a codificação com *BMA+BMAI*, sem a transmissão da imagem de erro. Isto permite-nos concluir que, apesar da utilização de um maior número de bits para a transmissão da informação de compensação de movimento, a maior eficiência na representação das alterações verificadas entre as várias imagens de uma sequência, permite às novas transformações desenvolvidas a obtenção de um aumento da qualidade

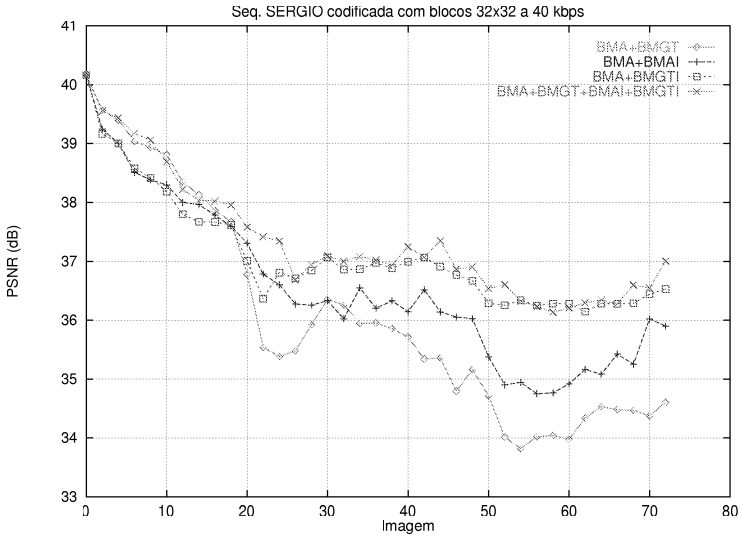


Figura 4.5: Codificação da sequência *Sérgio* com blocos 32×32.

da codificação.

A melhor qualidade patente na curva obtida com a combinação *BMA+BMAI*, relativamente à curva *BMA+BMGT*, pode ser explicada por duas razões: em primeiro lugar, a nova transformação *BMAI*, permite uma redução do erro da compensação de movimento para alguns dos blocos da imagem, face à técnica *BMGT*; por outro lado, a utilização de menos parâmetros para a compensação de movimento diminui o número de bits utilizados por cada bloco, o que permite um aumento do número de blocos segmentados, o que é, como já foi referido, um factor importante para a melhoria da qualidade de reconstrução.

Podemos também observar que entre as novas transformações desenvolvidas, a utilização de *BMGTI* permite obter melhores resultados que a transformação *BMAI*, por representar melhor as transformações verificadas nos blocos codificados. O facto desta transformação utilizar um maior número de bits que as restantes implica um decréscimo no número de blocos segmentados. Mesmo assim, esta técnica permite um ganho ao nível da qualidade visual.

A utilização de todas as transformações no processo de compensação de movimento, permite a escolha da técnica mais adequada para cada bloco codificado, pelo que seria de esperar uma melhoria da qualidade final. Esta melhoria é limitada pela taxa de transmissão imposta no início da codificação. Este factor limita as escolhas efectuadas pelo codificador, segundo as regras apresentadas na secção anterior. É no entanto notória a melhoria dos resultados obtidos para este caso. A utilização de todas as técnicas em conjunto permite a atenuação das quebras de qualidade verificadas pelas outras combinações. O resultado é a codificação da sequência de entrada com uma qualidade objectiva superior à dos restantes

casos.

A utilização de blocos de menores dimensões permite à partida uma melhoria na qualidade de reconstrução de qualquer sequência, por conseguir uma melhor representação das alterações verificadas. No entanto, o número de blocos irá obviamente aumentar, pelo que aumentará também o número de parâmetros de estimação de movimento, e logo, o débito binário gerado.

No caso particular do codificador utilizado, que não transmite a imagem de erro, todo a informação gerada corresponde à codificação dos parâmetros de compensação de movimento, pelo que uma diminuição da dimensão dos blocos tem consequências imediatas na taxa de transmissão.

A figura 4.6 ilustra os resultados obtidos na codificação de cada imagem da sequência, para blocos com uma dimensão inicial de 16×16 pixels e uma taxa de transmissão total de cerca de 75 kbps. Os blocos iniciais com 16×16 pixels, permitem a segmentação apenas até ao primeiro nível, segundo o processo descrito na secção anterior.

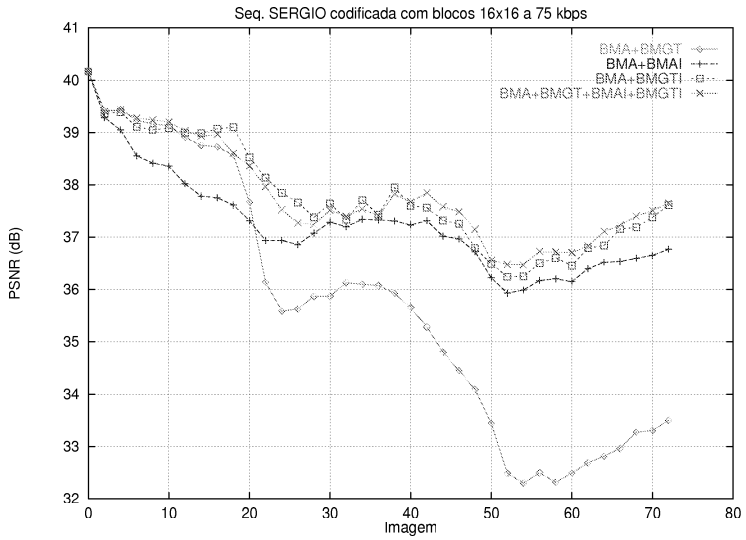


Figura 4.6: Codificação da sequência *Sérgio* com blocos 16×16.

Mais uma vez se observa uma melhoria da qualidade obtida pelas transformações propostas. Observa-se, no entanto, que os ganhos de qualidade conseguidos pela diminuição da dimensão inicial dos blocos não é considerável, relativamente ao gráfico da figura 4.5. A razão de ser deste facto prende-se com o processo de segmentação dos blocos durante a codificação. Este processo, quando utiliza blocos de maiores dimensões, permite representar grandes zonas uniformes, poupando bits que poderão ser utilizados numa codificação mais precisa das zonas com maior detalhe.

O gráfico da figura 4.7 representa os resultados da codificação da sequência *Sérgio* com blocos iniciais de dimensões 8×8. Neste caso não há segmentação, pelo que o número de blocos codificados é fixo para todas as imagens da sequência. Como consequência, a taxa de transmissão varia muito com o tipo de transformações utilizadas, sendo o controlo do débito pouco preciso. Para as situações representadas na figura, as taxas finais obtidas para as combinações *BMA+BMGT* e *BMA+BMGTI* foram próximas de 155 Kbps, enquanto que as combinações *BMA+BMAI* e *BMA+BMGT+BMAI+BMGTI* apresentaram taxas de transmissão de 165 Kbps e 171 Kbps, respectivamente.

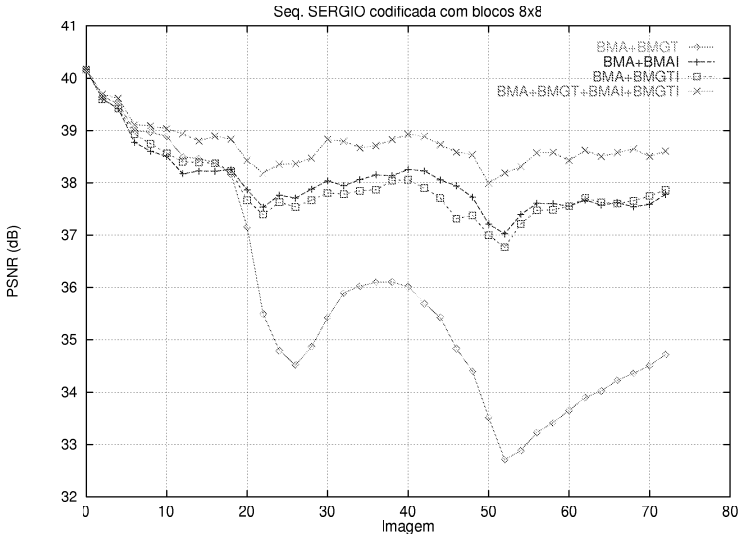


Figura 4.7: Codificação da sequência *Sérgio* com blocos 8×8.

O que se verifica neste caso, para além da subida acentuada da taxa de transmissão, é um aumento considerável também da *PSNR* das imagens reconstruídas. Genericamente, a utilização de blocos de pequenas dimensões, para além de melhorar a estimação de movimento de translação, apresenta duas particularidades interessantes:

- os blocos com poucos elementos permitem uma estimação mais precisa dos factores de compensação da luminância, através das equações (4.12) e (4.13), o que permite um aumento do desempenho das transformações propostas;

- a transformação geométrica de um bloco de pequenas dimensões, utilizando o deslocamento dos seus vértices numa grelha inteira, faz com que as deformações sobre cada elemento do bloco sejam mais acentuadas, do que no caso da utilização de blocos de grandes dimensões. Isto diminui a precisão com que cada transformação bilinear representa as deformações realmente verificadas, o que deteriora a qualidade da estimação de movimento.

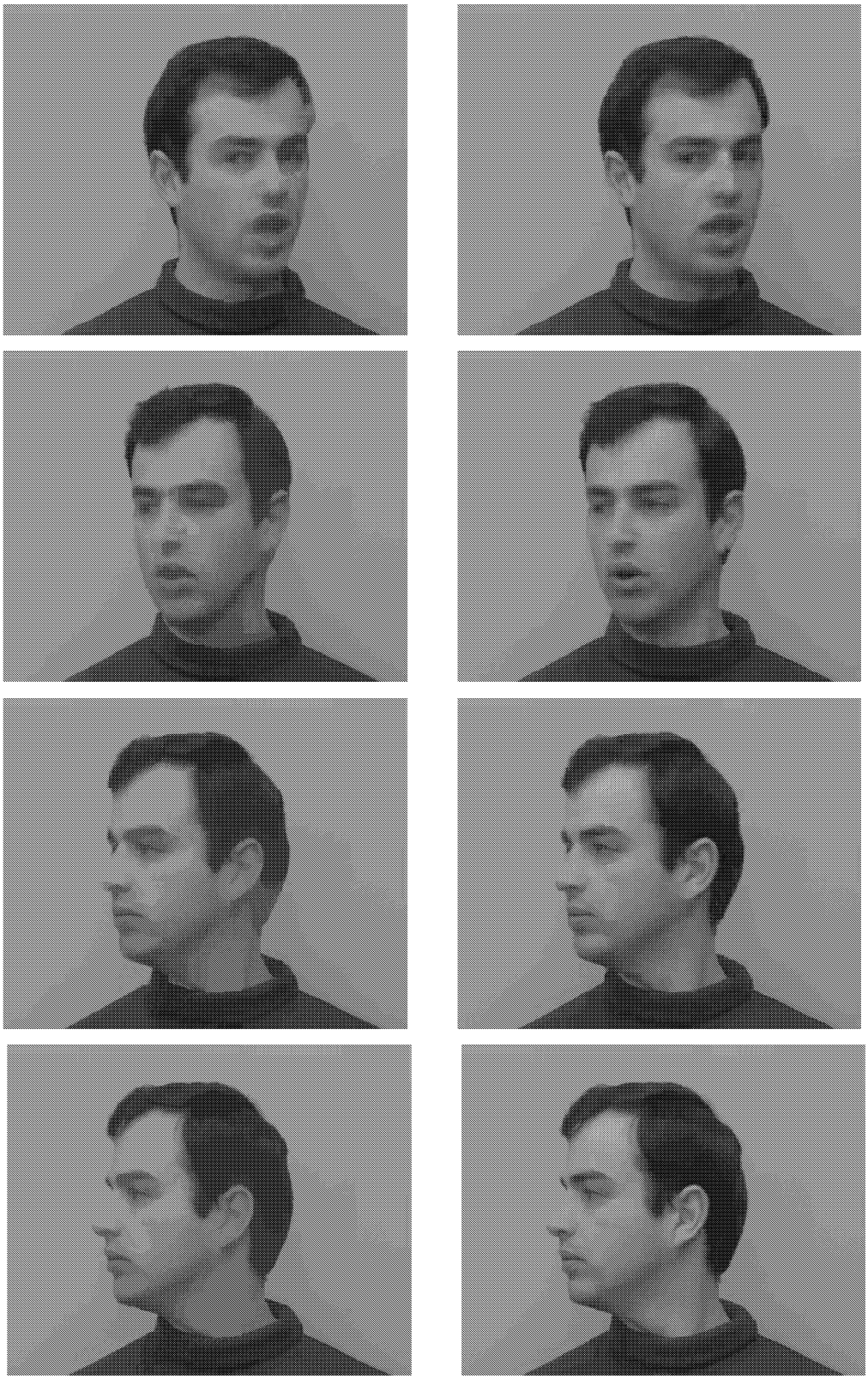


Figura 4.8: Imagens das sequências codificadas com *BMA+BMGT* (esquerda) e *BMA+BMAI+BMGT+BMGTI* (direita).

A análise dos factores objectivos de qualidade, realizada até aqui, tem uma correspondência imediata com a observação subjectiva da qualidade conseguida para cada imagem reconstruída. A figura 4.8 representa uma comparação visual entre algumas imagens resultantes da codificação com (*BMA+BMGT*) e com (*BMA+BMAI+BMGT+BMGTI*).

Pela observação das imagens apresentadas na figura tornam-se claras as vantagens das transformações propostas, ao nível da capacidade de codificação de cada imagem. Apesar de ser capaz de utilizar blocos deformados de uma imagem de referência, na reconstrução dos blocos da imagem actual, a utilização de transformações geométricas apenas no domínio espacial tem alguma dificuldade em reconstruir os pormenores das áreas recém expostas das imagens, bem como em manter a qualidade das áreas anteriormente existentes.

Esta dificuldade é notória nas zonas de muito detalhe como os olhos, o nariz e a boca do sujeito da cena, que apresentam alguma distorção, desde as imagens iniciais. Podemos ver, no entanto, que as transformações geométricas são adequadas para a codificação das zonas do plano de fundo, que surgem na imagem à medida que a cabeça roda para a direita. Este efeito é conseguido com o *"esticar"* das áreas do segundo plano já existentes, de modo a que estas cubram as novas secções que vão surgindo ao longo do tempo.

Os problemas das transformações tradicionais aumentam para os blocos que correspondem a áreas que surgem de novo na cena e que apresentam algum detalhe. Estas áreas correspondem principalmente à face esquerda que se encontra inicialmente encoberta e que é progressivamente exposta pelo movimento da cabeça. O aumento da qualidade de reconstrução é notório principalmente na zona da orelha e olho esquerdos, na linha de cabelo e genericamente em todas as áreas com maior detalhe, como os lábios e o nariz.

Outra melhoria conseguida pelas novas transformações observa-se nas *"tonalidades"* (níveis de cinzento) das imagens reconstruídas. Nas imagens resultantes da codificação com as transformações tradicionais é notória uma certa degradação dos tons da face esquerda. Isto deve-se ao facto desta face ter sido completamente reconstruída à custa da deformação de blocos que existiam na imagem anterior, pelo que não é possível o ajuste conveniente da sua luminância. Este ajuste só é possível com as novas transformações, que operam também sobre os valores dos elementos das imagens.

4.6.1 Efeitos da variação dos parâmetros de codificação

Depois de realizar a codificação de algumas sequências de teste, de modo a estabelecer empiricamente os níveis de decisão utilizados no sistema de codificação (cuja aplicação foi explicada na secção anterior), foram investigados os efeitos da variação de cada um dos parâmetros principais da codificação, sobre a qualidade final da sequência codificada. Assim interessa investigar a variação da qualidade das imagens codificadas com: a dimensão inicial do bloco utilizado, a taxa de transmissão média pretendida, a dimensão da janela de pesquisa utilizada na pesquisa dos vectores de movimento e deformação dos blocos, e finalmente, a influência dos passos de quantificação utilizados na codificação dos valores dos parâmetros de compensação do contraste e do brilho.

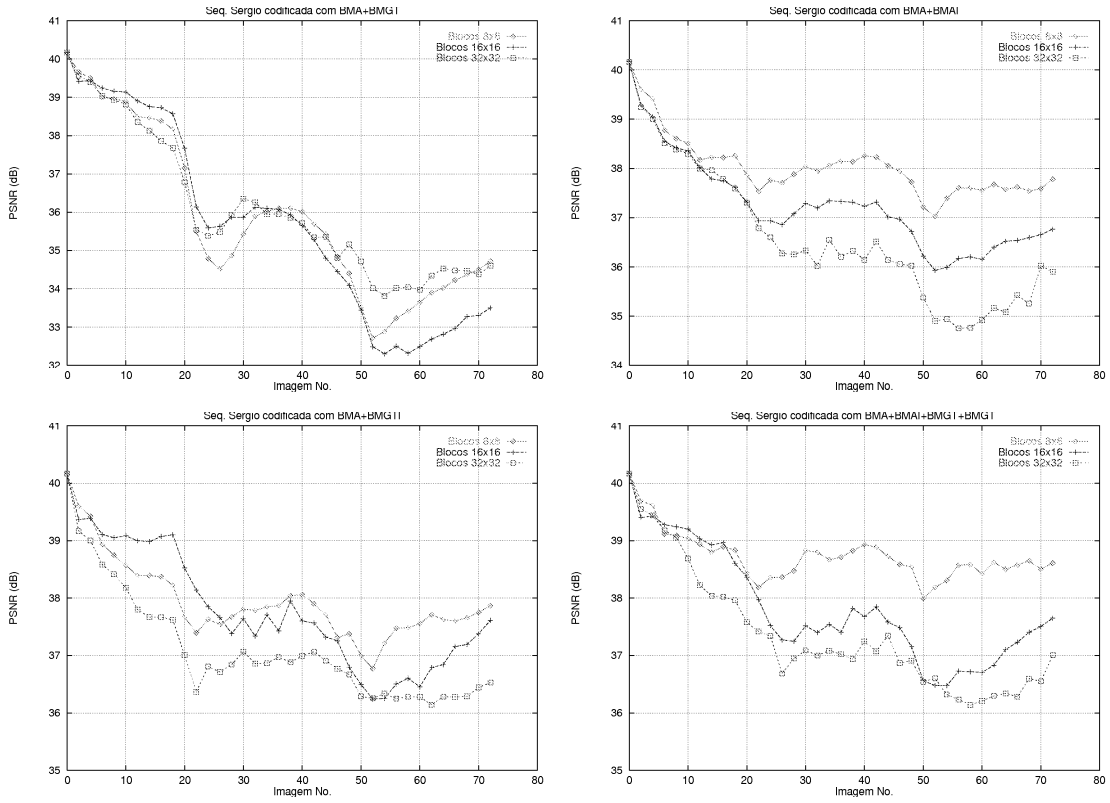


Figura 4.9: Variação da *PSNR* com a alteração do tamanho inicial dos blocos para a sequência *Sérgio*.

Os gráficos da figura 4.9 representam as curvas da relação sinal-ruído de pico para a codificação da sequência *Sérgio* utilizando blocos originais com dimensões 32×32, 16×16 e 8×8 e para as taxas de transmissão referidas.

Como já foi discutido anteriormente, o processo de segmentação incluído no codificador atenua os efeitos da redução da dimensão inicial do bloco, permitindo ao codificador

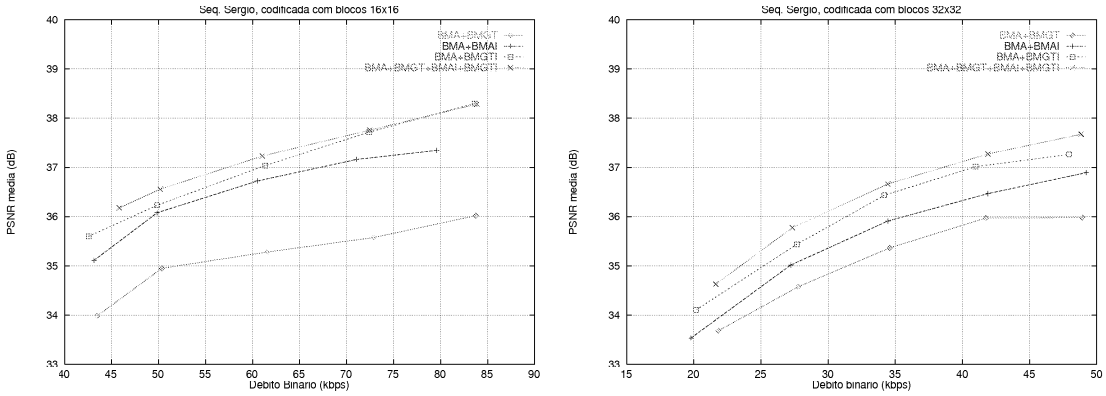


Figura 4.10: Variação da PSNR com a taxa de transmissão para a sequência *Sérgio*.

explorar as zonas uniformes de grandes dimensões com blocos não segmentados, o que resulta numa economia de bits que poderão ser utilizados em áreas com maior detalhe. No entanto, a tendência geral representada nos gráficos da figura é a da melhoria da qualidade objectiva das imagens codificadas com a diminuição da dimensão dos blocos, melhoria essa que é também acompanhada por um aumento inevitável da taxa de transmissão associada a cada caso.

Os gráficos da figura 4.10 mostram a variação da PSNR média com a taxa de transmissão na codificação da sequência *Sérgio*. A sequência foi codificada utilizando blocos com um tamanho inicial de 16×16 elementos, (com um deslocamento máximo de 8 pixels na pesquisa dos vectores de movimento) e com blocos 32×32 (com uma janela de pesquisa com um deslocamento máximo de 16 pixels). Variámos em cada teste efectuado a taxa de informação utilizada.

A figura 4.10 mostra a maior qualidade conseguida pelas transformações propostas, para todas as taxas de transmissão utilizadas. É possível observar também que os ganhos obtidos com as novas técnicas, com transformações da luminância, são aproximadamente constantes, chegando a ultrapassar ligeiramente os 2 dB, para os vários débitos testados com blocos 16×16.

Verifica-se também que, à medida que a taxa de transmissão aumenta, os resultados da técnica *BMA+BMGTI* aproximam-se dos obtidos pelo método *BMA+BMAI+BMGT +BMGTI*. Isto acontece porque, ao haver mais bits disponíveis para a codificação de cada imagem, o grande número de parâmetros utilizados pela transformação *BMGTI* torna-se menos penalizante, face à utilização da técnica *BMA*.

Os gráficos da figura 4.11 representam os efeitos da variação da janela de pesquisa sobre a qualidade final da sequência reconstruída. O valor em abcissas é o deslocamento

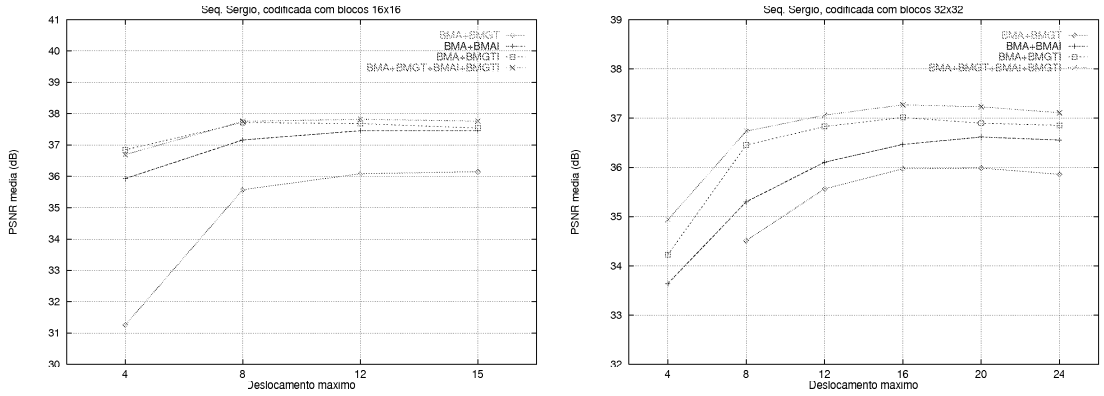


Figura 4.11: Variação da PSNR com o deslocamento máximo da janela de pesquisa, para a sequência *Sérgio*.

máximo considerado na pesquisa de cada um dos vectores de movimento (técnicas *BMA* e *BMAI*) ou dos vectores que definem a transformação bilinear utilizada na estimação de movimento (técnicas *BMGT* e *BMGTI*).

Se o valor considerado para o deslocamento máximo for inferior ao número de pixels que um determinado bloco se move efectivamente, então este factor vai introduzir uma limitação no processo de estimação de movimento. Este facto é notório nos casos em que se considerou um deslocamento máximo de 4 pixels, que se mostrou insuficiente para a sequência utilizada.

Quando o deslocamento máximo aumenta, ultrapassando o número de pixels que em média um bloco se desloca na cena, qualquer aumento adicional deste parâmetro de codificação é irrelevante. O aumento da dimensão da janela de pesquisa para lá do valor referido pode ter na realidade um efeito adverso: ao aumentar o deslocamento máximo, estamos a aumentar também a gama de variação dos valores que serão introduzidos no codificador entrópico, o que reduz a eficiência deste sistema, levando à utilização de mais bits na codificação da informação de estimação de movimento. Os efeitos deste fenómeno são reduzidos, mas podem ser observados nas curvas dos gráficos da figura 4.11.

Os três gráficos da figura 4.12 representam a variação da *PSNR* média, para as 72 imagens da sequência *Sérgio* codificada, em função dos valores dos passos de quantificação utilizados para os factores de compensação da luminância. Os testes foram efectuados utilizando blocos com uma dimensão inicial de 16×16 elementos, um deslocamento máximo de 8 pixels e uma taxa de transmissão de 75 kbps.

A análise dos resultados dos testes efectuados permite concluir que os efeitos da variação do passo de quantificação para o factor de compensação c são mais notórios

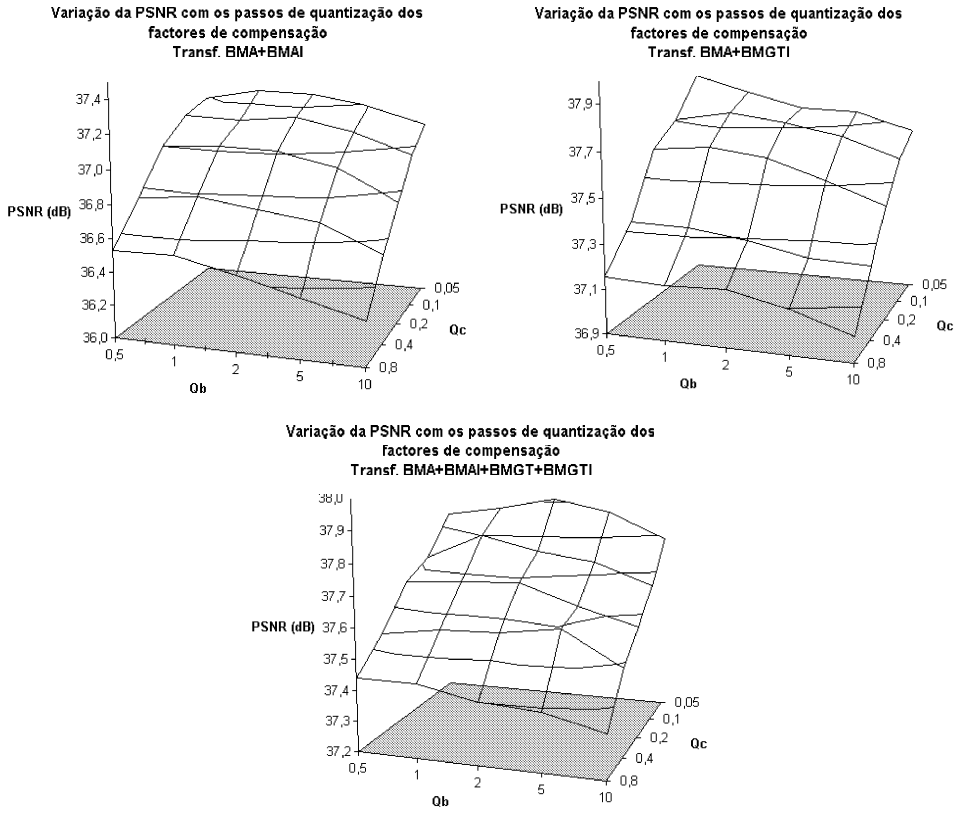


Figura 4.12: Variação da PSNR com os passos de quantificação dos factores de compensação da luminância, Q_c e Q_b, para a sequência *Sérgio*.

que os efeitos da variação do passo de quantificação para o factor de compensação b.

Por outro lado, é possivel observar que a diminuição dos passos de quantificação utilizados só é benéfica até um certo ponto, a partir do qual se verifica um decréscimo na qualidade. Este fenómeno está associado ao aumento do número de bits causado pela utilização de passos de quantificação mais finos, que terá de ser compensado com uma poupança de bits ao nível da segmentação dos blocos ou na selecção da transformação utilizada.

Todos os resultados apresentados nas figuras 4.5 a 4.11 desta secção, foram obtidos utilizando passos de quantificação iguais a $0,1$, para o valor c e 2, para o valor b.

4.6.2 Resultados da codificação de outras sequências

Outras sequências foram codificadas com as técnicas propostas, tendo os resultados obtidos demonstrado consistentemente uma melhoria da qualidade de codificação. De seguida, alguns dos resultados dos testes efectuados serão apresentados.

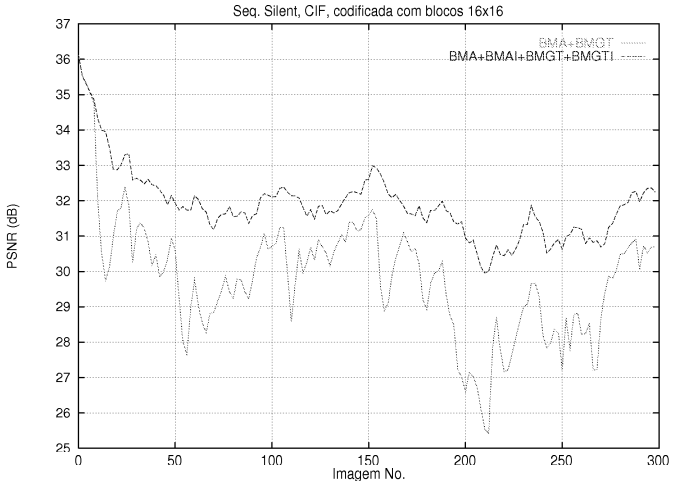


Figura 4.13: Resultados da codificação da sequência *Silent* com as transformações convencionais (*BMA+BMGT*) e as novas transformações (*BMA+BMGT+BMAI+BMGTI*).

A figura 4.13 apresenta os resultados de codificação obtidos para a sequência *Silent*, com formato *CIF*. Esta sequência representa uma mulher a falar com linguagem gestual, e é interessante por envolver uma grande quantidade de movimento dos membros superiores, embora não haja o aparecimento de muitas áreas antes encobertas. Os resultados apresentados correspondem à codificação das 300 imagens da sequência, utilizando blocos com uma dimensão inicial de 16 pixels, um deslocamento máximo de 8 pixels para a pesquisa dos vectores de movimento e uma taxa de transmissão de aproximadamente 73 kbps. Mais uma vez foi utilizada uma subamostragem temporal por 2 das imagens da sequência original.

Podemos observar que a utilização das novas transformações na estimação e compensação de movimento, permite a codificação de um grande número de imagens da sequência, sem a transmissão da imagem de erro. Os ganhos de qualidade obtidos em relação à utilização dos métodos tradicionais de compensação de movimento com transformações geométricas, atingem os 4 dB, sendo a curva da *PSNR* final muito mais regular para o caso das novas técnicas.

Codificação de sequências *QCIF*

O formato *QCIF* é muito utilizado em aplicações de codificação de vídeo com baixos débitos, como a videotelefonia e a videoconferência. É pois importante a realização de alguns testes de codificação de sequências com este formato. Os resultados obtidos confirmaram uma vez mais a melhoria de qualidade que as novas técnicas permitem alcançar. A figura 4.14, representa as curvas de *PSNR* obtidas na codificação da sequência *Sérgio*,

(formato *QCIF*).

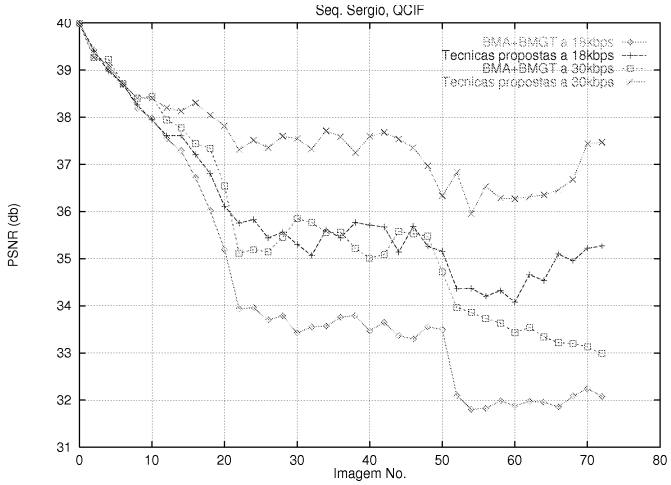


Figura 4.14: Resultados da codificação da sequência *Ségio* (*QCIF*) com as técnicas convencionais e as novas técnicas.

A figura 4.15 representa os resultados da codificação da sequência *Mother and Daughter* (formato *QCIF*). Nesta figura podemos observar uma queda abrupta da relação sinal-ruído da sequência codificada com as técnicas tradicionais, entre as imagens 56 e 58.

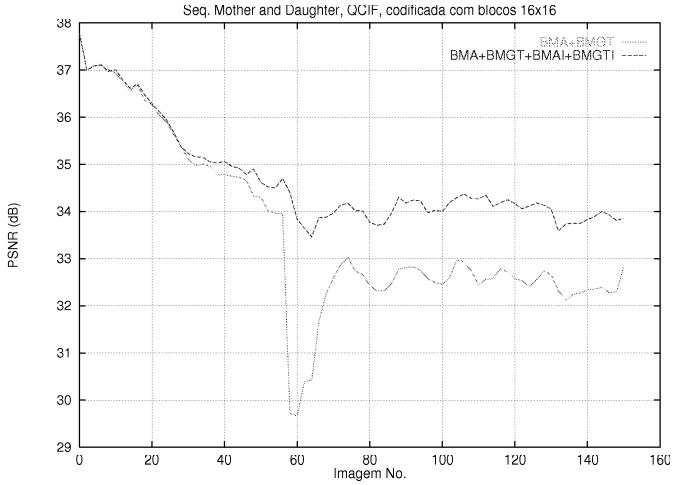


Figura 4.15: Resultados da codificação da sequência *Mother and Daughter* (*QCIF*) com as transformações originais e as novas transformações.

Se analisarmos a sequência original em pormenor (figura 4.16), podemos ver que este instante corresponde ao momento em que a senhora faz um movimento brusco, que causa o aparecimento do seu braço esquerdo na imagem. A compensação desta situação mostrou-se extremamente difícil com as transformações geométricas tradicionais, na ausência de transmissão da imagem de erro. A utilização das novas transformações na estimação de movimento, permite, no entanto, ultrapassar esta dificuldade apenas com uma perda

mínima de qualidade, comparada com a verificada no caso anterior.

Figura 4.16: Pormenor da sequência *Mother and Daughter*: imagens 56, 58, 60, 62 e 64.

4.7 O método de *Kamikura et al*

Quando o autor deste trabalho se encontrava já em fase de conclusão do presente documento, encontrou um artigo publicado em finais de 1998 [54], que descreve um método com alguns pontos comuns com o trabalho aqui apresentado (já aceite para publicação na altura em que o artigo referido foi publicado [55]).

A técnica apresentada em [54] utiliza também a compensação do valor da luminância, como modo de melhorar a eficiência dos processos de estimação e compensação de movimento. Seguindo um raciocínio idêntico ao apresentado nas secções iniciais deste capítulo, os autores chegam às mesmas equações para o cálculo dos coeficientes de compensação da luminância de um bloco (equações (4.12) e (4.13)). Este conceito de estimação dos valores de compensação utilizando uma técnica de optimização por mínimos quadrados é o ponto em que os dois trabalhos se tocam.

A aplicação destes valores na estimação e compensação de movimento é, no entanto, feita pelos autores de [54] de um modo bastante diferente do que foi aqui apresentado. Os factores de compensação calculados são utilizados apenas para a compensação global das alterações da luminância, ou seja, é determinado apenas um conjunto de valores para toda a imagem. Para determinar esse par de valores os autores seguem o seguinte algoritmo:

1. para cada bloco de dimensão fixa 16×16, determinar o valor dos parâmetros de compensação e do vector de movimento que minimizam o erro médio quadrático entre o bloco de referência e o bloco actual, num processo de pesquisa exaustiva idêntico ao utilizado na técnica *BMAI*, apresentado na secção 4.3;

2. os valores globais utilizados na compensação de toda a imagem são os dois valores que ocorrem com mais frequência, entre os pares determinados no ponto anterior

para cada um dos blocos da imagem. Estes valores são quantificados e transmitidos uma vez por cada imagem;

3. para cada bloco a codificar, é feito um teste de modo a determinar se a compensação com os valores obtidos no ponto anterior provoca uma melhoria da qualidade de reconstrução, face ao método de correspondência entre blocos. Se isso acontecer, esse bloco é marcado com um bit de controlo que é transmitido para o descodificador.

Este método de estimação e compensação de movimento foi integrado no algoritmo de codificação *"Verification Model"*, versão 3, do grupo MPEG-4. Os resultados apresentados revelam uma adequação particular deste método à compensação de fenómenos como o ajuste da abertura da lente, situações de *fade in* e *fade out* e mudanças de iluminação, ou seja, situações em que os valores de luminância variam uniformemente para toda a imagem. No entanto, foram também utilizadas cenas em que esta variação não é tão uniforme ao longo de toda a imagem, tendo os autores verificado também melhorias em relação ao método de codificação convencional.

A transmissão de apenas um par de valores de compensação para cada imagem, implica à partida uma redução da taxa de transmissão utilizada por este método, relativamente aos métodos desenvolvidos neste trabalho, que utilizam factores de compensação diferentes para cada bloco codificado. No entanto, seria de esperar também que esse acréscimo de informação permitisse um aumento da qualidade de reconstrução da imagem codificada.

De modo a ter uma ideia do desempenho destes dois métodos, a sequência *Sérgio* foi codificada utilizando uma implementação de método descrito em [54] (sem a transmissão da imagem de erro). Foram utilizados passos de quantificação de $1/32$ e 2, respectivamente para os valores c e b, de compensação da luminância e blocos fixos de 16×16, como descrito no artigo.

A figura 4.17 representa os resultados obtidos, em termos da *PSNR* para cada imagem reconstruída da sequência. Os resultados obtidos são comparados com outros dois resultados obtidos com os métodos descritos neste trabalho:

- um método que utiliza apenas as transformações *BMA+BMAI*, semelhantes às utilizadas pelo esquema de [54], com os mesmos passos de quantificação e sem a utilização de segmentação. Com este teste, pretende-se avaliar a eficácia da utilização de diferentes factores de compensação da luminância para diferentes blocos da imagem, relativamente à compensação global das variações de luminância;

- o método de codificação com compensação de movimento utilizando as transformações *BMA+BMAI+BMGT+BMGTI*, com segmentação e de blocos de dimensão 32×32;

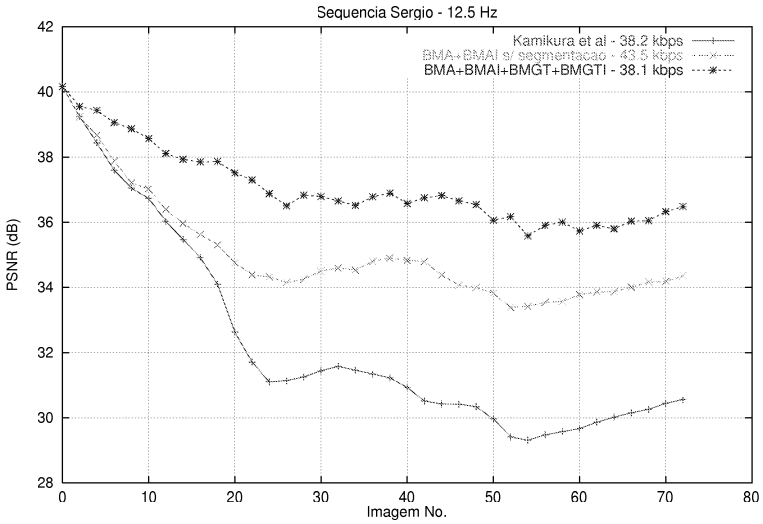


Figura 4.17: Resultados da codificação da sequência *Sérgio* com o método de *Kamikura et al* e com os métodos apresentados neste trabalho.

A partir do gráfico da figura, podemos concluir que a utilização de um par de valores de compensação para cada bloco é um factor de melhoria da qualidade de reconstrução. No entanto, num esquema que utilize *BMA+BMAI* sem segmentação, o codificador não tem onde poupar bits, pelo que cada novo bloco que utiliza a transformação *BMAI* origina a transmissão de dois novos valores. Isto causa um aumento razoável da taxa de transmissão, que é o preço da melhoria notória da qualidade do sinal. A taxa de transmissão seria idêntica à obtida pelo método de [54] se fosse codificado apenas um bloco com *BMAI* e todos os restantes com *BMA*, o que não seria uma comparação adequada. Ao nível da complexidade computacional não há grandes diferenças, já que as operações envolvidas nestes dois métodos são muito semelhantes.

Quando se utiliza a compensação da luminância de cada bloco associada à estimação de movimento com transformações geométricas, num sistema de codificação com segmentação e blocos iniciais de maior dimensão, podemos verificar um aumento considerável da relação sinal-ruído, para taxas de transmissão semelhantes. Isto deve-se ao elevado número de possibilidades que o sistema tem ao seu dispor para a codificação de cada bloco, que lhe permite poupar bits nos blocos menos problemáticos, utilizados na codificação mais perfeita dos blocos com maior detalhe.

No entanto, todo este esquema de decisão requer um maior tempo de computação, já que cada bloco é codificado com todas as transformações possíveis, assim como todos os blocos resultantes da sua segmentação. Finalmente é tomada uma decisão baseada na qualidade obtida por cada uma das codificações efectuadas. Isto torna este esquema bastante mais complexo do que o método apresentado em [54].

Capítulo 5

Codificação de Sequências Estereoscópicas

Uma das evoluções possíveis da televisão actual é a chamada *televisão estereoscópica* (ou *TV3D*). A utilização de vídeo estereoscópico permite a percepção de profundidade, melhorando a noção da posição relativa dos objectos da cena, entre outras vantagens perceptuais para o utilizador do sistema.

Com o desenvolvimento da televisão digital e da televisão de alta definição, a capacidade de utilizar sinais de vídeo estereoscópico é neste momento tecnologicamente viável. Para além da difusão de TV3D, o vídeo tridimensional pode ser utilizado em aplicações de educação e treino (por exemplo, simuladores de voo e condução), na indústria cinematográfica e de entretenimento e em aplicações médicas e de tele-operação [36].

Neste capítulo investigamos a aplicação das transformações desenvolvidas neste trabalho, apresentadas anteriormente, na codificação de sequências estereoscópicas. Após uma breve introdução aos aspectos principais da utilização de vídeo tridimensional, serão discutidas algumas técnicas de codificação de sequências estereoscópicas. Estas sequências podem ser encaradas com um caso particular de sinais de vídeo com *vistas múltiplas*, que serão apresentados brevemente na secção 5.2.3.

Finalmente será discutida a aplicação das transformações geométricas e de luminância propostas na codificação de sequências de vídeo 3D. Esta aplicação foi baseada em várias técnicas já conhecidas para a codificação de sequências estereoscópicas. Os resultados obtidos com as várias técnicas implementadas, bem como as implicações da utilização das novas transformações no domínio da luminância, serão então apresentados e comentados.

5.1 Introdução

A utilização de sequências estereoscópicas permite uma melhor percepção das distâncias e profundidades, melhorando a percepção da posição relativa dos objectos na cena. Para além disso, este tipo de sequências permite uma melhoria na qualidade de imagem e resolução subjectivas e na percepção dos reflexos e cintilações da cena.

O sistema visual humano permite uma percepção tridimensional do mundo devido, principalmente, a dois tipos de factores: factores *psicológicos*, como a análise das sombras, das oclusões e das dimensões relativas dos vários objectos na cena e factores *fisiológicos*, dos quais se destacam:

- *acomodação*: consiste no ajuste da distância focal das lentes dos olhos humanos pelos músculos que os rodeiam;
- *convergência*: consiste no ajuste da direcção de cada um dos olhos, de modo a que a intercepção dos eixos ópticos coincida com o objecto que se pretende visualizar. O ângulo definido pelos dois eixos ópticos designa-se por *ângulo de convergência*;
- *disparidade binocular*: consiste na diferença que existe entre as imagens adquiridas por cada olho, devido ao facto destes se encontrarem ligeiramente afastados (tipicamente entre os 62 e os 70 mm, para um adulto);
- *paralaxe de movimento*: consiste nas diferenças nas imagens de cada vista devidas ao movimento da cena ou do observador.

Estes factores estão na base do desenvolvimento dos sistemas de aquisição e representação de sequências estereoscópicas. Uma discussão breve mas interessante deste assunto pode ser encontrada em [56].

5.1.1 Técnicas de visualização de vídeo estereoscópico

As técnicas de visualização de sinais de vídeo tridimensionais podem ser classificadas em dois grupos: técnicas *autoestereoscópicas* (não requerem nenhum tipo de óculos ou outro acessório para a percepção tridimensional) e técnicas *estereoscópicas*, que exigem a utilização de um acessório especial.

Os métodos de visualização autoestereoscópica mais conhecidos são talvez os hologramas. Outras técnicas utilizadas são, entre outras, as técnicas volumétricas e as técnicas baseadas em paralaxe. Uma discussão genérica destes métodos pode ser encontrada em

[36]. Em [56, 57, 58] são também propostas novas soluções para a implementação de sistemas de visualização autoestereoscópicos.

Nas técnicas estereoscópicas são normalmente utilizadas duas imagens bidimensionais, apresentadas independentemente a cada um dos olhos do observador. Estas sequências distintas são normalmente adquiridas por duas câmaras diferentes, cuja disposição permite a obtenção de uma sensação realista de profundidade.

Como já foi referido, estas técnicas exigem a utilização de um acessório especializado. Exemplos destes acessórios são os tradicionais óculos vermelho-azul ou com lentes polarizadas, que permitem de algum modo filtrar a imagem destinada a cada olho. Outros dispositivos mais complexos, consistem por exemplo em sistemas com LCD's que o utilizador coloca directamente em frente dos olhos. Algumas das técnicas estereoscópicas de visualização de vídeo tridimensional são apresentadas em [36].

5.1.2 Geometria estereoscópica

Na aquisição de imagens estereoscópicas utilizam-se sistemas com duas configurações principais: com câmaras *convergentes* ou com câmaras *paralelas*.

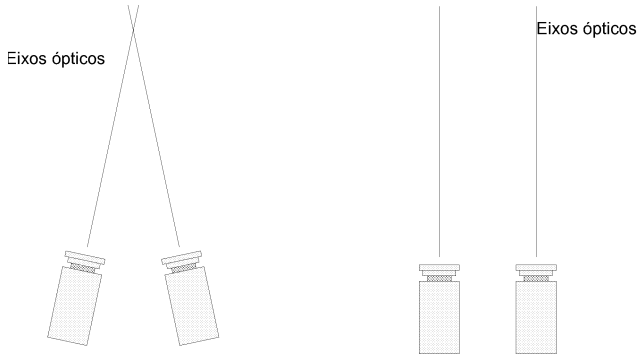


Figura 5.1: Sistemas de aquisição de imagens estereoscópicas com configuração: **a)** convergente e **b)** paralela.

Nos sistemas com câmaras convergentes (figura 5.1 a), as duas câmaras são ligeiramente rodadas de modo a criar um ponto de convergência entre as duas. Estes sistemas seguem o modelo do sistema de visão humano, tendo a desvantagem de introduzir distorções geométricas entre as imagens adquiridas.

Os sistemas de câmaras paralelas (figura 5.1 b) não têm este inconveniente, sendo no

entanto mais difíceis de implementar, pois exigem lentes com um maior campo de visão e um sistema de ajuste mecânico mais complexo [36].

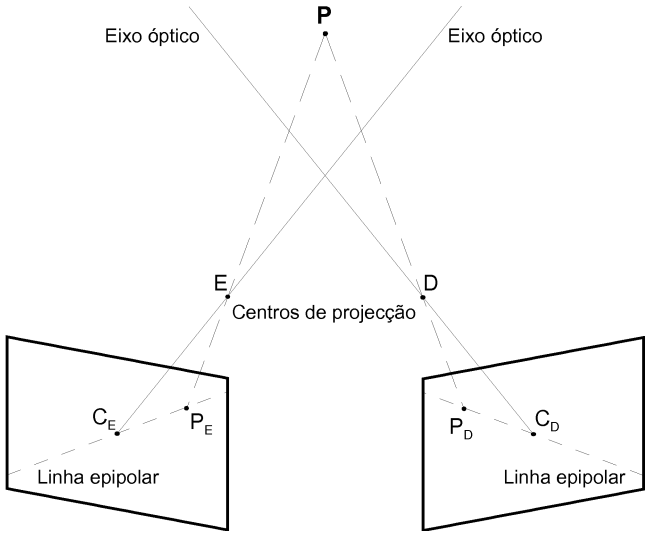


Figura 5.2: Geometria de um sistema estereoscópico de câmaras convergentes, com eixos ópticos coplanares.

A figura 5.2 representa a geometria de um sistema estereoscópico convergente de eixos ópticos coplanares. Um ponto P no espaço 3D é projectado em perspectiva nos pontos P_E e P_D, respectivamente nas imagens esquerda e direita, segundo linhas de projecção que passam pelos centros de projecção E e D das lentes.

A *disparidade* do ponto P é a distância entre os pontos P_E e P_D, quando as imagens das duas vistas são alinhadas. O valor da disparidade de um ponto é inversamente proporcional à sua distância aos centros de projecção. A determinação dos pares de pontos (P_E, P_D) correspondentes é designada por *estimação da disparidade.*

Este problema é normalmente bidimensional. No entanto, se os eixos ópticos são coplanares, os pontos P_E e P_D correspondentes pertencem a uma linha, designada por *linha epipolar.* Esta linha é definida pela intersecção do plano de cada imagem com o plano definido pelos pontos P, E e D. Neste caso, a pesquisa do ponto correspondente pode ser restringida ao espaço unidimensional da linha epipolar.

No caso particular de uma configuração paralela (figura 5.3), as linhas epipolares são horizontais e coincidentes com as linhas de pixels das imagens das vistas, o que evita a determinação da linha epipolar.

A estimação da disparidade de um ponto nem sempre é possível, devido a oclusões que se verificam de uma vista para a outra.

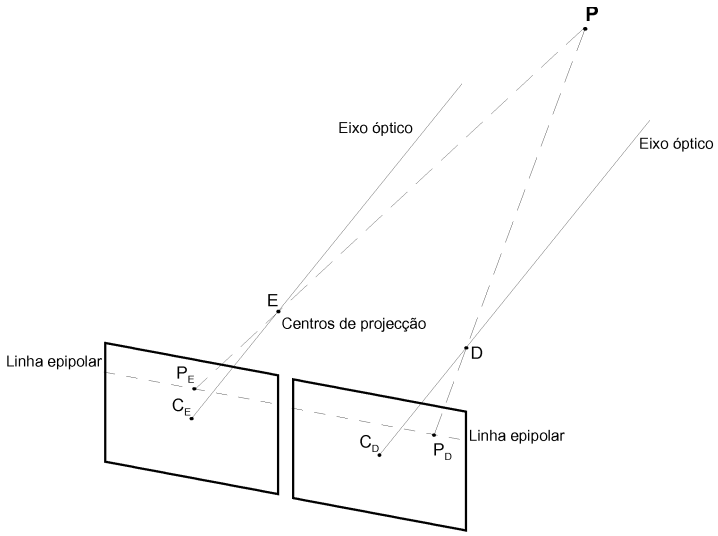


Figura 5.3: Geometria de um sistema estereoscópico de câmaras paralelas.

A geometria do sistema de visualização tem de respeitar as características do sistema de aquisição. A correspondência entre estes dois sistemas é no entanto um processo complicado, devido ao número de transformações envolvidas desde a aquisição até à visualização: a transformação entre o espaço tridimensional da cena e os dois espaços bidimensionais das imagens esquerda e direita, entre estes e os espaços dos ecrãs e finalmente a correspondência para o espaço tridimensional de visualização.

Mesmo em sistemas bem construídos, a utilização de vídeo estereoscópico pode introduzir um conjunto de distorções visuais, que dificultam a percepção da localização dos objectos na cenas e reduzem a qualidade subjectiva do sinal visualizado. Algumas destas distorções são a curvatura do plano de profundidades, a distorção trapezoidal, a não linearidade da profundidade e a distorção radial [36].

Independentemente destas distorções, a visualização de sinais estereoscópicos pode causar um desconforto fisiológico e mental, que normalmente aumenta com o tempo de visualização. Este fenómeno é causado pela diferença entre os pontos de convergência e de acomodação, já que os olhos do utilizador convergem para um ponto virtual, situado a trás ou à frente do ecrã, enquanto que a focagem é feita no plano do ecrã propriamente dito.

O sistema visual humano tem no entanto uma característica favorável à codificação de sequências estereoscópicas. Quando se apresenta uma imagem perfeita a um olho e uma imagem de má qualidade ao outro, a maioria das pessoas tem a percepção de uma imagem simultaneamente nítida e tridimensional [59]. Este facto pode ser explorado codificando

uma das sequências com menor qualidade, o que permite reduzir a taxa de transmissão que lhe está associada.

5.2 Codificação de sequências estereoscópicas

A utilização de sequências estereoscópicas implica, à partida, a transmissão do dobro da informação envolvida na codificação de sequências normais (monoscópicas). Daqui se compreende bem a necessidade de esquemas eficientes de compressão de sinais de TV3D.

Existem três abordagens fundamentais ao problema da codificação de sinais de vídeo estereoscópicos:

- técnicas *simulcast*, que codificam de forma independente cada uma das sequências das vistas esquerda e direita. Isto significa que ambas as sequências podem ser descodificadas em receptores convencionais;

- técnicas de codificação estereoscópica *compatíveis*, onde uma das sequências é codificada de forma independente e a outra é codificada relativamente a esta. Isto significa que é ainda possível reproduzir uma das sequências num receptor convencional, razão da designação dada a este tipo de técnicas;

- técnicas de codificação *conjunta*, em que as duas imagens são codificadas conjuntamente, o que normalmente impede a sua descodificação por um descodificador monoscópico.

À primeira vista as técnicas *simulcast* serão menos eficientes que qualquer uma das restantes, pois não exploram as semelhanças entre as imagens correspondentes de cada vista. Do mesmo modo, será também natural pensar que as técnicas de codificação conjuntas serão as mais eficientes em termos da compressão conseguida.

No entanto, as técnicas compatíveis são as mais atraentes do ponto de vista da implementação prática, por permitirem o melhor compromisso entre a eficiência de codificação e a compatibilidade com os sistemas monoscópicos. Se considerarmos o cenário da migração de sistemas convencionais de TV digital para sistemas de TV3D, a utilização destas técnicas permitiria aos utilizadores continuar a utilizar os seus equipamentos monoscópicos, sem perda de compatibilidade, até optarem pela aquisição de um sistema estereoscópico.

5.2.1 Estimação e compensação da disparidade entre vistas

A disparidade consiste na diferença entre as imagens das duas vistas num dado instante, devida às diferentes perspectivas das duas câmaras que filmaram a cena. As técnicas de compressão de sequências estereoscópicas exploram as semelhanças entre as duas vistas utilizando *estimação e compensação de disparidade*.

A compensação da disparidade entre as duas vistas pode ser mais complexa do que a compensação de movimento, que discutimos nos capítulos anteriores. No entanto, as técnicas normalmente utilizadas na estimação e compensação de movimento podem também ser aplicadas no caso da disparidade.

Em sequências de vídeo estereoscópicas, adquiridas com um sistema de câmaras que tende a seguir o modelo da vista humana, o valor da disparidade vertical é tipicamente muito reduzido. O valor da disparidade horizontal pode ser no entanto bastante elevado (superior às diferenças típicas verificadas numa sequência devido ao movimento), dependendo da cena a ser filmada e da geometria do sistema de aquisição [59].

Algumas simplificações podem ser feitas nos processos de estimação e compensação de disparidade, se forem consideradas as características do sistema de aquisição. Uma simplificação comum é considerar que a disparidade vertical é nula, quando se utilizam sistemas de câmaras paralelas, em que as linhas epipolares são paralelas às linhas dos pixels das imagens.

5.2.2 Codificação compatível de sequências estereoscópicas com estimação e compensação de disparidade

Os métodos compatíveis de codificação de sequências estereoscópicas utilizam a codificação de uma das vistas (normalmente designada por *vista principal*) de uma forma independente e fazem a codificação da outra vista (*vista auxiliar*) em relação a esta. A codificação da vista auxiliar em relação à vista principal utiliza normalmente esquemas de estimação e compensação de disparidade, que podem ser associados ou não à estimação e compensação de movimento.

A figura 5.4 representa um diagrama de blocos genérico de um sistema com estas características. Neste sistema, os dados gerados pela codificação da sequência principal e da sequência auxiliar são multiplexados. No descodificador, os dados recebidos são desmultiplexados e utilizados na descodificação da sequência a que correspondem.

A sequência principal é codificada independentemente, utilizando por exemplo com-

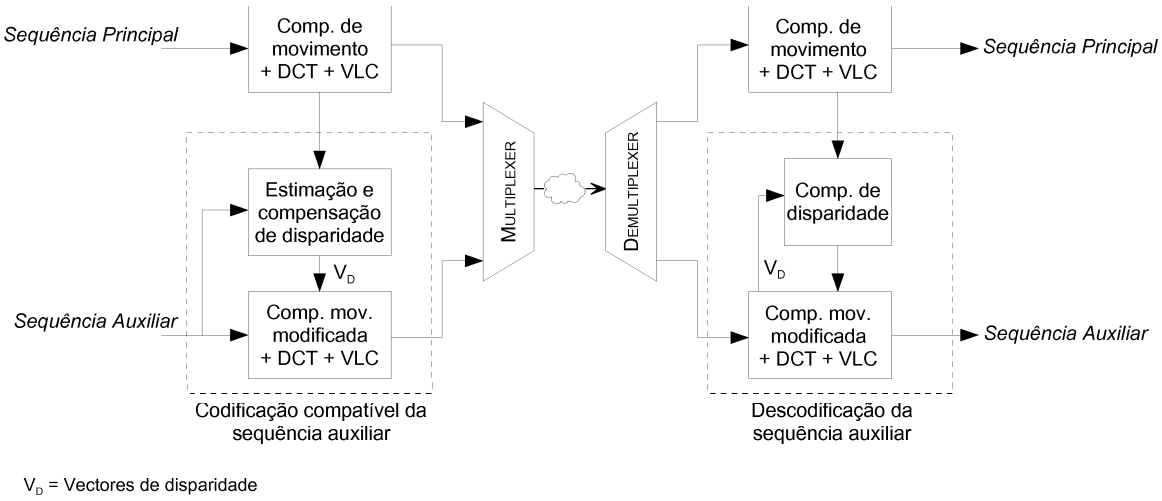


Figura 5.4: Diagrama de blocos de um codificador compatível genérico de sequências estereoscópicas.

pensação de movimento e DCT com blocos. Tal como anteriormente, é construída no codificador uma cópia de cada imagem descodificada da sequência, que servirá de referência para a codificação da imagem seguinte.

No caso da codificação de sequências estereoscópicas, essa imagem é também utilizada como referência no processo de estimação e compensação de disparidade da sequência auxiliar. Neste caso, os métodos utilizados são normalmente idênticos aos métodos aplicados à estimação de movimento de sequências monoscópicas.

A codificação da sequência auxiliar pode utilizar simultaneamente estimação e compensação de disparidade e de movimento, sendo neste caso utilizadas também como referência, as imagens reconstruídas da sequência auxiliar. Em qualquer um dos casos, a compensação de disparidade visa reduzir o erro de predição, que pode ser codificado como acontecia no caso das sequências monoscópicas.

Na figura 5.5 são representadas duas hipóteses de codificação da sequência auxiliar utilizando apenas estimação e compensação de disparidade. No primeiro caso (fig. 5.5a)) é utilizada compensação de disparidade simples, enquanto que na segunda hipótese (fig. 5.5b)), a estimação e compensação de disparidade é realizada à custa de duas imagens de referência, pertencentes à sequência principal. Esta técnica será designada por compensação *bidireccional* de disparidade, por analogia com a compensação bidireccional de movimento.

A figura 5.5c) representa um esquema de codificação compatível que combina a estimação e compensação de disparidade com um processo de estimação e compensação de

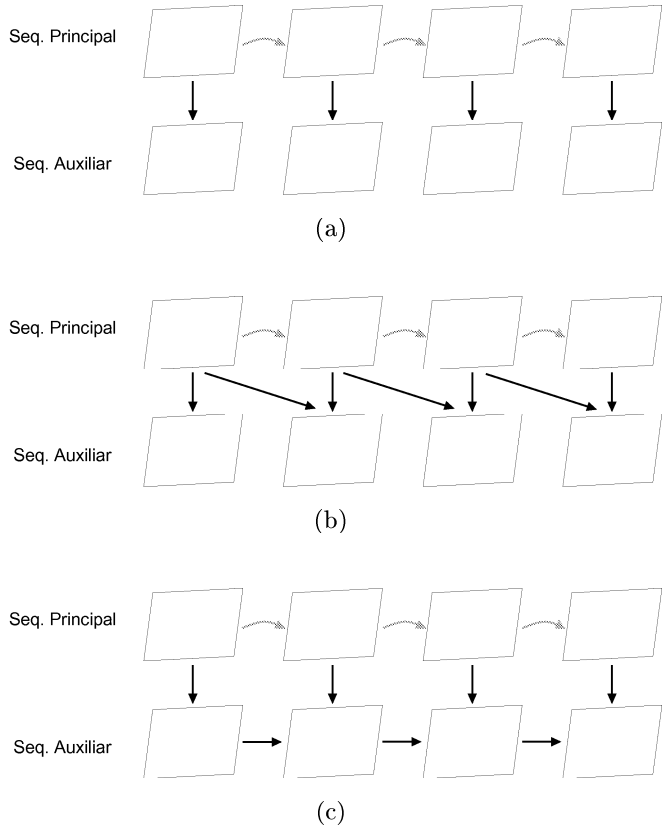


Figura 5.5: Codificação compatível de sequências estereoscópicas utilizando: **a)** estimação simples de disparidade, **b)** estimação bidireccional de disparidade e **c)** estimação de movimento e de disparidade.

movimento, relativo à imagem anterior da sequência auxiliar.

A codificação da sequência auxiliar pode utilizar diversos tipos de predição e usar também periodicamente imagens do tipo I, de modo a reduzir o erro da codificação.

5.2.3 Codificação de sequências com vistas múltiplas

Sinais de vídeo com vistas múltiplas[1] são, como o próprio nome indica, caracterizados pela existência de várias sequências, correspondentes a diferentes pontos de vista da cena filmada. O vídeo estereoscópico pode ser encarado como um destes casos, em que o número de vistas é limitado a dois.

As aplicações de sequências de vídeo com vistas múltiplas incluem mais uma vez situações de treino e educação interactivas, televisão interactiva e aplicações na indústria de entretenimento, entre outras. A utilização de várias perspectivas permite um maior

[1] Tradução do inglês *multiviewpoint video*.

realismo e percepção da cena visualizada, possibilitando explorar o que se encontra por trás dos objectos, para além de permitir ao utilizador uma melhor estimação das distâncias e posições relativas dos objectos.

A apresentação destas sequências é feita com sistemas específicos, que consideram na sua implementação a geometria utilizada no sistema de aquisição. Um caso bem conhecido deste tipo de aplicações é o chamado vídeo panorâmico, que apresenta numa tela circular, colocada em torno do utilizador, uma perspectiva de 360^{o} da uma cena filmada com várias câmaras organizadas circularmente. Outros tipos de ecrãs podem ser baseados em sistemas lenticulares.

Em [36] é feita uma breve discussão de alguns dos tipos de sistemas utilizados na aquisição e apresentação de sequências de vídeo com vistas múltiplas.

Ao nível da codificação destas sequências, mais uma vez se percebe a necessidade do desenvolvimento de técnicas que permitam reduzir a taxa de transmissão envolvida, que seria originalmente a taxa de uma sequência normal multiplicada pelo número de vistas. As técnicas utilizadas classificam-se usando as mesmas três categorias empregues na codificação de sequências estereoscópicas: técnicas *simulcast*, técnicas *compatíveis* e técnicas de codificação *conjunta*.

Existem dois tipos de soluções para reduzir a taxa de transmissão necessária na codificação: as técnicas que limitam a resolução e as técnicas que limitam o número de pontos de vista codificados [36].

No primeiro caso, as imagens de várias vistas são sub-amostradas e agrupadas numa só imagem, de modo que a resolução da imagem resultante do agrupamento seja idêntica à resolução original. Deste processo resultam duas sequências, que são então codificadas utilizando uma técnica de codificação estereoscópica. Depois de descodificada, cada imagem é dividida, sendo extraídas as imagens das sequências originais, que podem então ser interpoladas de modo a recuperar a sua dimensão original.

No caso das técnicas que utilizam restrição dos pontos de vista codificados, o número das sequências transmitidas é limitado, sendo as restantes vistas interpoladas pelo sistema de apresentação. As vistas transmitidas podem ser codificadas utilizando uma expansão directa dos sistemas de codificação estereoscópica. Considerando o caso da transmissão de três vistas, a vista central pode ser codificada independentemente, sendo cada uma das vistas laterais codificadas em relação a esta, segundo um modelo de codificação estereoscópica compatível. No receptor, uma vista correspondente a uma perspectiva não

codificada pode ser reconstruída à custa destas três vistas.

5.2.4 Exemplos de técnicas propostas para a codificação de vídeo estereoscópico

Algumas das técnicas de codificação de sequências estereoscópicas e com vistas múltiplas foram desenvolvidas de modo a manter a compatibilidade com as normas de codificação de vídeo existentes. Entre estas destacam-se as técnicas de codificação compatíveis que utilizam os vários perfis escaláveis da norma MPEG-2. Nestas técnicas, a sequência correspondente à vista principal é codificada na camada base, sendo utilizadas as camadas de expansão na transmissão das sequências auxiliares.

Estas técnicas são discutidas genericamente em [36]. Descrições de exemplos particulares destes métodos podem ser encontrados em [60], para sequências estereoscópicas e em [61] para o caso de sequências com vistas múltiplas.

Em [60] é também apresentado um esquema de codificação de sequências estereoscópicas que utiliza as técnicas de codificação de sequências interlaçadas definidas na norma MPEG-2. Neste caso, cada vista da sequência é considerada como um campo de uma sequência virtual interlaçada, cuja codificação é feita utilizando os esquemas de predição baseados em campos.

Outras técnicas de codificação de sequências estereoscópicas, não compatíveis com as normas MPEG, foram também propostas. Em [62] é apresentada uma técnica de codificação conjunta, que efectua a compressão de sequências estereoscópicas no domínio da frequência.

Na técnica apresentada em [63] são definidas à partida as posições de um conjunto de imagens de ambas as sequências, organizadas periodicamente, que são codificadas como imagens do tipo I. As restantes imagens de ambas as sequências serão codificadas segundo um de dois modelos: ou utilizando estimação e compensação de movimento baseada em blocos, em relação a uma única imagem I da mesma sequência, ou utilizando uma mistura de estimação e compensação bidireccionais de movimento e disparidade. Neste caso, cada imagem é codificada à custa da imagem anterior da mesma sequência e de três imagens de referência da sequência da outra vista.

Em [64, 65] é proposto um esquema de codificação de sequências estereoscópicas que utiliza também uma combinação de compensação de disparidade entre vistas e de movimento dentro de uma sequência. A compensação da disparidade é no entanto efectuada

sobre um conjunto de blocos de dimensão variável, organizados numa árvore quaternária. Estes blocos são definidos de modo a que, dentro de um bloco, o valor da disparidade entre as imagens seja aproximadamente constante.

A utilização de esquemas de quantificação perceptual, baseados nas características de percepção binocular humanas, permitem por vezes uma melhoria do desempenho subjectivo da codificação. Em [66] é utilizada uma técnica de codificação de imagens estereoscópicas baseada na compensação de disparidade e utilização de DCT. Neste método, as matrizes com os passos de quantificação dos coeficientes da DCT foram construídas de forma a explorar as características particulares da imagem de erro residual resultante da compensação de disparidade.

Muitas técnicas utilizam na codificação de sequências 3D o conhecimento dos parâmetros geométricos e ópticos do sistema de aquisição. Este conhecimento é normalmente utilizado na reconstrução de vistas auxiliares a partir de vistas conhecidas, utilizando métodos de visão computacional.

Um exemplo de uma técnica de codificação de sequências estereoscópicas que assume o conhecimento de todos os parâmetros do sistema de aquisição é descrito em [67]. Após a segmentação dos objectos da cena e a construção de um mapa que define as suas profundidades, a codificação é baseada na estimação e compensação do movimento tridimensional de cada um desses objectos. Este método permite construir directamente a vista auxiliar à custa da vista principal. Esta técnica foi incorporada num codificador híbrido que utiliza também compensação tradicional de disparidade e de movimento.

Um conjunto de técnicas que também utilizam o conhecimento de todos os parâmetros do sistema de aquisição e que têm vindo a ser utilizadas na codificação de sequências estereoscópicas são as técnicas baseadas em objectos. Um exemplo de uma destas técnicas de codificação pode ser encontrado em [68].

5.3 Codificação de sequências estereoscópicas com transformações no domínio da luminância

Após um breve estudo das principais técnicas de codificação de sequências estereoscópicas, estudámos a aplicação das transformações desenvolvidas neste trabalho à codificação deste tipo de sequências.

No capítulo anterior verificámos que a utilização das transformações de luminância na

estimação e compensação de movimento, permite uma melhoria na qualidade de codificação de sequências monoscópicas. Nomeadamente, foi observado que os melhores resultados eram obtidos para o caso em que se utilizava a combinação das quatro transformações consideradas ($BMA+BMAI+BMGT+BMGTI$), associada a blocos com uma dimensão inicial de 32×32 pixels.

Pretendemos agora verificar até que ponto a utilização das transformações de luminância é adequada à estimação e compensação da disparidade entre vistas, para além do movimento entre imagens consecutivas. Para isso, o codificador utilizado foi desenvolvido de modo a permitir a codificação compatível de sequências estereoscópicas.

De modo a investigar as diferenças de desempenho entre os vários métodos de codificação compatível discutidos na secção 5.2.2, foram implementados três tipos de codificadores estereoscópicos. Todos eles codificam a sequência principal (neste caso a esquerda) de um modo independente, ou seja, utilizando o método descrito no capítulo anterior. A codificação da sequência direita (sequência auxiliar) foi feita segundo os três modelos já apresentados:

- codificação com *compensação de disparidade simples* (figura 5.5a)), que utiliza como referência para a codificação de cada imagem direita apenas a imagem esquerda correspondente;

- codificação com *compensação de disparidade bidireccional* (figura 5.5b)), que codifica cada imagem direita à custa da imagem esquerda correspondente e da imagem esquerda anterior;

- codificação com *compensação de disparidade e compensação de movimento* (figura 5.5c)), que codifica cada imagem direita à custa da imagem esquerda correspondente (compensação de disparidade) e da imagem direita anterior (compensação de movimento).

Em cada caso, foram testadas as transformações geométricas tradicionais (métodos $BMA+BMGT$) e as novas transformações no domínio da luminância (com $BMA+BMAI+BMGT+BMGTI$). Os resultados destes métodos foram também comparados com a codificação independente das duas sequências, ou seja a utilização de uma técnica *simulcast*.

No caso das codificações que utilizam mais do que uma imagem de referência (compensação de disparidade bidireccional e associada à compensação de movimento), a escolha da imagem de referência a utilizar foi feita para cada bloco da imagem codificada. Isto

implica que cada bloco da imagem actual seja codificado duas vezes, utilizando cada uma das imagens de referência consideradas.

Qualquer uma destas codificações utiliza o método explicado no capítulo anterior, ou seja, escolha da transformação a utilizar de entre as disponíveis e segmentação, ou não, do bloco a codificar, baseadas num compromisso entre a taxa de transmissão e o erro de codificação. O resultado deste processo é um conjunto de parâmetros que permitem codificar o bloco actual à custa de um bloco de referência.

Após cada bloco da imagem actual ter sido codificado utilizando ambas as imagens de referência, há que escolher qual a imagem a utilizar na codificação do bloco. Esta escolha é efectuada mais uma vez em função de um compromisso entre a qualidade obtida (definida pelo MAE do bloco reconstruído) e a quantidade de informação necessária para a codificação do bloco actual, utilizando cada imagem de referência.

Devido à estrutura do codificador utilizado, não é possível à partida ter uma ideia exacta do número de bits que resultam da utilização de cada uma das transformações estudadas, pelo que mais uma vez teve de ser utilizado um esquema de decisão baseado em regras definidas heuristicamente.

Para cada bloco inicial, de dimensão 32×32, há que considerar dois factores na estimação da quantidade de informação necessária para a sua codificação: o número de vezes que o bloco foi segmentado e o tipo de transformação utilizada para a codificação de cada sub-bloco. Cada vez que um bloco é segmentado, quatro novos blocos têm de ser codificados, o que aumenta a quantidade de informação utilizada. Por outro lado, a utilização das diferentes transformações implica a transmissão de um diferente número de parâmetros para cada uma, como já foi discutido no capítulo anterior.

Neste trabalho, concentrámo-nos no aspecto da codificação das sequências estereoscópicas genéricas. Os métodos apresentados foram desenvolvidos de modo a serem capazes de codificar eficientemente qualquer sequência de entrada, independentemente das características do sistema de aquisição utilizado. Isto significa que não é necessário um conhecimento prévio dos parâmetros geométricos e ópticos do sistema de câmaras utilizado.

Os resultados dos testes realizados foram avaliados em função da qualidade objectiva das sequências descodificadas e da taxa de informação produzida, para algumas sequências de teste, e são apresentados na secção seguinte.

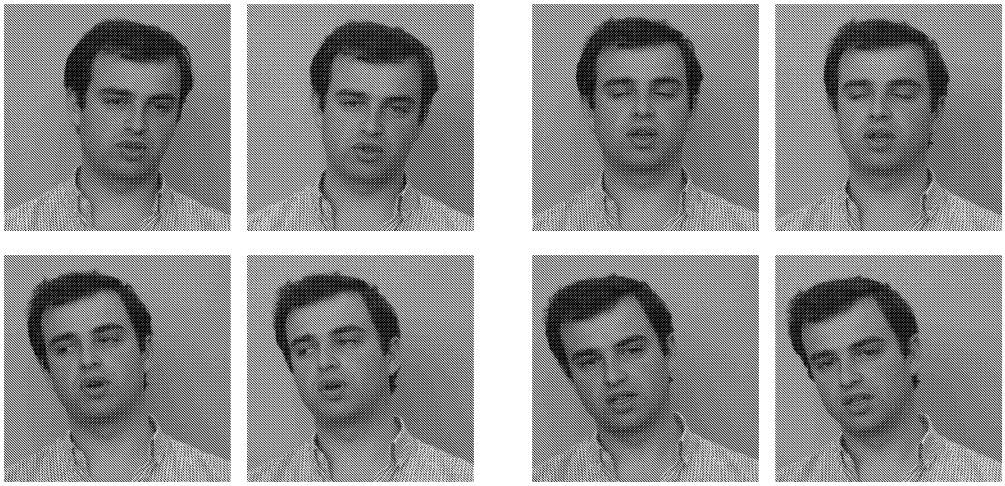


Figura 5.6: Imagens 0, 10, 20 e 31 das vistas esquerda e direita, da sequência estereoscópica *Sérgio*.

5.3.1 Resultados

Foram realizados testes de codificação de algumas sequências estereoscópicas de modo a avaliar os resultados obtidos por cada uma das configurações apresentadas anteriormente. Por outro lado, foram também comparados os resultados obtidos pelas técnicas de estimação e compensação de movimento com transformações geométricas, com e sem transformações no domínio da luminância.

Utilizaram-se mais uma vez sequências do tipo *cabeça e ombros*. Cada uma das vistas da sequência *Sérgio* estereoscópica é composta por 32 imagens, de 256×256 elementos cada, adquiridas com uma frequência temporal de 25 Hz. Algumas imagens desta sequência são apresentadas na figura 5.6.

Como já referimos, foram testadas quatro configurações para o codificador: a técnica *simulcast*, codificação com estimação e compensação de disparidade simples (DC), codificação com estimação e compensação de disparidade bidireccional (DC *Bidirec*) e codificação com estimação e compensação de disparidade e de movimento ($MC{+}DC$). Em todas elas, as imagens da sequência esquerda (sequência principal) são codificadas independentemente, pelo que a comparação dos resultados será feita pela avaliação da qualidade obtida na codificação da sequência direita (auxiliar).

Nos testes apresentados, foi codificada apenas uma imagem em cada duas, tendo as implicações desta opção sido já discutidas no capítulo anterior. Mais uma vez, a codificação de cada sequência utilizou apenas a informação das compensações de movimento e disparidade, não tendo sido utilizada a imagem do erro de predição.

A codificação *simulcast* da sequência direita pode ser encarada como uma codificação com as mesmas características das restantes, mas com a particularidade de utilizar apenas compensação de movimento. Neste caso, a primeira imagem da sequência tem de ser codificada de modo *intra*, já que não existe nenhuma imagem anterior para servir de referência. Esta solução foi também utilizada no caso da codificação *MC+DC*.

A codificação da primeira imagem como *intra* nestes dois casos, não torna a comparação com os resultados das técnicas *DC* e *DC Bidirec* desigual. Isto porque a primeira imagem que estas técnicas utilizam como referência, neste caso a primeira imagem da sequência esquerda, foi também codificada como *intra* e apresenta uma qualidade de reconstrução equivalente.

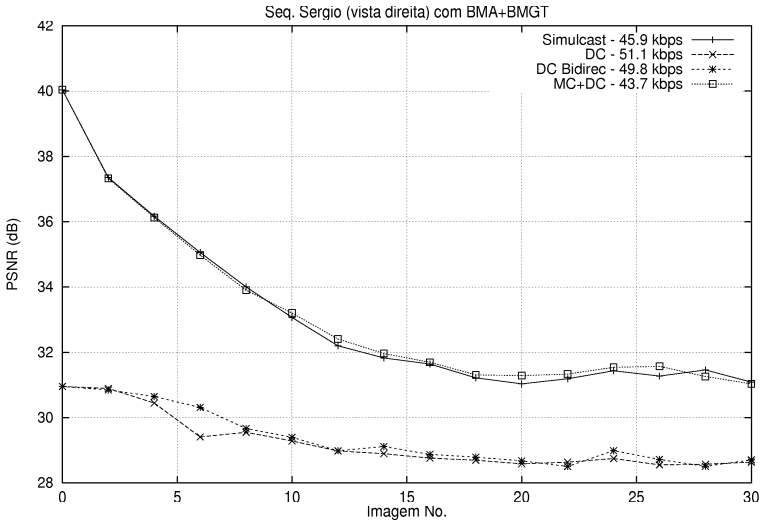


Figura 5.7: Codificação da sequência *Sérgio* estereoscópica com *BMA+BMGT*.

O gráfico da figura 5.7 apresenta os resultados obtidos pelas quatro técnicas de codificação consideradas, para as imagens da vista direita da sequência *Sérgio* estereoscópica. Neste teste foram utilizadas as transformações geométricas tradicionais, *BMA+BMGT*, com blocos de 32×32 (dimensão inicial).

Da análise deste gráfico podemos observar que as técnicas *simulcast* e *MC+DC* obtêm resultados muito superiores aos obtidos pelas técnicas que não utilizam compensação de movimento. A qualidade de codificação estabiliza em valores que são 2 dB superiores para os casos *simulcast* e *MC+DC*, o que demonstra que estas técnicas, por utilizarem compensação de movimento, possuem uma melhor capacidade de reconstrução das imagens da sequência direita.

Outra observação importante relaciona-se com a taxa de transmissão média utilizada por cada uma das técnicas utilizadas. As técnicas que utilizam apenas estimação e com-

pensação de disparidade utilizam uma taxa de transmissão muito superior. Este facto deve-se à utilização de um muito maior número de blocos segmentados, necessários para manter a qualidade de reconstrução em níveis aceitáveis.

Comparativamente, a técnica *DC Bidirec* permite obter uma qualidade de codificação que é marginalmente superior à obtida pela técnica *DC*. No entanto, a codificação com compensação de disparidade bidireccional utiliza uma taxa de transmissão média (49.8 kbps) que é menor que a obtida com compensação de disparidade simples (51.1 kbps).

Por outro lado, a utilização de compensação de movimento e disparidade (técnica *MC+DC*) permite obter resultados apenas ligeiramente superiores aos da técnica *simulcast*. Estes resultados são no entanto conseguidos utilizando uma taxa de transmissão média bastante inferior (43.7 kbps em vez de 45.9 kbps).

De notar que nos casos das técnicas *DC Bidirec* e *MC+DC* tem de ser transmitida informação que identifica a imagem de referência utilizada para cada bloco. Para o caso de duas imagens de referência, isto implica um bit extra por cada bloco da imagem a codificar.

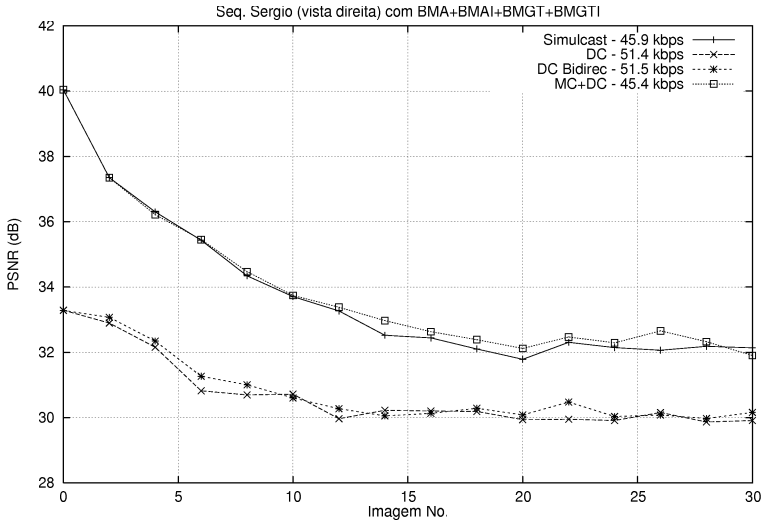


Figura 5.8: Codificação da sequência *Sérgio* estereoscópica com *BMA+BMAI+BMGT +BMGTI*.

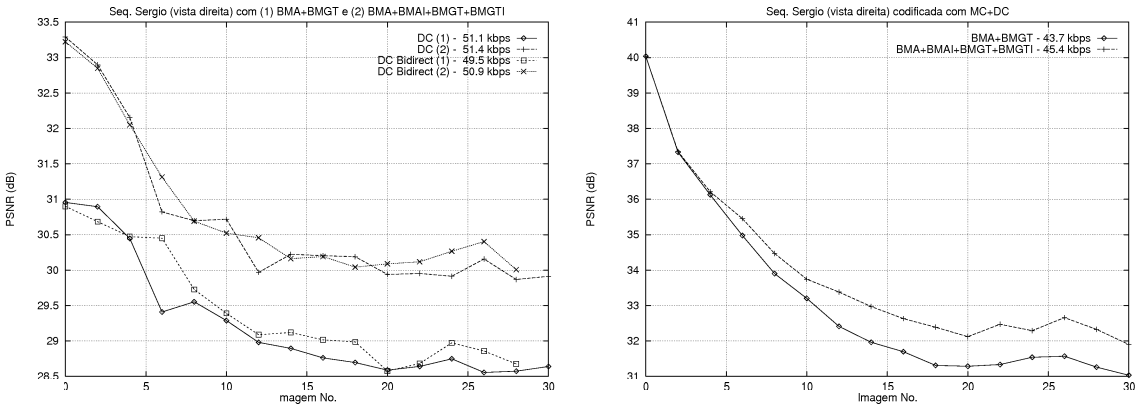
O gráfico da figura 5.8 representa os resultados obtidos com as mesmas técnicas e para a mesma sequência, só que desta vez utilizando também as novas transformações de luminância, *BMAI* e *BMGTI*.

Neste caso, os resultados comparativos entre as quatro técnicas de codificação estereoscópica mantêm-se coerentes com os obtidos para a codificação com *BMA+BMGT*. A técnica *MC+DC* é ligeiramente melhor que a codificação independente da sequência di-

reita, conseguindo para além disso uma redução na taxa de transmissão média. Por outro lado, a utilização de compensação bidireccional de disparidade é praticamente equivalente à compensação simples de disparidade em termos da qualidade de codificação.

Não se verifica neste caso uma melhoria da taxa de informação para a compensação bidirecional de disparidade. Isto significa que a compensação simples de disparidade utilizando todas as transformações é mais eficiente, de tal modo que os seus resultados se aproximam dos obtidos com a utilização de duas imagens de referência.

Globalmente, continua a verificar-se que a utilização apenas de compensação de disparidade não permite obter resultados tão bons como as técnicas que utilizam compensação de movimento, verificando-se uma diferença de cerca de 2 dB entre os resultados dos dois grupos de técnicas.



(a) (b)

Figura 5.9: Comparação dos resultados da codificação da sequência *Sérgio* estereoscópica, com as transformações *BMA+BMGT* e *BMA+BMAI+BMGT+BMGTI*

Os gráficos da figura 5.9 fazem a comparação directa dos resultados obtidos por cada uma das técnica de codificação estereoscópica compatíveis, utilizando apenas transformações geométricas espaciais (*BMA+BMGT*) ou também as transformações da luminância propostas (*BMA+BMAI+BMGT+BMGTI*).

O gráfico da figura 5.9a) apresenta os resultados das técnicas que utilizam apenas compensação de disparidade. Podemos assim verificar uma melhoria de cerca de 1 dB na qualidade obtida pela aplicação das transformações de luminância à compensação da disparidade entre as imagens das duas vistas.

Na figura 5.9b) são comparados os resultados obtidos na codificação com compensação de movimento e disparidade (*MC+DC*). Mais uma vez, verificamos que a utilização das

transformações no domínio da luminância, propostas neste trabalho, permite uma melhoria da qualidade da sequência codificada próxima de 1 dB.

Esta melhoria resulta não só de uma melhor compensação da disparidade, como acontecia no caso anterior, mas também de uma melhor eficácia da compensação de movimento por parte das novas transformações, já verificada no capítulo 4. O aumento de qualidade associado à utilização das novas transformações é acompanhado por um pequeno aumento da taxa de informação utilizada.

Capítulo 6

Conclusões e Trabalho Futuro

Neste trabalho estudámos a aplicação de transformações geométricas e no domínio da luminância na estimação e compensação de movimento, em sistemas de codificação de sequências de imagens. Foi também estudada a aplicação das mesmas transformadas na codificação de sequências estereoscópicas, nomeadamente nos processos de estimação e compensação de movimento e de disparidade.

A utilização de transformações geométricas na estimação e compensação de movimento é um processo bem conhecido. Para além das técnicas genéricas que têm vindo a ser propostas na literatura, algumas das quais foram apresentadas brevemente no capítulo 3, um esquema de estimação e compensação de movimento com transformações geométricas faz parte da norma 4 do grupo MPEG.

As transformações geométricas têm a capacidade de compensar um maior número de tipos de alterações entre as imagens de uma sequência. Isto explica o seu melhor desempenho, face às técnicas de estimação e compensação de movimento baseadas apenas na translação de blocos, designadas normalmente por técnicas *BMA*.

Esta melhoria verifica-se apesar destas técnicas utilizarem inevitavelmente um maior número de parâmetros de estimação e compensação de movimento. A principal desvantagem destes métodos está relacionada com a sua maior complexidade computacional, relativamente às técnicas *BMA*. Esta complexidade pode no entanto ser atenuada com a utilização de algoritmos optimizados.

Do mesmo modo, são há muito conhecidas as vantagens da utilização das técnicas de estimação e compensação de movimento associadas a esquemas de segmentação de blocos em outros de menores dimensões. Um dos modos mais divulgadas de segmentação de blocos são as *árvores quaternárias*, em que cada bloco é dividido em quatro blocos

de iguais dimensões. A vantagem desta representação está associada à eficiência da sua codificação.

A utilização de blocos de diferentes dimensões permite codificar de forma mais eficiente grandes áreas, constituídas por pixels com movimento idêntico, com o uso de blocos de grandes dimensões, poupando assim bits que podem ser utilizados na codificação das áreas da imagem com mais detalhe, ou com movimentos mais difíceis de compensar, nas quais são utilizados blocos de dimensões mais reduzidas.

6.1 Estimação e compensação de movimento com transformações da luminância

A contribuição deste trabalho centrou-se no estudo de um novo tipo de transformações, que são aplicadas no domínio da luminância. Estas operações transformam os elementos de um bloco de uma imagem de referência, de modo a estimar um bloco da imagem que se pretende codificar.

Esta transformação abrange os valores de todos os elementos do bloco de referência, que são multiplicados por um factor de escala constante, c, e somados a outro valor constante, b. A determinação destes valores utiliza uma técnica de estimação por mínimos quadrados, de modo que a aplicação da transformação, correspondente aos valores calculados, minimize o erro médio quadrático (MSE) entre os blocos em causa.

De modo a minimizar a carga computacional associada a este processo, foram determinadas as expressões algébricas que permitem calcular directamente os valores dos factores de compensação c e b, a partir dos valores de todos os elementos dos blocos de referência e do bloco a estimar.

Esta nova técnica de transformação dos valores de luminância foi associada a duas técnicas tradicionais de estimação e compensação de movimento: a técnica BMA e a técnica $BMGT$, originado dois novos métodos que foram designados por $BMAI$ e $BMGTI$. Estes métodos combinam a nova transformação com compensação de movimento de translação e compensação de movimento com transformações bilineares.

Foi desenvolvido um sistema de codificação de sequências de imagens capaz de utilizar estas quatro técnicas, escolhendo para cada bloco a mais adequada. Para além disso, o codificador desenvolvido utiliza também segmentação do blocos, de modo a optimizar a dimensão do bloco de referência utilizado na codificação de cada bloco da imagem actual.

Este sistema foi utilizado na codificação de várias sequências de imagens, tendo sido analisados os resultados obtidos em termos da qualidade objectiva da sequência descodificada ($PSNR$) e da taxa de transmissão necessária (em $kbps$). Da análise destes resultados, apresentada no capítulo 4, foi possível retirar um conjunto interessante de conclusões:

- a utilização de transformações bilineares na compensação de movimento permite um ganho na qualidade das imagens codificadas, devido à sua capacidade para compensar rotações, inclinações segundo um dos eixos e situações de zoom, entre outras;
- a utilização da compensação de luminância associada à técnica *BMA* (técnica *BMAI*), permite a obtenção de melhores resultados do que a técnica *BMGT*, pois alia a compensação de movimento de translação à compensação da luminância dos elementos da imagem codificada;
- se a compensação da luminância for utilizada juntamente com a compensação de movimento com transformações geométricas (técnica *BMGTI*), os resultados obtidos são ainda melhores do que no caso anterior;
- os melhores resultados são obtidos pela combinação das quatro técnicas de codificação, *BMA*, *BMAI*, *BMGT* e *BMGTI*, de um modo que permite ao codificador seleccionar qual a transformação mais adequada para cada bloco a codificar, baseando-se num compromisso entre a qualidade objectiva do bloco codificado e a taxa de transmissão utilizada por cada transformação;
- apesar da utilização de blocos de pequenas dimensões permitir uma melhor qualidade das imagens descodificadas, o aumento da taxa de transmissão que esta opção envolve não é justificado. A utilização de blocos de maiores dimensões é mais favorável, pois permite um melhor compromisso entre a qualidade das imagens descodificadas e a taxa de transmissão necessária;
- da observação anterior concluímos que a segmentação dos blocos utilizados é um factor fundamental para o desempenho dos vários métodos de codificação estudados.

Para além da análise apresentada dos resultados objectivos, foi também realizada uma análise subjectiva da qualidade das sequências descodificadas. Nesta análise foi possível observar a capacidade que as novas transformações apresentam na codificação eficiente das zonas da imagem que correspondem a novas áreas recém expostas. Concluímos assim que as transformações propostas são capazes de compensar eficientemente fenómenos de

alterações espaciais e temporais da luminância e oclusão entre objectos da cena. Estes fenómenos são difíceis de compensar com transformações geométricas e principalmente com a técnica *BMA*.

Foi também efectuado um estudo dos efeitos da variação de alguns parâmetros importantes de codificação na qualidade final.

Foi possível retirar algumas conclusões desse estudo, nomeadamente:

- os efeitos da utilização de blocos de pequenas dimensões é mais notório quando se utilizam as novas transformações, relativamente às técnicas que utilizam apenas transformações espaciais. Esta observação pode ser justificada por duas razões: primeiro, a utilização de blocos com um menor número de pixels melhora a estimação dos factores de compensação da luminância, e em segundo lugar, as deformações geométricas de blocos de pequenas dimensões representam de forma menos precisa as alterações de cada um dos seus elementos;

- as novas transformações permitem obter melhores resultados para uma grande gama de variação da taxa de transmissão utilizada. Para taxas de transmissão elevadas, nota-se uma aproximação entre a qualidade obtida pela técnica *BMA+BMGTI* e a combinação das quatro técnicas investigadas;

- o aumento da janela de pesquisa utilizada na determinação dos vectores de compensação de movimento só é benéfico quando não se ultrapassa o deslocamento máximo dos elementos da imagem;

- os valores dos passos de quantificação utilizados na transmissão dos valores de compensação de luminância têm um efeito directo na qualidade alcançada. Verificou-se experimentalmente que estes valores devem ser aproximadamente 0.05 para o factor de multiplicação c e entre 1 e 2 para o factor b.

6.2 Codificação de sequências estereoscópicas

A aplicação das transformações desenvolvidas à codificação de sequências estereoscópicas foi feita de modo a estudar dois pontos fundamentais: a melhoria na qualidade das sequências descodificadas, introduzida pela utilização dessas técnicas e as diferenças de desempenho entre quatro esquemas de codificação de sequências estereoscópicas.

Dos resultados obtidos foi possível concluir que a utilização das transformações de luminância na estimação e compensação de disparidade permite atingir novamente ganhos

na qualidade de codificação (que atingem cerca de 1 dB), face à utilização das técnicas baseadas em transformações espaciais geométricas.

Em relação às diferenças entre as técnicas de codificação estereoscópicas, foram analisadas: a codificação *simulcast*, codificação com estimação e compensação normais de disparidade (*DC*), codificação com estimação e compensação bidireccionais de disparidade (*DC Bidirec*) e finalmente codificação com estimação e compensação de disparidade e de movimento (*MC+DC*).

Os resultados obtidos permitem concluir que a utilização exclusiva de compensação de disparidade (em que se efectua a codificação de cada imagem da sequência auxiliar apenas à custa de uma ou mais imagens da sequência principal), é inferior às técnicas que utilizam também estimação e compensação de movimento.

Isto acontece principalmente em sequências em que a disparidade entre as imagens das duas vistas é elevada. No entanto, e de um modo geral, a utilização de imagens de referência pertencentes também à sequência auxiliar, permite uma melhor estimação da imagem actual.

Dentro das técnicas que utilizam apenas estimação e compensação de disparidade, a utilização de técnicas bidireccionais é vantajosa face à utilização de apenas uma imagem de referência.

A técnica de codificação compatível de sequências estereoscópicas que combina a compensação de disparidade com a compensação de movimento mostrou ser a melhor das quatro, com ganhos ao nível da qualidade objectiva e da taxa de transmissão, relativamente à segunda melhor técnica, que no caso dos testes realizados mostrou ser a codificação *simulcast*.

6.3 Avaliação global das técnicas propostas

A utilização de novos esquemas de estimação e compensação de movimento na codificação de sequências vídeo é um assunto bastante abordado na literatura científica dedicada a este tema. É no entanto nossa convicção que os métodos propostos neste trabalho apresentam uma abordagem original a este problema, tendo sido realizado um estudo útil para a avaliação dos resultados que as técnicas propostas permitem obter.

Os resultados deste estudo demonstraram que a utilização destas técnicas é viável, permitindo melhorias, por vezes significativas, na qualidade das sequências codificadas. Este sucesso não deve no entanto impedir a realização de algumas observações sobre o

desempenho das técnicas propostas.

A melhoria de qualidade das imagens codificadas, apesar de significativas, são conseguidas através de um aumento da complexidade computacional dos processos de estimação e compensação de movimento e de disparidade.

O método de codificação utilizado, que selecciona a transformação em função da qualidade que obtém para cada bloco, implica que cada bloco seja codificado várias vezes (uma por cada transformação a testar). Do mesmo modo, a segmentação de um bloco só é feita após uma codificação de teste, que permite avaliar os ganhos conseguidos. Todas estas codificações adicionais consomem tempo de computação, que é o preço a pagar pelo aumento de eficiência conseguido.

No entanto, este compromisso entre o acréscimo da complexidade e a melhoria da qualidade, está subjacente à grande maioria dos desenvolvimentos verificados nos sistemas de codificação de sequências vídeo. Exemplos desta situação vão desde a utilização de vectores de movimento com precisão fraccionária até à utilização de estimação e compensação bidireccional de movimento, técnicas cuja complexidade adicional foi facilmente aceite devido às melhorias de qualidade que introduzem no resultado final.

Apesar de ter sido demonstrada uma boa capacidade de codificação utilizando apenas os dados de estimação e compensação de movimento, a utilização das técnicas estudadas neste trabalho em aplicações reais poderá implicar a transmissão ocasional da imagem de erro, de modo a atenuar a perda de qualidade da sequência descodificada.

Ao contrário do que acontecia com os métodos puramente espaciais de estimação e compensação de movimento, as transformações de luminância não podem ser directamente extrapoladas para as imagens de crominância, em sequências coloridas.

6.4 Trabalho futuro

Ao longo deste trabalho, por cada conclusão alcançada, várias linhas de investigação foram aparecendo no horizonte, correspondentes a vários caminhos possíveis de explorar para lá do âmbito desta tese.

Algumas dessas hipóteses têm alguma relevância num contexto próximo do método de codificação utilizado. Neste contexto, seria por exemplo interessante o estudo de esquemas de decisão que melhorassem a escolha da técnica a utilizar na codificação de cada bloco da imagem. Estes esquemas iriam substituir as regras heurísticas desenvolvidas neste

trabalho, cuja optimização dependeu da nossa experiência e sensibilidade para o problema.

Outro estudo interessante, relacionado intimamente com este problema, seria determinar as diferenças existentes entre cada método proposto, ao nível da taxa de informação utilizada para cada bloco. É nossa convicção que uma estimativa precisa destes valores não é facilmente alcançável, devido à estrutura da codificação utilizada, que envolve, por exemplo, uma codificação aritmética dinâmica dos parâmetros de compensação de movimento. No entanto, este estudo poderia ajudar nas decisões a efectuar na codificação de cada bloco.

Como já referimos, estes dois pontos de investigação permitiriam, de algum modo, melhorar principalmente o desempenho do sistema de codificação utilizado. Outras possibilidades de investigação relacionam-se com a aplicabilidade das transformações propostas às normas de codificação de vídeo conhecidas, nomeadamente as normas *MPEG* e *H.26X*.

Deste modo, seria interessante estudar quais as implicações que as melhorias nos processos de estimação e compensação de movimento, verificadas neste trabalho, trariam aos resultados obtidos por um codificador tradicional, baseado em *BMA* e na codificação do erro de predição com DCT.

De um modo mais abrangente, parece-nos que a utilização de transformações no domínio da luminância dos elementos das imagens, como modo de melhorar a qualidade da estimação da imagem a codificar, é uma ideia que faz sentido mesmo em técnicas que não utilizam o esquema de codificação apresentado.

Poderiam também ser estudadas formas de aplicação das técnicas propostas em sequências de imagens coloridas.

Bibliografia

[1] *CCIR Rec. 601-2: Encoding parameters of digital television for studios*, 1990.

[2] A. K. Jain, *Fundamentals of Digital Image Processing.* Information and System Sciences Series, Prentice Hall, 1989.

[3] N. Abramson, *Information theory and coding.* McGraw-Hill, 1963.

[4] D. A. Huffman, "A method for construction of minimum redundancy codes," *Proceedings Institute of Electrical and Radio Engineers*, vol. 40, pp. 1098–1101, Sept 1952.

[5] T. C. Bell, J. G. Cleary, and I. H. Witten, *Text Compression.* Prentice-Hall, 1990.

[6] N. Jayant, J. Johnston, and R. Safranek, *Handbook of visual communications*, ch. 3 - Image compression based on models of human vision, pp. 73 – 125. Academic Press, 1995.

[7] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards - Algorithms and Architectures.* Kluwer Academic Publishers, 1995.

[8] W. B. Pennebaker and J. L. Mitchell, *JPEG - Still Image Data Compression Standard.* Chapman and Hall, International Thomson Publishing, 1993.

[9] N. Jayant, "Signal compression: Technology targets and research directions," *IEEE Journal on Selected Areas in Communications*, vol. 10, pp. 796–818, June 1992.

[10] J. Mitchell, W. Pennebaker, C. Fogg, and D. LeGall, *MPEG Video Compression Standard.* Digital Multimedia Standards Series, Chapman & Hall, 1996.

[11] T. Naveen and J. W. Woods, *Handbook of visual communications*, ch. 8 - Subband and Wavelet Filters for High-Definition Video Compression, pp. 265–298. Academic Press, 1995.

[12] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, pp. 205–220, April 1992.

[13] M. Vetterli and C. Herley, "Wavelets and filter banks: Theory and design," *IEEE Transactions on Signal Processing*, vol. 40, pp. 2207–2232, September 1992.

[14] A. E. Jacquin, "Fractal image coding: A review," *Proceedings of the IEEE*, vol. 81, pp. 1451–1465, October 1993.

[15] M. F. Barnsley, *Fractals Everywhere.* Academic Press, New York, 1988.

[16] A. Wright, "Fractals transform image compression," *Electronics World + Wireless World*, March 1992.

[17] A. Gersho and R. M. Gray, *Vector quantization and signal compression.* Kluwwer Academic Publishers, Boston, 1992.

[18] A. Gersho, S. Gupta, and S.-W. Wu, *Handbook of visual communications*, ch. 6 - Vector Quantization Techniques in Image Compression, pp. 189–222. Academic Press, 1995.

[19] H. Li, A. Lundmark, and R. Forchheimer, "Image sequence coding at very low bitrates: A review," *IEEE Transactions on Image Processing*, vol. 3, pp. 589–609, September 1994.

[20] T. Koga, K. Linuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," *NTC 81 Proc.*, pp. G5.3.1–5, New Orleans, LA, Dezembro 1981.

[21] A. Puri, H. M. Hang, and D. L. Schlling, "An efficient block-matching algorithm for motion compensated coding," *Proceedings of the ICASSP 87 Conference*, pp. 25.4.1–25.4.4, 1987.

[22] P. Strobach, "Tree-structured scene adaptive coder," *IEEE Transactions Communications*, vol. 38, pp. 477–486, April 1990.

[23] Y. Cohen, M. Landy, and M. Pavel, "Hierarchical coding of binary images," *IEEE Trans. on pattern analysis and machine intelligence*, vol. PAMI-7, pp. 284–298, May 1985.

[24] E. Shusterman and M. Feder, "Image compression via improved quadtree decomposition algorithms," *IEEE transactions on image processing*, vol. 3, pp. 207–215, March 1994.

[25] G. Schuster and A. Katsaggelos, "Optimal decomposition for quad-trees with leef dependencies," *Proceedings of SPIE*, vol. 3024, pp. 59–70, January 1997.

[26] M. Ghanbari, S. de Faria, I. N. Goh, and K. T. Kan, "Motion compensation for very low bit-rate video," *Signal Processing: Image Communication*, no. 7, pp. 567–580, 1995.

[27] G. Abrantes and F. Pereira, "Mpeg-4 facial animation technology: survey, implementation, and results," *IEEE Transactions on circuits and systems for video technology*, vol. 9, March 1999.

[28] K. Aizawa, *Handbook of visual communications*, ch. 10 - Model-based Coding, pp. 341–364. Academic Press, 1995.

[29] H. G. Musmann, M. Hotter, and J. Ostermann, "Object-oriented analysis-synthesis coding of moving images," *Signal Processing: Image Communications*, vol. 1, pp. 117–138, 1989.

[30] M. Hotter, "Optimization and efficiency of an object-oriented analysis-synthesis coder," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 4, pp. 181–194, April 1992.

[31] H.261, *ITU-T Recommendation H.261 - Line transmission of non-telefone signals. Video-codec for audiovisual services at p×64kbits.* ITU-T, March 1993.

[32] H.320, *ITU-T Recommendation H.320 - Line transmission of non-telefone signals. Narrow-band visual telefone systems and terminal equipment.* ITU-T, March 1993.

[33] H.263, *ITU-T Recommendation H.263 - Video coding for low bit rate communication.* ITU-T, February 1998.

[34] MPEG-1, *ISO/IEC JTC1 CD 11172. Coding of moving pictures and associated audio for digital storage media at up to 1.5 Mbit/s.* International Organization for Standardization (ISO), 1992.

[35] MPEG-2, *ISO/IEC JTC1 CD 13818. Generic coding of moving pictures and associated audio.* International Organization for Standardization (ISO), 1994.

[36] B. Haskell, A. Puri, and A. Netravali, *Digital video: an introduction to MPEG-2.* Digital Multimedia Standards Series, Chapman & Hall, 1997.

[37] MPEG-4, *ISO/IEC JTC/SC29/WG11 N3342. Overview of the MPEG-4 Standard.* International Organization for Standardization (ISO), Noordwijkerhout, March 2000. http://drogo.cselt.stet.it/mpeg/standards/mpeg-4/mpeg-4.htm.

[38] F. Pereira, "Mpeg-4: Why, what, how and when?," *invited paper for the Tutorial Issue on the MPEG-4 Standard, Signal Processing: Image Communication*, vol. 15, January 2000. http://www.img.lx.it.pt/∼fp/artigo96/artigo96.htm.

[39] R. Koenen, F. Pereira, and L. Chirglione, "Mpeg-4: context and objectives," *Special Issue on MPEG-4 of Signal Processing: Image Communication*, vol. 9, May 1997.

[40] L. Chiariglione, "The mpeg-4 standard," *Journal of the China Institute of Communications*, September 1998.
http://drogo.cselt.it/ufv/leonardo/paper/china98/china98.html.

[41] F. Pereira, "Mpeg-7: a standard for describing audiovisual information," *invited talk at Colloquium on Multimedia databases and MPEG-7, London, UK*, January 1999. http://www.img.lx.it.pt/∼fp/artigo93/artigo93.htm.

[42] MPEG-7, *ISO/IEC JTC1/SC29/WG11 N3349. Overview of the MPEG-7 Standard.* International Organization for Standardization (ISO), Noordwijkerhout, March 2000. http://drogo.cselt.it/ufv/leonardo/mpeg/standards/mpeg-7/mpeg-7.htm.

[43] V. Seferidis and M. Ghanbari, "General approach to block-matching motion estimation," *Journal of Optical Engineering*, vol. 32, pp. 1464–1474, July 1993.

[44] S. M. de Faria, *Very Low Bit Rate Video Coding using Geometric Transform Motion Compensation.* PhD thesis, Department of Electronic Systems Engineering - University of Essex, June 1996.

[45] G. Wolberg, *Digital Image Warping.* Los Alamitos, California, USA: IEEE Computer Society Press, 1990.

[46] A. N. Netravalli and J. B. Robbins, "Motion-compensated television coding: Part 1," *Bell System Technology Journal*, pp. 631–670, Março 1979.

[47] Y. Nakaya and H. Harashima, "Motion compensation based on spatial transformations," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, pp. 339–356, June 1994.

[48] M. Ghanbari, S. de Faria, I. N. Goh, and K. T. Tan, "Motion compensation for very low bit-rate video," *Signal Processing: Image Communication*, vol. 7, pp. 567–580, 1995.

[49] Y. Nakaya and H. Harashima, "An iterative motion estimation method using triangular patches for motion compensation," *Proc. SPIE Visual Comm. and Image Processing*, pp. 546–557, November 1991.

[50] C.-L. Huang and C.-Y. Hsu, "A new motion compensation method for image sequence coding using hierarchical grid interpolation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, pp. 42–52, february 1994.

[51] G. J. Sullivan and R. L. Baker, "Motion compensation for video compression using control grid interpolation," *Proc. ICASSP'91, Toronto, Canada*, pp. 2713–2716, May 1991.

[52] Y. Yokoyama, Y. Miyamoto, and M. Ohta, "Very low bit rate video coding using arbitrarily shaped region-based motion compensation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, pp. 500–507, December 1995.

[53] V. A. Patel, *Numerical Analysis.* Harcourt Brace College Publishers, 1994.

[54] K. Kamikura, H. Watanabe, H. Jozawa, H. Kotera, and S. Ichinose, "Global brightness-variation compensation for video coding," *IEEE Transactions on circuits and systems for video technology*, vol. 8, pp. 988–1000, December 1998.

[55] N. Rodrigues, V. Silva, and S. de Faria, "General geometric transform for video coding," *Proc. II Conferência de Telecomunicações - ConfTele'99*, pp. 708–711, April 1999.

[56] M. Siegel and S. Nagata, "Just enough reality: confortable 3-d viewing via microstereopsis," *IEEE Transactions on circuits and systems for video technology*, vol. 10, pp. 387–396, April 2000.

[57] R. Börner, B. Duckstein, O. Machui, H. Röder, T. Sinnig, and T. Sikora, "A family of single-user autostereoscopic displays with head-traking capabilities," *IEEE Transactions on circuits and systems for video technology*, vol. 10, pp. 234–243, March 2000.

[58] K. Hopf, "An autostereoscopic display providing confortable viewing conditions and a high degree of telepresence," *IEEE Transactions on circuits and systems for video technology*, vol. 10, pp. 359–365, April 2000.

[59] I. Dinstein, G. Guy, J. Rabany, J. Tzelgov, and A. Henik, "On stereo image coding," *Proceedings of the international conference on pattern recognition*, pp. 357–359, November 1988.

[60] B. T. Tseng and D. Anastassiou, "Compatible video coding of stereoscopic sequences using mpeg-2's scalability and interlaced structure," *International Workshop on HDTV 94, Torino - Italy*, October 1994.

[61] B. T. Tseng and D. Anastassiou, "Multi-viewpoint video coding with mpeg-2 compatibility," *IEEE Transactions on circuits and systems for video technology*, vol. 6, August 1996.

[62] F. Labonté, C. T. L. Dinh, J. Faubert, and P. Cohen, "Spatiotemporal spectral coding of stereo image sequences," *IEEE Transactions on circuits and systems for video technology*, vol. 9, pp. 144–155, February 1999.

[63] P. D. Gunatilake, M. W. Siegel, and A. G. Jordan, "Compression of stereo video streams," *SMPTE International workshop on HDTV 93, Ottawa - Canada*, October 1993.

[64] S. Sethuraman, *Stereoscopic image sequence compression using multiresolution and quadtree decomposition based disparity- and motion-adaptive segmentation.* PhD thesis, Dept. of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh PA, July 1996.

[65] M. Siegel, S. Sethuraman, J. McVeigh, and A. Jordan, "Compression and interpolation of 3d-stereoscopic and multi-view video," *Proc. SPIE conference, San Jose - California*, vol. 3012, pp. 227–238, February 1997.

[66] M. Moellenhoff and M. Maier, "Dct transform coding for stereo images for multimedia applications," *IEEE Transactions on industrial electronics*, vol. 45, pp. 38–43, February 1998.

[67] N. Grammalidis, S. Malassiotis, D. Tzovaras, and M. G. Strintzis, "Stereo image sequence coding based on three-dimensional motion estimation and compensation," *Signal Processing: Image Communication*, vol. 7, pp. 129–145, August 1995.

[68] D. Tzovaras, N. Grammalidis, and M. G. Strintzis, "Object-based coding of stereo image sequences using joint 3-d motion/disparity compensation," *IEEE Transactions on circuits and systems for video technology*, vol. 7, pp. 312–327, April 1997.