

PEDRO ALEXANDRE FERREIRA T. PEIXOTO

Técnicas de Codificação de
Vídeo para Baixas Cadências

DEEC
FEUP
1995



Faculdade de Engenharia da Universidade do Porto
Mestrado em Engenharia Electrotécnica e de Computadores

Técnicas de Codificação de Vídeo

Para Baixas Cadências

Pedro Alexandre Ferreira Teixeira Peixoto

1995

Pedro Alexandre Ferreira Teixeira Peixoto

Técnicas de Codificação de Vídeo

Para Baixas Cadências

DEEC

FEUP

Dissertação submetida para satisfação parcial
dos requisitos do grau de mestre em
Engenharia Electrotécnica e de Computadores
(Área de especialização de Informática Industrial)

Dissertação realizada sob a supervisão de
Professor Doutor Luís António Pereira de Meneses Corte-Real
do Departamento de Engenharia Electrotécnica e de Computadores
da Faculdade de Engenharia da Universidade do Porto

Porto, Outubro de 1995

cat

GEL 31042 / PEI / TEC

UNIVERSIDADE DO PORTO
FACULDADE DE ENGENHARIA
BIBLIOTECA
N.º 21628
CDU
DATA 6 / 5 / 1996
NN 4651

Resumo

Neste trabalho são descritas várias técnicas de compressão de vídeo de baixa complexidade para muito baixas cadências. Apresenta-se um sistema de codificação baseado na segmentação de imagens diferença, em blocos de dimensão e formas variáveis, e num módulo de quantificação vectorial, o qual foi o ponto de partida para o desenvolvimento de técnicas de codificação com vista à melhoria do seu desempenho. A introdução de técnicas de codificação de vídeo e respectivos algoritmos teve em especial atenção a complexidade de implementação, a qualidade da imagem codificada e a taxa de compressão obtida. São descritas quatro técnicas de codificação: a codificação por padrões visuais, compensação de movimento, identificação de fundos e codificação da cor. Os módulos de codificação referidos permitem, devido à independência da sua implementação, a construção de um sistema de compressão flexível que atinja os objectivos de complexidade, qualidade e taxa de compressão pretendidos.

Abstract

In this work several low complexity techniques for video compression at low bitrates are described. A compression system is presented based on the segmentation of the difference images, in variable dimension and shape blocks, and on a vector quantification module, which was the starting point for the development of coding techniques in order to improve the system's overall performance. The introduction of techniques for video coding and corresponding algorithms had in special consideration the complexity of the implementation, the quality of the coded image and the compression rate. Four coding techniques are described: visual pattern image coding, motion compensation, background identification and colour coding. The coding modules described allow, due to their implementation independence, the design of a flexible compression system that will attain the complexity, quality and compression rate goals.

2000

The year 2000 was a year of significant change for the company. In January, we launched our new product line, which was well-received by the market. In March, we announced our expansion into new markets, which was a major milestone for the company. In May, we held our annual shareholders' meeting, where we discussed our performance and future plans. In July, we launched our new website, which was a major improvement over our old one. In September, we announced our new CEO, who was a highly respected leader in the industry. In November, we held our annual conference, where we discussed our performance and future plans. In December, we announced our new financial results, which were a major improvement over our old ones.

2000 was a year of significant change for the company.

Agradecimentos

É com grande prazer que agradeço ao meu supervisor, Professor Luís Corte-Real, pela sua paciência, orientação e entusiasmo. Foi para mim uma honra trabalhar sob a sua supervisão exemplar.

Estou também muito grato à Engenheira Isabel Martins pelo seu apoio e encorajamento.

Quero também agradecer a todos os meus colegas do INESC, em particular, ao Grupo de Codificação Áudio e Vídeo, por toda a sua ajuda, apoio moral e amizade.

Agradecimentos são também devidos à JNICT pelo apoio financeiro prestado durante a execução deste trabalho.

Agradeço, ainda, a todos os meus amigos que me suportaram durante o tempo de execução deste trabalho. O meu muito especial agradecimento à amiga e colega Cristina Russo, cuja amizade, incentivo, boa-disposição e ajuda prestada foram determinantes para a conclusão desta dissertação.

Finalmente, agradeço aos meus pais e à minha irmã por todo o apoio que me deram.

Índice

Resumo 3

Abstract 5

Agradecimentos 7

Índice das figuras 13

Índice das tabelas 17

Capítulo 1 Introdução 19

1.1 Motivação do autor 21

1.2 Estrutura da tese 22

Capítulo 2 Compressão de Imagem 23

- 2.1 Introdução 23
- 2.2 Teoria da informação 25
- 2.3 Técnicas de compressão de imagem 28
 - 2.3.1 Codificação por predição 28
 - 2.3.2 Codificação por transformada 31
 - 2.3.3 Quantificação vectorial 36
 - 2.3.4 Compensação de movimento 43
 - 2.3.4.1 Técnica de emparelhamento de blocos 46
 - 2.3.4.2 Algoritmos recursivos 49

Capítulo 3 Compressão de Vídeo de Baixa Complexidade Para Baixas Cadências 51

- 3.1 Introdução 51
- 3.2 Segmentação de imagem 57
- 3.3 Módulo de quantificação vectorial 63

Capítulo 4 Técnicas de Codificação de Vídeo Para Baixas Cadências 67

- 4.1 Introdução 67
- 4.2 Codificação de imagem por padrões visuais 68
 - 4.2.1 Introdução 68
 - 4.2.2 Dicionário de padrões 71

4.2.3	Algoritmo de codificação	75
4.2.4	Algoritmo de decodificação	78
4.2.5	Análise de complexidade	78
4.2.6	Aplicação do algoritmo a blocos de dimensão e forma variáveis	80
4.2.7	Implementação e análise dos resultados obtidos	82
4.2.7.1	Estudo da magnitude do gradiente	83
4.2.7.2	Codificação dos blocos de transição e uniformes	94
4.2.7.3	Resolução dos casos especiais	98
4.2.7.4	Conclusões	105
4.3	Compensação de movimento	106
4.3.1	Introdução	106
4.3.2	Implementação e análise dos resultados obtidos	110
4.4	Identificação de fundos	117
4.4.1	Introdução	117
4.4.2	Implementação e análise dos resultados obtidos	121
4.5	Codificação de sequências de imagens a cores	131

Capítulo 5 Conclusões 135

5.1	Conclusões	135
5.2	Perspectivas de trabalho futuro	138

Referências 141

Índice das figuras

Capítulo 2

- 2.1 - Sistema de predição básico
- 2.2 - Codificação por transformada unidimensional
- 2.3 - Representação geométrica da codificação por transformada a 3 dimensões
- 2.4 - Sistema de quantificação vectorial
- 2.5 - Exemplo de uma partição de $\mathbb{R}^2$
- 2.6 - Exemplo de regiões de Voronoi em $\mathbb{R}^2$
- 2.7 - Janela de procura
- 2.8 - Compensação de movimento em blocos de dimensão variável

Capítulo 3

- 3.1 - Sistema de codificação básico
- 3.2 - Sistema de videovigilância
- 3.3 - Formato QCIF
- 3.4 - Sistema de compressão de imagem
- 3.5 - Sistema de decodificação de imagem
- 3.6 - Segmentação de imagem em blocos de dimensão fixa
- 3.7 - Segmentação de imagem em regiões não regulares
- 3.8 - Segmentação estruturada em árvores quaternárias
- 3.9 - Segmentação não estruturada em um passo
- 3.10 - Segmentação não estruturada em múltiplos passos
- 3.11 - Exemplo de uma tabela de códigos estruturada em árvore
- 3.12 - Decimação dos blocos de maiores dimensões

Capítulo 4

- 4.1 - Geometria de visualização
- 4.2 - Cálculo das componentes direccionais do gradiente
- 4.3 - Padrões quadrados com as orientações de 90° , 45° , 0° e -45°
- 4.4 - As orientações principais

- 4.5 - Padrões rectangulares
- 4.6 - Resultados obtidos para blocos de dimensão e forma variáveis
- 4.7 - Resultados do estudo da magnitude do gradiente (sequência "Miss America")
- 4.8 - Resultados do estudo da magnitude do gradiente (sequência "INESC")
- 4.9 - Distribuição da magnitude do gradiente - imagens original e diferença (32 imagens da sequência "Miss America")
- 4.10 - Distribuição da magnitude do gradiente - imagens original e diferença (100 imagens da sequência "INESC")
- 4.11 - Distribuição da média para imagens originais
- 4.12 - Distribuição da média para imagens diferença
- 4.13 - Distribuição da magnitude do gradiente para vários tipos de segmentação
- 4.14 - $|\nabla I|_{\min}$ vs. número de blocos uniformes e de transição
- 4.15 - $|\nabla I|_{\min}$ vs. relação sinal-ruído (sequência "Miss America")
- 4.16 - Relação sinal-ruído para dois intervalos de quantificação
- 4.17 - Imagem original e imagem codificada com 6 bits para os blocos uniformes e 10 bits para os blocos de transição (8ª imagem da sequência "Miss America")
- 4.18 - Exemplos de blocos de erros não detectáveis
- 4.19 - Casos especiais contemplados
- 4.20 - Exemplo da resolução de outros blocos especiais, por meio da adição dos grupos escolhidos
- 4.21 - Sequência "INESC" com resolução de Casos Especiais (CESP) e Sem Casos Especiais (NCESP)
- 4.22 - Imagem codificada sem e com a resolução dos casos especiais (10ª imagem da sequência "Miss America")

- 4.23 - Compensação de movimento antecedendo a segmentação
- 4.24 - Segmentação e compensação de movimento realizadas conjuntamente
- 4.25 - Compensação de movimento precedendo o bloco de codificação
- 4.26 - Exemplo do algoritmo CSA para $W=8$
- 4.27 - Imagem compensada em movimento (8ª imagem da sequência “Miss America”)
- 4.28 - Imagem de fundo no codificador e decodificador
- 4.29 - Sistema de compressão
- 4.30 - Sistema de decodificação
- 4.31 - Imagem de fundo e contadores
- 4.32 - Actualização dos contadores e da imagem de fundo
- 4.33 - Imagem original, imagem de fundo e imagem codificada (22ª imagem da sequência “Miss America”)
- 4.34 - A luminância e as duas componentes da cromaticidade (formato QCIF)
- 4.35 - Sistema de compressão da cor

Índice das tabelas

Capítulo 3

3.1 - Tipo de blocos usados na segmentação

3.2 - Resultados da segmentação de 32 imagens da sequência “Miss America”

Capítulo 4

4.1 - Mapeamento do valor da tangente na orientação principal correspondente

4.2 - Análise de complexidade

4.3 - Número de bits de codificação do gradiente vs. SNR

4.4 - Codificação da média em dois intervalos distintos de quantificação

4.5 - Atribuição do número de bits a cada tipo de blocos

- 4.6 - Taxa de compressão obtida
- 4.7 - Casos especiais vs. número de blocos
- 4.8 - Análise de complexidade de diversos algoritmos de procura
- 4.9 - Resultados comparativos entre a utilização ou não utilização da compensação de movimento
- 4.10 - Número de blocos de cada dimensão obtidos com e sem compensação de movimento
- 4.11 - Resultados finais do módulo de compensação de movimento
- 4.12 - Número de blocos utilizados como fundo em função do limiar de semelhança
- 4.13 - Número de blocos utilizados e de blocos de actualização da imagem de fundo
- 4.14 - Limiar dos contadores vs. número de blocos de actualização e utilização
- 4.15 - Relação entre as dimensões dos blocos de actualização e o número de actualizações e utilizações, bem como a área construída da imagem de fundo
- 4.16 - Relação entre a dimensão dos blocos de utilização e o número de utilizações e actualizações
- 4.17 - Percentagem de cada tipo de bloco
- 4.18 - Resultados da relação sinal-ruído resultante do módulo de identificação de fundos

Capítulo 1

Introdução

A capacidade de permuta de informação é intrinsecamente humana. A impossibilidade de comunicar tem como consequência imediata o isolamento e levanta um obstáculo à evolução em todas as áreas do conhecimento. Mais importante, a sobrevivência do homem no mundo é conseguida através da recolha e análise de informação. Não é pois de estranhar que se considere a informação como um bem precioso a adquirir.

Se a informação e a sua comunicação sempre foram valiosas para o homem desde o princípio dos tempos, no mundo moderno a falta de informação é de tal forma evitada que o homem passa grande parte do seu dia a tentar actualizar o conhecimento do mundo que o rodeia.

A enorme quantidade de informação que o homem moderno tem que tratar no seu dia-a-dia, apresenta-lhe o problema de como a gerir eficientemente. Se a gestão da informação é importante, a comunicação da informação a outros indivíduos, é essencial.

Sem dúvida, a palavra escrita ou falada é um modo usual na troca de informação. No entanto, a imagem é muitas vezes um meio mais seguro e inequívoco de transmissão de informação.

A frase “uma imagem vale mil palavras” é, provavelmente, considerada por muitos como um axioma, particularmente por aqueles que necessitam de comunicar uma ideia ou ponto de vista. Por esta razão, a imagem é uma forma preferencial de veicular informação.

Mas, se a afirmação de que uma imagem vale mil palavras é literalmente verdadeira, a frase torna-se ainda mais exacta quando a mesma se aplica à área da informática. A elevada quantidade de unidades de informação que são necessárias para a representação de uma imagem digitalizada, obriga a um esforço no desenvolvimento de técnicas capazes de o reduzir.

De facto, o eficiente armazenamento em suporte magnético de imagens para posterior visualização tem sido objecto de intensa investigação. Devido à restrição imposta pela limitada capacidade da memória de um computador, bem como para uma mais simples gestão das imagens armazenadas, a compressão da imagem é um objectivo por todos pretendido.

Por outro lado, a transmissão de uma imagem digitalizada através de um canal de comunicação beneficiará igualmente com o desenvolvimento de técnicas de compressão de imagem. De facto, a limitada capacidade de um canal de comunicações, devido às restrições físicas inerentes, impõe um tempo excessivamente longo para a transmissão de imagens não codificadas.

Mais exigente ainda do que o envio de uma imagem, é a transmissão de sequências de imagens, como é o caso das aplicações de videotelefone ou de videoconferência. Nestas, a particularidade reside no facto de ser pretendida uma compressão de imagem, não só eficiente, mas rápida, de modo a ser possível executá-la em tempo real e assim manter um elevado número de imagens transmitidas por segundo.

O presente trabalho situa-se na área de compressão de sequências de imagens de baixa complexidade para muito baixas cadências. Entende-se por compressão de baixa complexidade, aquela que pode ser efectuada tendo por base um hardware pouco

poderoso. Deste modo, as técnicas descritas neste trabalho são passíveis de implementação num vulgar computador pessoal. O objectivo da compressão de vídeo para muito baixas cadências, consiste na transmissão das imagens codificadas através de canais cujo débito binário é igual ou inferior a 64 Kbits/s, canais estes oferecidos pela Rede Digital com Integração de Serviços; a videoconferência, o videotelefone e a videovigilância são possíveis aplicações das técnicas desenvolvidas nesta área da compressão de vídeo.

1.1. Motivação do autor

Um dos factores de motivação para a realização deste trabalho consistiu na participação do autor num projecto de videovigilância do INESC. Neste projecto pretendeu-se o desenvolvimento de técnicas eficazes de codificação, de modo a ser possível a transmissão de imagens vídeo, através da Rede Digital com Integração de Serviços, de uma estação remota para uma estação central de visualização.

As limitações da banda disponível para a transmissão e do hardware utilizado implicaram o estudo de algoritmos de baixa complexidade de implementação, que permitissem uma elevada taxa de compressão e uma qualidade das imagens codificadas aceitável para aplicações de videovigilância.

O presente trabalho vem no seguimento do esforço até agora desenvolvido, no sentido de obter sistemas eficientes de compressão de imagens vídeo em tempo real. Pretendeu-se o estudo de diversas técnicas de codificação, que diminuíssem a complexidade da implementação, permitissem o aumento da capacidade de compressão do sistema, bem como a melhoria da qualidade das imagens finais.

1.2. Estrutura da tese

Esta tese está organizada em cinco capítulos. Após a introdução ao tema da tese, seguem-se os capítulos destinados ao estudo da compressão de vídeo.

- ⇒ *Capítulo 2*: O Capítulo 2 [*Compressão de Imagem*] faz uma breve introdução à área da compressão de imagem, bem como à teoria da informação. Adicionalmente é feita a apresentação das técnicas mais utilizadas: a codificação por predição, a codificação por transformadas, a quantificação vectorial e a compensação de movimento.
- ⇒ *Capítulo 3*: Neste capítulo [*Compressão de Vídeo de Baixa Complexidade Para Baixas Cadências*] é descrito um sistema de compressão que foi o ponto de partida para o trabalho desta tese.
- ⇒ *Capítulo 4*: O estudo de alternativas para a codificação no sistema base, foi o objectivo do Capítulo 4 [*Técnicas de Codificação de Vídeo Para Baixas Cadências*]. A codificação de imagem por padrões visuais, a compensação de movimento, o algoritmo de identificação de fundos, bem como a codificação da cor, foram objecto de análise.
- ⇒ *Capítulo 5*: No Capítulo 5 [*Conclusões*] são apresentadas as conclusões mais relevantes da realização deste trabalho e é identificada uma área passível de ser explorada num eventual trabalho futuro.

Capítulo 2

Compressão de Imagem

2.1 Introdução

Na compressão de imagem pretende-se que esta seja representada com o menor número de unidades de informação possível [Jai81]. Este objectivo é motivado pelo número elevado de bits que a digitalização de uma imagem gera e que, para o seu armazenamento ou transmissão, não é de modo algum o mais conveniente. As técnicas de codificação de imagem intentam, explorando a estrutura geométrica e estatística das imagens, reduzir (comprimir) o número de bits necessário à representação e reprodução de um duplicado fiel da imagem original. As características do receptor, seja ele humano ou um sistema de processamento de imagem, deverão ser uma preocupação constante, pois será ele que irá julgar a qualidade da imagem reconstruída.

A digitalização de uma imagem consiste na amostragem espacial e na quantificação em números inteiros, da luminosidade obtida por um sistema de aquisição analógico.

É designada por forma canónica a matriz de N por M , de números inteiros, cada um codificado com B *bits* e representando um elemento da imagem ou *pixel*. Necessita-se, pois, de $N \times M \times B$ bits para descrever uma qualquer imagem que tenha sido adquirida. Provavelmente, esta será uma das piores representações possíveis do ponto de vista de economia de unidades de informação.

Num algoritmo de compressão é explorada a *redundância* de informação existente numa imagem, consistindo esta na informação que pode ser recuperada no decodificador ainda que não tivesse sido transmitida ou armazenada. Existe, ainda, um outro tipo de informação, denominada de *irrelevante*, que apesar da sua extracção conduzir a uma imagem reconstruída diferente da original, a distorção introduzida não é detectável pelo sistema visual humano. Pode ainda distinguir-se uma outra classe de informação, a *relevante*, sendo esta a informação que se for retida no codificador originará distorção visível pelo observador.

Pretende-se com as técnicas de compressão de imagem elaborar algoritmos capazes de compactar a informação importante, eliminando de alguma forma a redundância. Desta forma, no receptor poder-se-á, através da operação inversa, reconstruir a imagem original. Se, com o objectivo de aumentar a taxa de compressão, for consentido que alguma informação irrelevante ou pouco relevante seja eliminada, no receptor obter-se-á uma imagem reconstruída que será de algum modo diferente da original.

Pode, numa primeira abordagem, classificar-se os algoritmos de compressão em compressores sem perda e compressores com perda. Os primeiros permitem a reconstrução no decodificador de uma imagem fiel ao original. Os compressores com perda consentem a existência de distorção na imagem final. Para ser possível comparar diferentes algoritmos, bem como avaliar a distorção por si introduzida, existem diversas medidas de distorção, umas mais ligadas a conceitos matemáticos, outras mais perto de uma avaliação feita pelo sistema visual humano.

2.2 Teoria da informação

Sendo o propósito da codificação de imagem o uso de forma eficiente da informação, é de todo o interesse o estudo da informação contida nas imagens, de modo a ser possível a utilização das principais conclusões desenvolvidas na *Teoria da Informação*. A teoria da informação é uma teoria matemática, que se concentra em três conceitos básicos:

1. A *Medida da Informação*
2. A *Capacidade de um canal* de comunicações para transmitir informação
3. A *Codificação* como meio de utilizar os canais com toda a sua capacidade.

Esta teoria fornece pistas para melhorar a eficiência da comunicação da informação e estabelece limites para essa eficiência. Shannon foi quem, em 1948, publicou o artigo “*A mathematical theory of communication*” [Sha48], considerado como a base da moderna teoria da informação, onde procurou um método geral de medir a quantidade de informação e a capacidade de um canal. Shannon formulou um teorema que é considerado como o *Teorema Fundamental da Teoria da Informação* que faz uso dos conceitos acima enunciados. A conclusão mais importante deste teorema é a possibilidade da transmissão da informação sem erros, através de um canal com ruído, se a codificação for adequada de modo a adaptar a fonte e o canal para a máxima transferência fiável de informação. Como resultado do teorema fundamental, mesmo que o canal seja ruidoso, é sempre possível efectuar uma codificação de canal, de modo a que a informação transferida não seja corrompida. A separação de um sistema de codificação em duas partes principais, a fonte e o canal, permite a resolução de dois problemas distintos. Por um lado, o problema de erro na transmissão da informação devido às características do canal. Por outro lado, o problema da codificação da fonte de informação. É codificando a fonte que se obtém a compressão dos dados, aumentando, desse modo, a eficiência da comunicação.

Existe um ramo da teoria da informação que se dedica especialmente à compressão de dados: a “*rate-distortion theory*” [Ber71][Gra90]. Esta teoria afirma que é possível

determinar um limite mínimo para o débito de transferência de informação de um sinal (em bits por símbolo), dado um certo valor da distorção. Se a taxa de transmissão for inferior a esse limite, o valor da distorção será mais elevado. Acrescenta, ainda, que uma maior aproximação a esse limite é efectuada pela codificação de longas sequências de dados [VeB93].

Um resultado que é de interesse imediato, é o facto de ser possível, em teoria, calcular o número mínimo de bits por pixel necessário para que na recepção se obtenha uma reprodução sem erros da imagem original [Cla85]. Este valor constitui um bom guia de comportamento quando uma imagem é sujeita a uma operação de compressão com perda de informação. Procurando, desse modo, alcançar uma maior taxa de compressão, esses sistemas são usados abaixo do limiar de preservação de informação, introduzindo uma certa distorção. Na prática, é geralmente difícil calcular esse valor devido ao enorme número de relações entre a ocorrência dos diversos valores dos elementos da imagem, mas é possível calcular uma estimativa.

Tem-se então um sistema constituído por uma fonte que produz informação, a qual é enviada ao receptor através de um canal de comunicação. Essa informação pode ser quantificada pela medida de informação

$$I(x) = f(p(x)) = -\log_b p(x)$$

sendo I , a *auto-informação*, uma função da probabilidade dos símbolos produzidos pela fonte. A base do logaritmo pode ser a que se escolher como conveniente, sendo usual a utilização da base 2 e, neste caso, o valor da auto-informação é obtido em bits. Esta medida permite o conhecimento da informação transportada quando o símbolo de ordem x é transmitido, se a fonte for discreta e os símbolos forem estatisticamente independentes.

Tendo em atenção a fórmula da auto-informação é visível que, quanto maior for a probabilidade de ocorrência do evento $x=X$, sendo x um elemento arbitrário da imagem e X um valor particular da luminância, menor é a informação transportada por esse acontecimento. Por outro lado, quanto maior a improbabilidade de uma ocorrência, maior a informação que é comunicada.

A taxa de informação média é dada pela *entropia* (medida em bit/símbolo)

$$H(x) = - \sum_{x=1}^N p(x) \log_2 p(x)$$

onde $p(x)$ é a probabilidade da amostra x ter um valor X de um alfabeto de N símbolos. Esta medida de entropia não leva em consideração o facto de uma amostra ser estatisticamente dependente dos seus vizinhos; é apelidada de *entropia de ordem zero*.

No caso de uma imagem ser quantificada em $N=2^B$ símbolos, sendo B o número de bits de cada elemento da imagem, e sendo os símbolos equiprováveis, obtém-se

$$p(x=X) = 2^{-B}$$

e a auto-informação

$$I(x) = -\log_2(2^{-B}) = B \text{ bits}$$

Portanto, e atendendo à fórmula de auto-informação, uma palavra binária de B bits contém B bits de auto-informação.

Da mesma maneira, pode observar-se que o valor máximo de $H(x)$ ocorre no caso de equiprobabilidade dos valores de x , resultando

$$H(x) = - \sum_{x=1}^{2^B} 2^{-B} \cdot \log_2(2^{-B}) = B \text{ bits}$$

Sendo B o número médio de bits necessário para representar cada elemento da imagem, sem que haja perda de informação. As imagens de origem natural não têm esta distribuição de probabilidades, sendo portanto possível a sua representação por menos bits do que B . Adicionalmente, a interdependência entre as amostras da imagem é outra razão para um decréscimo daquele valor. De facto, se a *entropia conjunta* de pares de elementos,

$$H(x,y) = - \sum_{x,y=1}^{2^B} p(x,y) \cdot \log_2 p(x,y) \quad \text{bits}$$

for levada em conta, é esperado um valor médio menor do que o obtido anteriormente. Em média, o conhecimento do valor de um elemento indica algo acerca dos seus vizinhos. Os casos onde não existe grande dependência inter-elementos são zonas de brusca transição da luminância, locais em que a informação de um pixel não permite o correcto conhecimento dos seus vizinhos.

2.3 Técnicas de compressão de imagem

Como já foi referido, para se conseguir compactar a informação ter-se-á que, de algum modo, utilizar técnicas que separem a redundância e a informação irrelevante da relevante, decidindo posteriormente quanto desta se vai transmitir ou armazenar. Será feito a seguir o estudo das técnicas mais utilizadas na compressão de imagem.

2.3.1 Codificação por predição

Neste tipo de codificação é utilizado o facto do valor de pixels espacialmente próximos estarem bastante correlacionados entre si, sendo por isso possível que se faça uma previsão do valor de um dado pixel tendo em conta pixels na sua vizinhança. Outro aspecto a levar em conta é, no caso de compressão de sequências de imagens, a correlação que existe no plano temporal entre imagens consecutivas [Jai81].

Pode ver-se na figura 2.1 os blocos constituintes de um sistema de predição básico, onde se distingue o filtro de predição e o bloco de quantificação.

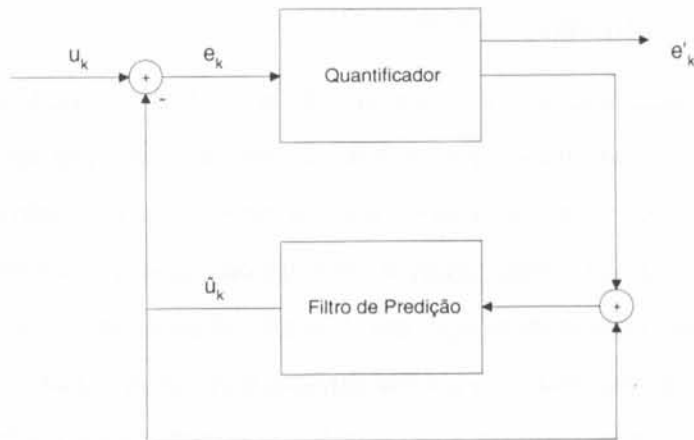
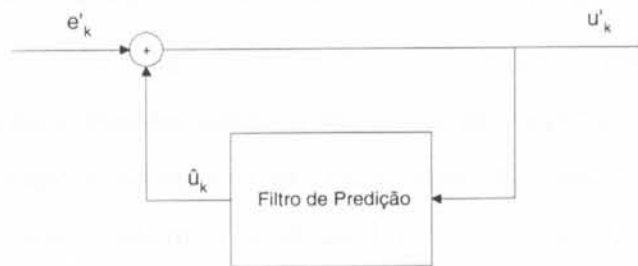
CodificadorDescodificador

fig. 2.1 - Sistema de previsão básico

O que se vai transmitir neste tipo de codificador é a quantificação do erro entre o sinal de entrada e a sua estimativa, sendo esta baseada nas amostras já transmitidas. Se o pixel a codificar for estatisticamente dependente dos anteriores, obter-se-á uma boa previsão e, consequentemente, um erro de previsão razoavelmente baixo. O bloco de quantificação traduz o erro de previsão numa palavra binária, de um alfabeto de N , que o represente da melhor forma. A escolha do método de previsão e do tipo de quantificação é de capital importância para o desempenho de um sistema deste tipo. Os filtros de previsão podem distinguir-se entre lineares ou não-lineares, de acordo com a função de previsão implementada. Outra característica importante é a dimensionalidade do filtro, sendo “unidimensional” no caso de se utilizarem amostras na mesma linha do pixel a estimar, “bidimensional” quando se utilizam também as linhas antecedentes, ou “tridimensional”

quando se usam os pixels de imagens anteriores [Net80]. Tenta-se, deste modo, explorar as possíveis interdependências dos pixels.

Considere-se uma sequência de variáveis aleatórias $\{u_n\}$ correspondentes ao valor das amostras dos pixels, e admita-se que tenham já sido transmitidas as primeiras $n=k-1$ amostras. De acordo com o que foi referido anteriormente, uma estimativa do valor de u_k é feita pelo filtro de predição, tendo em conta o valor das amostras anteriores. À saída do filtro tem-se então um valor estimado $\hat{u}_k$ que difere do original de $e_k = u_k - \hat{u}_k$. Este erro de predição é quantificado e transmitido em vez do sinal original u_k e, uma vez que o erro tem uma variância muito menor do que a da amostra, o quantificador poderá ter um menor número de níveis de quantificação, efectuando-se assim a compressão.

O erro é então quantificado num dos vários níveis do quantificador, sendo-lhe atribuído um valor e'_k que difere do erro calculado e_k de uma forma que depende das qualidades do bloco de quantificação.

Na recepção, é feita uma operação análoga de predição, operação esta que, pelo facto de se basear apenas em amostras já processadas, permite que o receptor esteja em perfeita sincronia com o emissor. Assim, é feita uma estimativa que, adicionada ao erro quantificado e transmitido, permite a reconstrução do sinal

$$u'_k = \hat{u}_k + e'_k$$

No final da reconstrução verificar-se-á um erro devido à quantificação que poderá ser calculado por

$$\delta u_k = u_k - u'_k = e_k - e'_k = q_k$$

ou seja, a diferença entre o sinal original e o sinal reconstruído no decodificador, é igual ao erro que se cometeu na substituição de e_k por e'_k [Jai81].

Dois tipos de codificadores podem ser imediatamente identificados dependendo do número de níveis de quantificação. Se só existirem 2 níveis, designa-se por *Modulação*

Delta (*Delta Modulation* ou *DM*). Se, por outro lado, o codificador tiver mais de 2 níveis, toma o nome de *Differential Pulse Code Modulation* (*DPCM*).

2.3.2 Codificação por transformada

Uma alternativa à codificação preditiva é a codificação por transformada. Neste tipo de codificação aplica-se uma transformação a um vector de dados da imagem, de modo a obter-se, no final, um outro vector cujos dados sejam menos correlacionados e concentrem num pequeno número de elementos a maior energia possível. Esta última condição permite tratar de forma diferenciada os diversos elementos, codificando os que têm associado um maior nível de energia com maior precisão e não dando tanta atenção, ou mesmo desprezando, os elementos de energia mais baixa.

O ideal neste tipo de codificação seria explorar a correlação existente entre todas as amostras da imagem, de modo a obter-se, no final, uma compactação da energia o mais eficiente possível. Esta metodologia é, no entanto, extremamente pesada do ponto de vista computacional, sendo normal e até natural a divisão da imagem em pequenos vectores sobre os quais a codificação é feita.

De modo semelhante ao que foi referido na secção da codificação preditiva, também a transformada pode ser “unidimensional”, “bidimensional” ou “tridimensional”. A aplicação de transformadas unidimensionais é efectuada sobre um vector de dados, com um certo número de elementos de uma linha ou coluna da imagem. Na maioria das aplicações utiliza-se uma transformada bidimensional; neste caso a transformada é aplicada a blocos de pixels da imagem. A transformada tridimensional, explora a relação entre imagens de uma sequência.

Um aspecto relevante das transformadas é o facto de todas as que se utilizam em compressão de imagem serem separáveis, isto é, a transformação a duas ou três dimensões pode ser obtida pela aplicação sucessiva da operação transformada unidimensional nas direcções apropriadas. Pode então tratar-se do caso unidimensional sem perda de generalidade para as outras transformadas.

O que se pretende é encontrar uma transformada que, aplicada a um vector de dados da imagem da forma

$$\mathbf{x} = [x_1, x_2, \dots, x_N]^T$$

em que os valores x_i correspondem a valores de pixels da imagem, se obtenha um outro vector $\mathbf{y}$ com as características de correlação e compactação de energia já enunciadas.

A operação é efectuada através de uma transformação linear do tipo

$$\mathbf{y} = \mathbf{A} \mathbf{x}$$

em que $\mathbf{A}$ é a matriz de transformação de $N \times N$ elementos.

Pode ver-se na figura 2.2 o esquema de blocos para uma codificação deste tipo, onde se destaca, na parte da codificação, o vector de entrada $\mathbf{x}$, a matriz da transformação $\mathbf{A}$, o vector resultante $\mathbf{y}$ e a sua quantificação; na recepção executa-se a operação inversa de quantificação obtendo o vector $\mathbf{y}'$, e a transformada inversa $\mathbf{B}$, tendo como resultado o vector final reconstruído $\mathbf{x}'$.

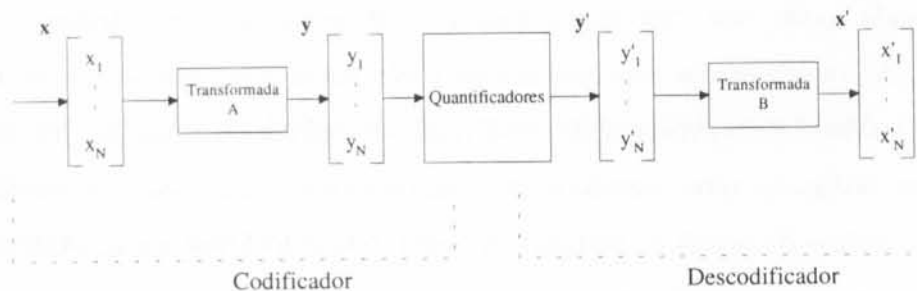


fig. 2.2 - Codificação por transformada unidimensional

Uma possível interpretação da operação de codificação por transformada é considerá-la equivalente à extracção de características de um vector. Os coeficientes da transformada, ou seja os elementos do vector $\mathbf{y}$ aos quais corresponde um valor elevado, são aqueles

cujos elementos do vector de dados e do vector base da transformada têm valor e sinal similar.

Outra análise, talvez mais explícita, que se pode efectuar é uma tradução geométrica da operação de transformação. Supondo que o vector de dados contém três elementos, sendo estes de alguma forma correlacionados, pode esperar-se que a maioria dos elementos se agrupem perto da linha $x = y = z$ (fig. 2.3).

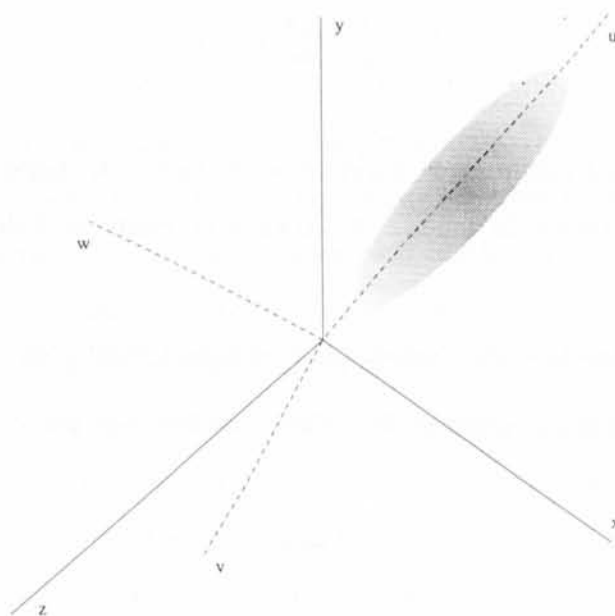


fig. 2.3 - Representação geométrica da codificação por transformada a 3 dimensões

Após a operação de transformação, obtém-se um novo sistema de coordenadas u , v e w , no qual os elementos da imagem têm o valor da componente u muito maior do que os das componentes v e w . O valor da coordenada u é agora o principal ponto de distinção entre os elementos. Devido à importância dos valores de cada coordenada, pode, então, transmitir ou armazenar-se com elevada precisão de codificação os elementos associados com a componente u , confinando os restantes a um tipo de codificação de inferior precisão ou mesmo não os processando. Da selecção dos elementos a transmitir ou armazenar, e da precisão da sua codificação, resulta a compressão dos dados. Um ponto importante a garantir é a invariância do produto interno escalar, o que implica uma conservação da

energia entre os dados e os coeficientes obtidos na transformação. A energia do vector de dados é compactada num certo número de coeficientes do vector obtido como resultado da transformada.

Pretende então encontrar-se uma matriz A que optimize a decorrelação dos dados, uma matriz B de reconstrução e os quantificadores mais apropriados, de modo a minimizar a distorção. Esta medida de distorção é normalmente representada pela média quadrática

$$D = \frac{1}{N} E \sum_{k=1}^N (x_k - x'_k)^2$$

em que D representa a distorção, N a dimensão do vector de dados, E é a esperança, x_k o valor do elemento k vector de dados e x'_k o valor do elemento k do vector obtido à saída do decodificador.

A solução do problema fica determinada pelas seguintes condições

1. A escolha óptima para a matriz de decodificação B é dada por

$$B = A^{-1} \Gamma$$

onde Γ é uma matriz diagonal cujos elementos são

$$\gamma_k = E[y_k y_k^*] / E[y_k^* y_k^*]$$

onde o símbolo * indica o conjugado, sendo y_k o valor do elemento k do vector resultante da transformação e y'_k o valor do elemento correspondente ao vector após a quantificação.

2. A matriz da transformada A óptima é a matriz de Karhunen-Loeve, cujas linhas são compostas pelos valores próprios ortogonais e normalizados da matriz de covariância do vector de dados.

3. A média do erro quadrático entre cada y_k e y'_k é minimizada pelo quantificador de Lloyd-Max; obtém-se, neste caso, a seguinte igualdade entre as matrizes

$$B = A^{*t}$$

ou seja, a matriz de reconstrução é igual à transposta do conjugado da matriz da transformação.

A transformada de Karhunen-Loeve maximiza a compressão da média da energia em quaisquer $M \leq N$ amostras de y . Assim, poder-se-á transmitir apenas os M elementos de maior amplitude, de modo a atingir a relação entre a compressão e a qualidade desejada [Jai81].

Apesar da transformada de Karhunen-Loeve ser a melhor em termos da minimização da média do erro quadrático, a complexidade da implementação limita o seu interesse prático. Em primeiro lugar, a função de covariância de uma imagem não é estacionária, implicando o uso de várias matrizes de covariância para cada região da imagem, ou usando uma matriz média para toda a imagem, com a consequente perda de desempenho. Um outro problema deve-se ao facto de os valores próprios da matriz por vezes serem singulares, não podendo, portanto, ser univocamente definidos [Net80]. Por fim, a necessidade de N^2 multiplicações por constantes, pode constituir um problema suplementar na implementação do algoritmo. Devido a estes problemas, foram estudadas técnicas de codificação por transformadas que, mantendo a capacidade de descorrelação dos elementos do vector de entrada, fossem de mais fácil aplicação em programas de compressão; são as chamadas *transformadas sub-óptimas*.

Muitas são as transformadas utilizadas, todas elas de maior simplicidade de implementação do que a oferecida pela transformada de Karhunen-Loeve. As transformadas de Fourier, Slant, Hadamard, Haar ou a de coseno são apenas vários exemplos de transformadas aplicadas à codificação de imagem. Destas, a transformada discreta de coseno (DCT), pertencente a uma família de transformadas sinusoidais, tem sido objecto de grande interesse no campo do processamento de imagem. O facto de se tratar de uma transformada de coeficientes reais e do seu desempenho aproximar-se do atingido pela transformada de Karhunen-Loeve, bem como a existência de algoritmos rápidos de processamento, levam a que esta transformada seja de utilização preferencial em processamento de imagem.

2.3.3 Quantificação vectorial

As técnicas anteriormente descritas têm em comum o facto de incluírem, a um determinado momento, um bloco de quantificação de escalares para posterior transmissão ou armazenamento da informação. Na codificação por predição, essa quantificação é feita após o cálculo da diferença entre o valor original e a sua estimativa. A codificação por transformada inclui esse bloco para quantificar os coeficientes obtidos e, embora a operação inicial seja efectuada sobre um vector de dados, a quantificação é sobre valores reais.

De acordo com um resultado fundamental da teoria da informação, pode sempre obter-se um melhor resultado se forem codificados vectores de dados em vez de escalares, mesmo que a fonte de dados não possua memória, ou seja, que os valores consistam numa sequência de variáveis aleatórias independentes [Gra84]. Deste ponto de vista, os sistemas que até agora foram estudados são inerentemente sub-óptimos. É portanto possível a realização de algoritmos de compressão de imagem que utilizem vectores, obtendo, pelo menos em teoria, melhores resultados do que os que codificam escalares. Nos últimos anos da década de 70 surgiram os primeiros trabalhos de quantificação vectorial aplicados à compressão de voz. A utilização deste tipo de quantificação na codificação de imagem revelou-se de grande interesse, visto o reduzido nível de complexidade de implementação e os elevados resultados das taxas de compressão.

O problema da quantificação vectorial pode ser visto como uma generalização do caso escalar para a quantificação de um vector. Partindo de um vector de dados, o quantificador vectorial procura numa tabela interna de códigos um outro vector de substituição que considera a sua melhor descrição. Após esse processo de busca, o vector codificado é representado por um índice de entrada da tabela de códigos mencionada (ver fig. 2.4).

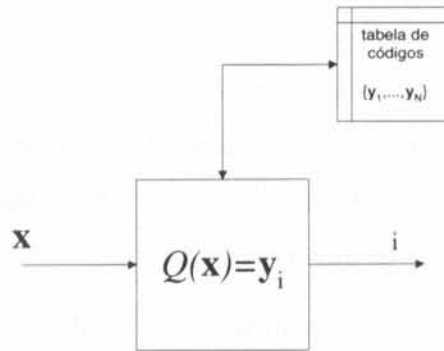


fig. 2.4 - Sistema de quantificação vectorial

De acordo com o referido anteriormente, pode encarar-se a quantificação vectorial como uma forma de reconhecimento de padrões, em que se procura relacionar o “padrão” de entrada, sendo este representado pelo vector de dados, por um de vários “padrões” pré-determinados.

Matematicamente, um quantificador vectorial Q de dimensão k e tamanho N realiza um mapeamento de um vector de entrada do espaço Euclidiano de dimensão k , $\mathcal{R}^k$, num conjunto finito C contendo N possíveis reproduções distintas de saída, *vectores de código* ou *palavras de código*.

Estas palavras de código são vectores do mesmo espaço $\mathcal{R}^k$, constituindo o conjunto C a que se dá o nome de tabela de códigos, ou seja, um dicionário de palavras de código.

Formalizando o que se afirmou, tem-se

$$Q: \mathcal{R}^k \rightarrow C$$

em que

$$C = (y_1, y_2, \dots, y_N) \quad , y_i \in \mathcal{R}^k$$

onde y_i são os vectores de código.

Podem considerar-se dois casos extremos na construção da tabela de códigos. Um primeiro caso é a constituição do dicionário dos vectores de código por um só vector.

Nessa condição, qualquer tentativa de codificação da entrada geraria uma única e constante saída. O segundo caso diz respeito a uma tabela de códigos com um número infinito de vectores, representando todo o espaço. Deste modo, a codificação conduziria a um vector de saída igual ao vector original; ou seja, uma reprodução exacta mas com a desvantagem de não se efectuar qualquer compressão.

Uma primeira medida que permitirá qualificar um sistema de quantificação vectorial, é a *precisão* ou *taxa de codificação*. Esta medida é formulada por

$$r = (\log_2 N) / k$$

cujos resultados, expressos em bit por componente do vector, indicam a resolução que se pode esperar com um conjunto de vectores de representação de dimensão k e tamanho N . Note-se que quanto maior for a tabela de códigos, para uma dimensão k fixa do espaço, maior será a precisão obtida. Sabendo-se que um vector é construído por um conjunto ordenado de pixels, pode, portanto, referir-se à precisão como bit por pixel e, assim, quanto maior for a dimensão do vector, maior é a compactação possível.

A operação de quantificação vectorial consiste, como já foi referido, na substituição de um vector de entrada por um outro e, visto que o conjunto de possíveis vectores de representação é um subconjunto do espaço Euclidiano original, existem vários vectores de entrada que originam uma mesma resposta. Pode portanto dividir-se o espaço $\mathcal{R}^k$ em N *partições* ou *células*. Definindo o conceito *célula* como

$$R_i = \{ \mathbf{x} \in \mathcal{R}^k : Q(\mathbf{x}) = y_i \}$$

onde a i -ésima célula é composta pelos vectores $\mathbf{x}$ pertencentes a $\mathcal{R}^k$ cujo vector da tabela de códigos correspondente é o mesmo y_i .

Evidentemente, para se conseguir uma relação unívoca entre $\mathcal{R}^k$ e C , é necessário que qualquer vector do primeiro espaço tenha um correspondente no segundo, ou seja, que a reunião de todas as células perfaçam na totalidade o espaço $\mathcal{R}^k$.

$$\bigcup_{i=1}^N R_i = \Re^k$$

e ainda que um vector de entrada esteja contido numa e numa só célula

$$R_i \cap R_j = \emptyset, \quad \forall_{i \neq j}$$

Pode observar-se na figura 2.5 um exemplo de uma partição de $\Re^2$.

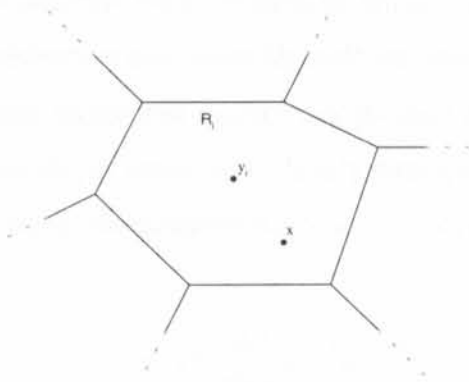


fig. 2.5 - Exemplo de uma partição de $\Re^2$

O vector a codificar x é atribuído a uma célula R_i , cujo vector de representação tem a nomenclatura y_i . Todos os vectores de entrada que estejam contidos naquela célula produzem uma mesma resposta do codificador.

Como foi já referido anteriormente, a operação de compressão é efectuada transmitindo o codificador apenas o índice de entrada na tabela de vectores de códigos. A operação pode ser definida da seguinte forma

$$\xi : \Re^k \rightarrow I$$

sendo $I = \{ 1, 2, \dots, N \}$ o conjunto dos índices dos elementos de C , ou seja, o conjunto dos índices dos vectores de código. Por outro lado, a operação de descodificação tem apenas

de efectuar a atribuição do vector correspondente ao índice recebido, sendo definida da seguinte forma

$$D : I \rightarrow \mathbb{R}^k$$

É evidente que a operação de codificação é a que necessita de maior tempo de processamento, visto ter que procurar um vector representativo da entrada na tabela de códigos, enquanto que a descodificação consiste na simples atribuição de um vector, dado um índice específico.

O objectivo primeiro de um sistema de quantificação vectorial é a substituição do vector original por uma sua representação. Essa permuta implica necessariamente uma *distorção* que pode ser associada a um custo $d(\mathbf{x}, \mathbf{x}')$, que se tenta de alguma forma minimizar. Na prática, a medida do desempenho global do sistema é calculada pela média do custo associado, durante um intervalo de tempo suficientemente longo

$$\bar{d} = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} d(\mathbf{x}_i, \mathbf{x}'_i)$$

onde $\{\mathbf{x}_i\}$ representa a sequência de vectores a serem codificados. A função de custo deve ser tratável de modo a ser possível a sua análise, facilmente computável, para permitir a sua implementação em tempo real e subjectivamente significativa para se poder relacionar os seus valores, elevados ou baixos, com uma pior ou melhor qualidade subjectiva [Gra84].

Das várias medidas de *distorção* utilizadas, algumas merecem algum relevo devido à conveniência e observância do que se acabou de enunciar. Uma das medidas mais usadas é o *erro quadrático* ou, mais precisamente, o *quadrado da distância Euclidiana* entre dois vectores $\mathbf{x}$ e $\mathbf{x}'$, definida como

$$d(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|^2$$

A *média do erro quadrático* está mais associada a uma medida de energia do erro e é definida por

$$D = E d(\mathbf{x}, \mathbf{x}') = E(\| \mathbf{x} - \mathbf{x}' \|^2)$$

Estas medidas estabelecem um critério para a escolha do vector de código correspondente a uma dada entrada. No entanto, qualquer que seja a medida de distorção utilizada, existem duas propriedades às quais um quantificador vectorial óptimo obedece.

Propriedade 1: Sendo o objectivo o da minimização da média da distorção e dado um descodificador específico D , nenhum codificador ξ de quantificação vectorial pode fazer melhor do que seleccionar um vector $\mathbf{y}$ de um conjunto C que minimize a distorção à saída, ou seja, seleccionar um símbolo $\mathbf{y}$ que obedeça a

$$d(\mathbf{x}, D(\xi(\mathbf{x}))) = \min_{\mathbf{y} \in C} d(\mathbf{x}, \mathbf{y})$$

Portanto, sendo dado um descodificador específico, o melhor codificador é aquele que faz um mapeamento do vector de entrada no *vizinho mais próximo*. A propriedade acima referida permite a concepção de um codificador óptimo dado um determinado descodificador. Por outro lado, a implementação de um descodificador sendo dado um codificador, é referida na propriedade seguinte.

Propriedade 2: Dado um codificador ξ , nenhum descodificador D pode fazer melhor do que atribuir a cada símbolo $\mathbf{y}$ o *centróide generalizado* (ou *centro de gravidade*) de todos os vectores codificados como $\mathbf{y}$, ou seja,

$$D(\mathbf{y}) = \text{cent}(\mathbf{y})$$

Os quantificadores vectoriais que utilizam a regra do vizinho mais próximo são também chamados de *quantificadores de Voronoi* e têm como propriedade particular o facto de, o conhecimento da tabela dos vectores de código e da medida de distorção usada, definir completamente a partição do espaço inicial.

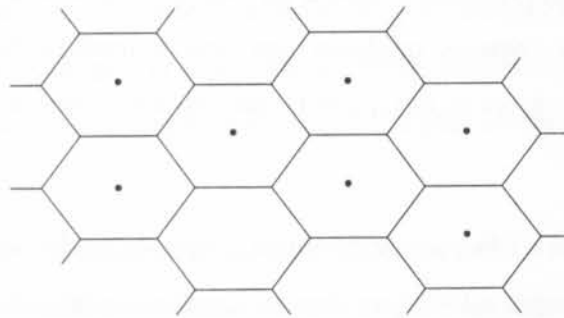


fig. 2.6 - Exemplo de regiões de Voronoi em $\mathbb{R}^2$

As propriedades acima enunciadas correspondem a uma generalização para a quantificação vectorial das condições definidas por Lloyd [Llo57] para um quantificador escalar óptimo. O critério do vizinho mais próximo obriga a que as regiões de quantificação sejam de Voronoi, enquanto que a segunda condição impõe o centróide como vector de representação de cada uma dessas regiões. Uma última condição de optimização, no caso de fontes com alfabeto discreto, como é o caso das imagens digitalizadas, é a impossibilidade da ocorrência de um vector a codificar sobre a fronteira de duas regiões, evitando assim a indecisão quanto ao vector de código.

A construção de tabelas de vectores de código é um problema central da quantificação vectorial, já que determina em grande parte o desempenho de todo o sistema. Uma tabela com um grande número de vectores, mas que não sejam representativos dos símbolos da fonte, determina um fraco desempenho do quantificador sendo, por isso, de grande importância o método utilizado na sua construção.

Um dos métodos mais utilizados é o *algoritmo de Linde-Buzo-Gray* [LBG80], por vezes referido como *algoritmo de Lloyd generalizado*. Trata-se de um algoritmo iterativo, tendo como base um conjunto de vectores de treino, uma tabela de vectores de código inicial,

uma medida de distorção e um critério de paragem. Não se consegue, no entanto, fugir ao problema de uma tabela inicial de vectores de código. Esta inicialização pode ser feita de várias formas, nomeadamente utilizando vectores igualmente distanciados no espaço, utilizando a técnica de *splitting* [Buz80][LBG80] ou usando uma escolha aleatória sobre os vectores de uma sequência de treino. A escolha de uma tabela inicial cujos vectores estejam numa vizinhança próxima da óptima é importante, uma vez que o algoritmo só garante encontrar mínimos locais, podendo divergir para uma solução de mínimo local distante do mínimo global.

Outra grande dificuldade de um sistema deste tipo, particularmente se é pretendida a sua implementação em tempo real, é a procura na tabela do vector de código que melhor representa o vector a codificar. É usual a classificação das técnicas de busca em técnicas óptimas e técnicas sub-óptimas. As primeiras procuram minimizar uma dada medida de distorção, escolhendo o vector que satisfaz essa condição. Das técnicas óptimas pode referir-se a *busca exhaustiva*, que consiste na obtenção da medida de distorção para todos os elementos da tabela, $d(\mathbf{x}, \mathbf{y}_i)$. Embora seja garantida a escolha do melhor vector para a minimização da distorção, o enorme peso computacional levou a que fossem procuradas outras técnicas, como a exploração da estrutura da própria tabela, que diminuíssem de alguma maneira o tempo necessário à busca do vector de código. As técnicas sub-óptimas produzem uma resposta que, podendo não ser o vector que minimiza a medida de distorção, reduz consideravelmente o tempo de processamento. Foram desenvolvidos diversos algoritmos de procura na tabela de códigos, baseados em técnicas sub-óptimas, sendo alguns descritos em [CGR84][ChG85][SoM89][TuH94].

2.3.4 Compensação de movimento

Na secção 2.3.1 [*Codificação por predição*] foi referido um tipo de codificação que utiliza a predição como método de reduzir a redundância entre elementos da imagem, quer no plano espacial a uma ou duas dimensões, quer no plano temporal.

A codificação por compensação de movimento baseia-se no princípio de que a imagem pode ser vista como um conjunto de objectos sobre um plano de fundo, objectos esses que

possuem três graus de liberdade podendo, portanto, efectuar movimentos de translação ou rotação.

O objectivo é determinar ou estimar um vector de deslocamento ou parâmetros que, aplicado a um objecto específico, permita caracterizar o tipo de movimento a que ele foi sujeito. Deste modo, o conhecimento desse vector permitirá, no decodificador, situar o objecto em causa na posição correcta.

A aplicação deste tipo de técnicas a sequências de imagens tem sido largamente usada, conseguindo-se elevados ganhos de desempenho na relação sinal-ruído, na taxa de transmissão, assim como na qualidade subjectiva resultante.

Entre os diversos métodos mais utilizados na compensação de movimento, uma característica em comum reside na tentativa de determinar um mesmo objecto, em duas imagens de uma sequência. Pretende conhecer-se a evolução, no espaço, de um objecto ou zona da imagem. Deste modo, é possível a *correspondência* entre duas zonas, de duas imagens de uma sequência. Esclarecendo-se esta correspondência, o cálculo do vector de deslocamento é de fácil execução, sendo igual à diferença das coordenadas das duas zonas de imagem. De um modo mais formal, pode considerar-se a correspondência como o processo pelo qual se identificam elementos em diferentes pontos de vista, representativos de um mesmo objecto em instantes de tempo diferentes, mantendo a identidade perceptual dos objectos em movimento ou em mudança [Veg89].

Um método que se utiliza para fazer corresponder um objecto entre duas imagens é o da *correlação cruzada*. Este método procura um valor máximo ou uma mudança de fase de uma função de correlação, sendo desta forma obtido um vector de deslocamento relativo ao objecto em causa. Em [ALR78] este método é aplicado ao cálculo do movimento de nuvens, através de imagens adquiridas por satélite.

Outro processo possível é o de atribuir uma determinada *função*, cujas propriedades são conhecidas, aos objectos a estudar. Assim, qualquer modificação nos parâmetros da função implica uma alteração do valor final, permitindo obter informações acerca das eventuais alterações de forma do objecto. Haass e Brubaker em [HaB80] fizeram corresponder a

luminância de uma nuvem a uma função tal, que lhes permite detectar movimentos de rotação.

Algumas *características* próprias de cada objecto também permitem a observação de movimento. O acompanhamento dessas características em sequências de imagens permite a determinação do vector do deslocamento ocorrido. Endlich e Wolf apresentam um método que, baseado em reconhecimento de padrões, possibilita o conhecimento da deslocação de nuvens [WHE77][EnW81].

Métodos mais complexos são implementados no caso de se estar na presença de objectos *não-rígidos*. Neste caso, é possível a alteração de forma e características, dificultando, deste modo, a correspondência de um mesmo objecto em duas imagens distintas. Combinações de técnicas aplicadas a objectos rígidos, como a correlação, complementada com uma descrição de possíveis modificações de forma, são usadas neste tipo de situações. Chen e Penna em [ChP86] introduzem o conceito de movimento generalizado, em que o movimento de um corpo rígido é uma particularização de um sistema mais complexo.

De todos os métodos baseados no conceito de correspondência entre dois objectos, um dos mais utilizados é o processo de *emparelhamento de blocos*. Este método parte do princípio que pixels espacialmente próximos sofrem de um movimento de translação semelhante. A imagem é dividida em pequenos blocos e procura fazer-se corresponder cada um deles a um outro numa imagem anterior, quantificando em seguida o vector de deslocamento de translação. Uma possível alternativa a este método consiste nos algoritmos *recursivos*. Neste caso, é calculada uma estimativa para cada *pixel* da imagem, com base na imagem actual e nas imagens já codificadas.

Devido às exigências que um sistema em tempo real impõe, algumas restrições à liberdade de movimento dos objectos tiveram de ser levadas em conta, tais como considerar somente movimentos de translação e admitir que os objectos são rígidos. Os requisitos em tempo de processamento apenas permitem que alguns algoritmos de compensação de movimento sejam passíveis de implementação em tempo real. Das técnicas de correspondência acima referidas, as que mais se adequam a uma baixa complexidade de implementação são as de

emparelhamento de blocos. Seguidamente, irão ser apresentadas com mais detalhe as técnicas referidas.

2.3.4.1 Técnica de emparelhamento de blocos

Neste tipo de processamento, a imagem a codificar é dividida em pequenos blocos, cada um sujeito a um *emparelhamento*, ou seja, a uma correspondência com uma dada posição numa imagem antecedente.

Originalmente, Jain e Jain propôs em [JaJ81] a divisão da imagem em pequenos blocos de dimensão fixa. Nesse algoritmo é efectuada uma busca numa área restrita, chamada *janela*, da imagem anterior, tentando-se localizar um bloco cuja semelhança com o bloco a codificar seja a maior possível. O vector de deslocamento é então obtido pela diferença das coordenadas entre o bloco da imagem actual e o bloco da imagem precedente.

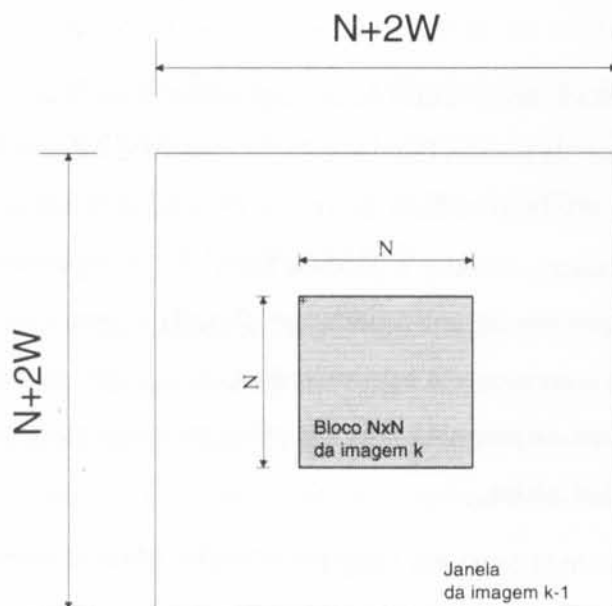


fig. 2.7 - Janela de procura

No decodificador, a recepção do vector de deslocamento permite fazer corresponder o bloco considerado da imagem actual, com um bloco da imagem anterior. Logo, a

descodificação é efectuada considerando o bloco codificado como uma reposição de um bloco da imagem antecedente.

As medidas de semelhança mais usadas, que determinam qual o bloco na imagem anterior correspondente ao bloco a codificar, são a *média do erro quadrático*

$$mse(i,j) = \frac{1}{N^2} \sum_{m=1}^N \sum_{n=1}^N [f(m,n) - f(m+i,n+j)]^2$$

$-W \leq i,j \leq W$

e a *média do erro absoluto*

$$mae(i,j) = \frac{1}{N^2} \sum_{m=1}^N \sum_{n=1}^N |f(m,n) - f(m+i,n+j)|$$

$-W \leq i,j \leq W$

onde $f(m,n)$ representa o bloco corrente bloco de N^2 pixels nas coordenadas (m,n) e $f(m+i,n+j)$ corresponde ao bloco na imagem anterior nas novas coordenadas $(m+i,n+j)$. O valor de W determina a dimensão da janela de busca dentro da qual o bloco é pesquisado, procurando minimizar a função de semelhança.

Adoptando uma das medidas, é necessário um método para localizar o bloco na janela de dimensão W considerada que satisfaça a condição de minimização. A *busca exaustiva* é um método seguro mas computacionalmente intenso, necessitando de calcular a função $(2W+1)^2$ vezes. São desejáveis soluções que diminuam a necessidade de cálculo da função de semelhança e o tempo de execução, como o *método de busca logarítmica bidimensional* utilizado por Jain e Jain em [JaJ81], onde são apenas necessários $(2+7\log_2 W)$ pontos de busca. Outro algoritmo utilizado por Koga *et al.* em [Kog81] é o *método de busca da direcção do vector em três passos*, exigindo este método $(1+8\log_2 W)$ buscas. Kappagantula e Rao [KaR83] introduziram o *algoritmo de estimação de movimento modificado*, com $(1+6\log_2 W)$ pontos de procura, e têm sido apresentados outros algoritmos como, por exemplo, a *procura de direcções conjugadas* de Srinivasan e Rao [SrR84], ou o *algoritmo de procura ortogonal* de Puri *et al.* [PHS87]. A diminuição

dos pontos em que se faz o cálculo da função de semelhança tem de ser criteriosa para que a solução obtida seja, pelo menos, próxima da óptima.

A dimensão da janela determina o número dos pontos de busca e cálculo da função de semelhança. Por outro lado, a dimensão do bloco determina o número de parcelas de cálculo da função. Deste modo, considerando um bloco quadrado de dimensão 4 por 4, a função de semelhança deverá efectuar 15 operações de subtração; um bloco maior de dimensão 8 por 8 necessitará de 63 subtrações. Por esta razão, a escolha da dimensão do bloco a compensar deverá ser criteriosa, pois um bloco demasiado grande implicará um aumento geométrico do tempo de processamento.

Como foi já referido, o facto de se estar a tratar de blocos, impõe um movimento de translação igual para todos os elementos contidos nesse bloco. Tem-se portanto de adaptar a dimensão do bloco, de forma a que não seja grande demais para que não ocorram falsos movimentos, nem pequeno demais, pois isso elevaria o número de vectores a transmitir, diminuindo o ganho na taxa de compressão.

Para conciliar as vantagens em ganhos de compressão de blocos de grandes dimensões com as vantagens em termos de fidelidade e qualidade subjectiva dos blocos de dimensões mais pequenas, um esquema de divisão da imagem em blocos de proporções variáveis é muitas vezes desejável.

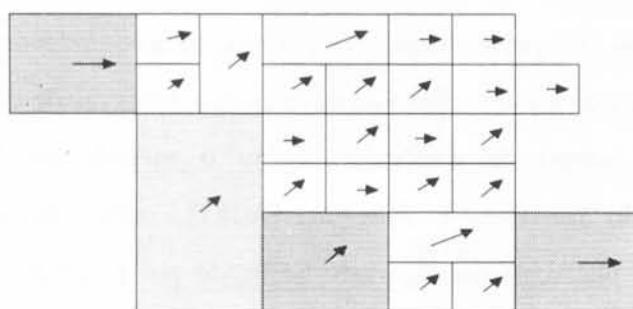


fig. 2.8 - Compensação de movimento em blocos de dimensão variável

Nesse caso, os blocos são classificados segundo parâmetros de movimento, sendo atribuídos blocos de dimensões mais pequenas a zonas de maior actividade e blocos

maiores a locais de pouco movimento. Deste modo consegue-se uma melhor adaptação às propriedades da imagem

2.3.4.2 Algoritmos recursivos

Quanto menor for a dimensão do bloco, melhor é o resultado final da compensação de movimento; no caso extremo, a cada *pixel* é atribuído um vector de deslocamento. Embora esta solução seja a melhor no que respeita à qualidade subjectiva, a transmissão de um tão grande número de vectores daria origem a um aumento de unidades de informação relativamente à imagem original, o que é incompatível como o objectivo da compressão de imagem.

Os algoritmos recursivos superam essa dificuldade, já que não é necessária a transmissão do vector de deslocamento. Assumindo uma estimativa inicial do deslocamento do *pixel*, novas estimativas são calculadas iterativamente de acordo com um termo de actualização

$$d_{i+1} = d_i + a_{i+1}$$

onde d_{i+1} é a mais recente estimativa e a_{i+1} é a parcela de actualização da iteração $i+1$. A estimativa inicial d_0 pode ser o vector de deslocamento de um pixel vizinho.

Netravali e Robbins [NeR79] propuseram o uso do gradiente no cálculo do termo de actualização “ a ”

$$d_{i+1} = d_i - \frac{1}{2} \epsilon \nabla d_i [DFD(x, y, d_i)]^2$$

sendo $DFD()$ uma função da diferença entre imagens, ϵ uma constante de convergência e ∇d_i o operador gradiente a respeito de d_i .

O facto de não existirem algoritmos rápidos de processamento para a compensação de movimento utilizando este tipo de algoritmos, tem como desvantagem o fraco desempenho em termos de tempo de execução, levando a que a sua aplicação em tempo real não seja aconselhável.

Capítulo 3

Compressão de Vídeo de Baixa Complexidade Para Baixas Cadências

3.1 Introdução

A implementação de um método de compressão de vídeo está condicionada por restrições e tem objectivos relacionados com:

- ⇒ a qualidade da imagem reconstruída
- ⇒ a taxa de compressão
- ⇒ a complexidade de implementação

O ideal seria um sistema que conjugasse uma elevada qualidade da imagem reconstruída no receptor, com uma elevada taxa de compressão e baixo tempo de processamento. Embora seja desejável, a dificuldade de atingir esse conjunto de condições impõe

cadências num ou em dois pontos, para se obter um terceiro de acordo com as expectativas. Uma elevada taxa de compressão não é, na maioria dos sistemas, compatível com uma elevada qualidade ou com um baixo nível de complexidade. É, portanto, com os três critérios acima expostos que se vai julgar um possível sistema de compressão de imagem vídeo.

No capítulo precedente foram apresentados os conceitos teóricos de várias técnicas de compressão de imagem. As técnicas descritas exploram as relações existentes entre os pixels das imagens. Recentemente foi proposta uma outra classe de técnicas de compressão - a compressão baseada em objectos [AHS89][FoK89][WSW90].

As técnicas de compressão baseadas em objectos, contrastando com os métodos convencionais, tiram partido do conhecimento da geometria a três dimensões dos objectos representados nas imagens. O codificador tem um modelo genérico do mundo, ao qual tenta fazer corresponder os objectos da imagem. Deste modo, o codificador adapta os modelos genéricos dos objectos, aos objectos que são detectados na imagem. A evolução e mudança dos modelos, ocorridas durante a sequência de imagens, são transmitidas ao decodificador. Essa informação consiste apenas na mudança dos parâmetros do modelo, sendo desta forma obtida a compressão.

Embora promissora em termos de taxa de compressão obtida, a elevada complexidade de um sistema de compressão de imagem baseada em objectos, não permite a sua implementação em hardware pouco poderoso. O elevado tempo de processamento necessário é, de igual forma, uma desvantagem neste tipo de sistema de compressão. Não foi considerada a utilização de tais técnicas de codificação neste trabalho, uma vez que se deseja um baixo nível de complexidade para o sistema de compressão.

Pretendeu-se desenvolver um sistema de compressão que, aliando uma elevada taxa de compressão e qualidade da imagem reconstruída, ainda assim seja de baixa complexidade. Como foi já referido, a codificação de vectores de dados é, segundo a teoria da informação, preferível à codificação de escalares. Deste modo, um método exequível é o que se apresenta na figura 3.1.

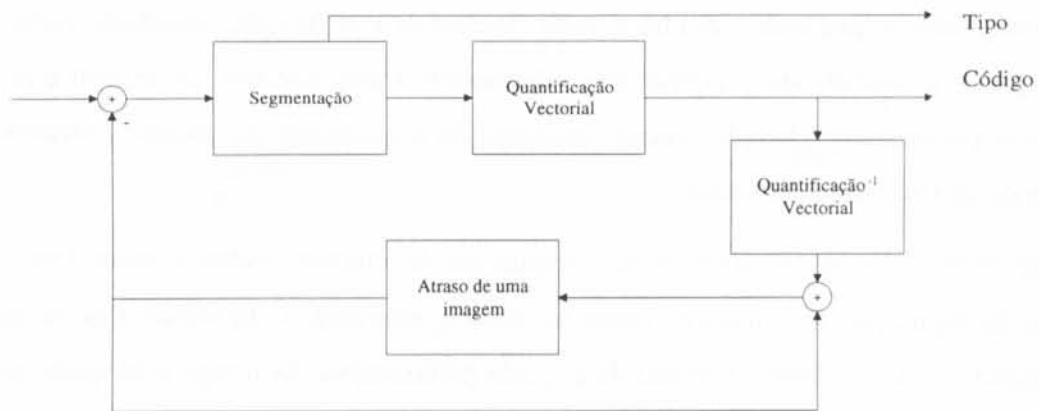


fig. 3.1 - Sistema de codificação básico

Este sistema é constituído por um módulo de segmentação que divide a imagem em blocos de tamanho variável consoante a informação de actividade local, um módulo de quantificação vectorial que é aplicado a cada bloco proveniente da segmentação e uma malha de realimentação constituída pela operação inversa da quantificação vectorial e por um bloco de atraso [Mar92].

Este sistema de compressão foi inicialmente proposto para codificação de imagens vídeo aplicado a sistemas de videovigilância (ver fig. 3.2). A imagem adquirida e codificada numa estação remota, é transmitida através de um canal de 64 Kbit/s da Rede Digital com Integração de Serviços (RDIS), para uma estação central, onde é feita a decodificação e visualização.

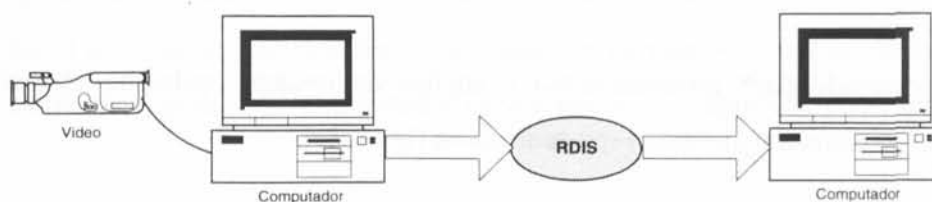


fig. 3.2 - Sistema de videovigilância

O presente trabalho pretende, com base neste sistema de codificação, introduzir variações e alterações, no sentido de conceber uma alternativa a este sistema, de acordo com os objectivos pretendidos: elevada taxa de compressão e qualidade da imagem, através de algoritmos de baixa complexidade.

O primeiro módulo de um sistema de compressão de imagens vídeo é naturalmente o sistema de aquisição de imagem; neste módulo é realizada a aquisição das imagens provenientes de uma câmara, que são de seguida processadas. As imagens às quais se irá aplicar o algoritmo de compressão são de formato QCIF, contendo 180×144 pixels de resolução da luminância e duas componentes de crominância de 90×72 pixels, conforme se pode observar na figura 3.3.

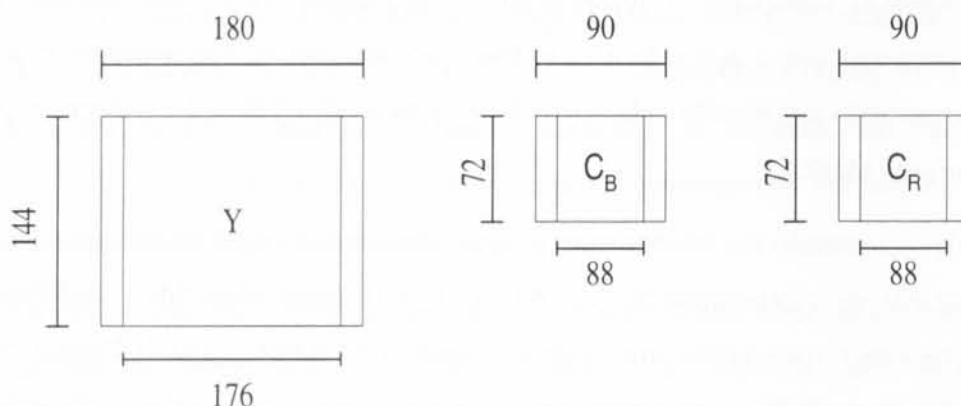


fig. 3.3 - Formato QCIF

O sistema orientado para processamento e análise de imagem utilizado obedece, em termos gerais ao esquema que está representado na figura 3.4.

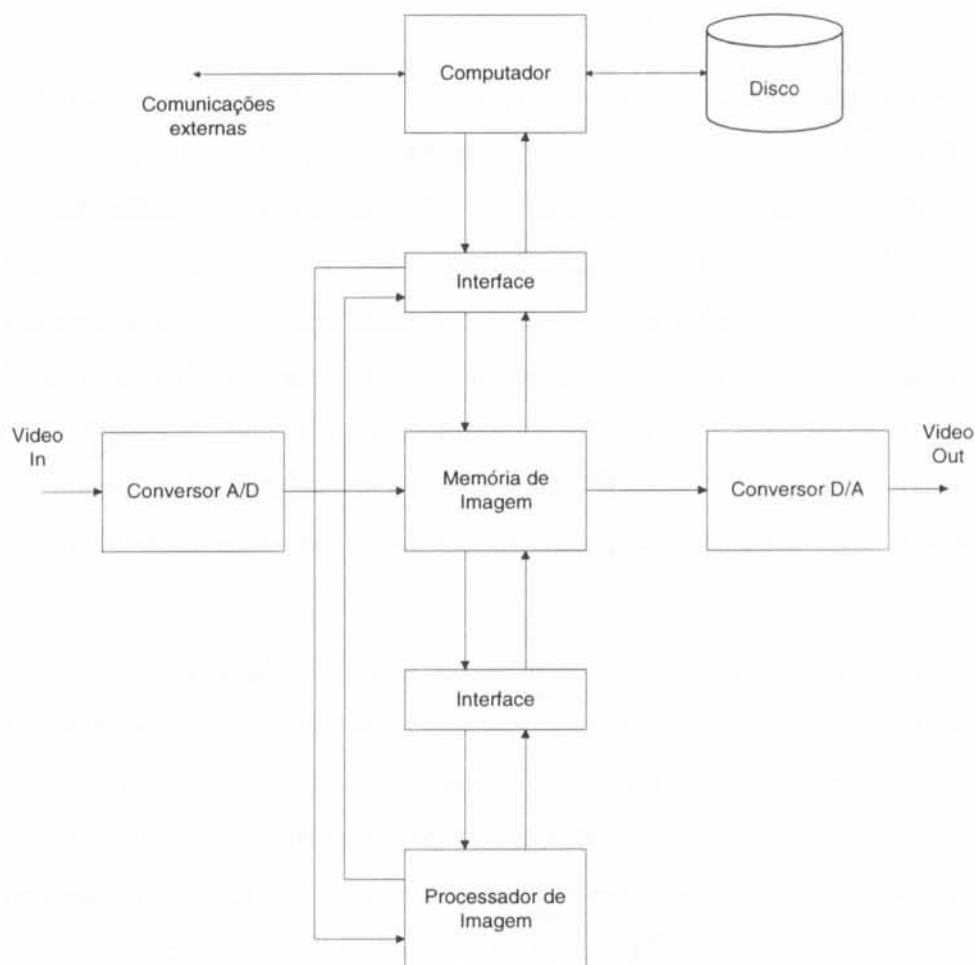


fig. 3.4 - Sistema de compressão de imagem

No receptor, tem-se um outro sistema em que não existe obviamente o módulo de aquisição. O módulo de processamento de imagem é facultativo, quer no codificador quer no decodificador, podendo o computador encarregar-se dos algoritmos de compressão.

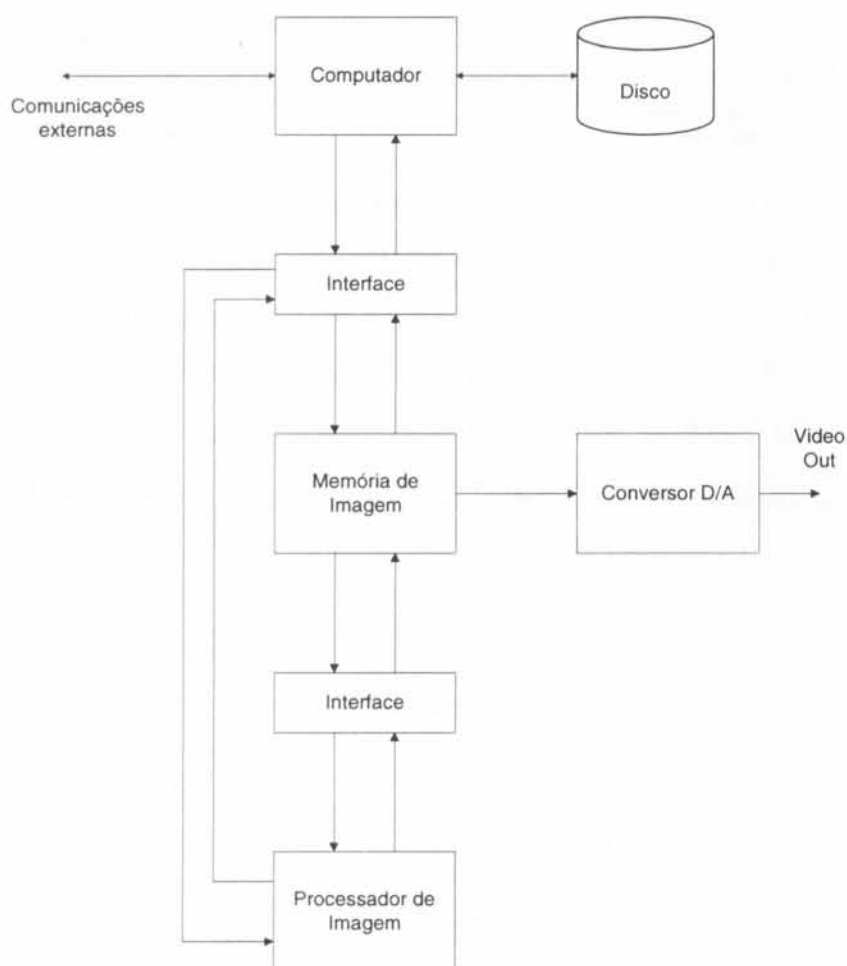


fig. 3.5 - Sistema de decodificação de imagem

Na implementação do algoritmo base, bem como das modificações introduzidas, o processador utilizado foi um TMS320C30 da Texas Instruments, processador digital de sinal de elevado desempenho, com uma capacidade de processamento de 33.3 MFLOPS (Milhões de Operações em Vírgula Flutuante por Segundo) e 16.7 MIPS (Milhões de Instruções por Segundo), sendo portanto indicado para este tipo de aplicações de processamento de vídeo em tempo real [TeI88].

Em seguida, far-se-á o estudo de cada um dos módulos constituintes do algoritmo base [Mar92].

3.2 Segmentação de imagem

A aplicação da quantificação vectorial na compressão de imagens necessita de vectores de dados, sobre os quais a operação é realizada. A definição desses vectores deve permitir explorar a redundância existente quer no plano espacial quer no plano temporal. Como se observou anteriormente, a redundância espacial é melhor explorada tomando como dados de entrada uma matriz, correspondente a um bloco numa zona específica da imagem. A redundância temporal existente em sequências de imagens poderá ser reduzida se se considerarem várias imagens simultaneamente, e daí extrair a informação relevante. Um método simples e eficaz consiste em efectuar a diferença entre duas imagens consecutivas, permitindo deste modo que, para as zonas inalteradas, não seja necessário afectar bits de codificação.

A operação de segmentação divide a imagem a codificar em blocos de determinada dimensão, sobre os quais se aplicam as técnicas de compressão. Uma possível partição da imagem consiste em dividi-la em blocos de dimensão pré-determinada e fixa. As dimensões dos blocos determinam a qualidade resultante da compressão, bem como a taxa de compressão atingida. Blocos de grandes dimensões permitem taxas de compressão mais elevadas, mas em contrapartida a compressão é de implementação mais complexa. Por outro lado, blocos de dimensões menores, não permitindo taxas elevadas de compressão, são mais indicados para zonas onde existe maior actividade. Existe por isso um compromisso a resolver, entre a dimensão do bloco e o resultado obtido da compressão.

A segmentação de imagem em blocos de dimensão fixa é um método simples, mas que carece de naturalidade na divisão. A divisão da imagem em blocos de dimensão fixa não permite, de um modo geral, a separação de zonas de diferentes graus de actividade. É portanto normal que esteja contido num bloco parte de uma zona de baixa actividade e parte de outra com elevada actividade. Desta forma, não é possível a codificação com diferentes graus de resolução para as diferentes zonas de actividade.



fig. 3.6 - Segmentação de imagem em blocos de dimensão fixa

A desvantagem de uma classificação não natural, como é a segmentação em blocos de dimensão fixa, leva a considerar a possibilidade da realização da segmentação consistindo na divisão da imagem em regiões não regulares. Neste tipo de segmentação, não é pretendido a divisão da imagem em blocos, mas sim a separação dos objectos que a imagem contém.



fig. 3.7 - Segmentação de imagem em regiões não regulares

É, pois, possível com este tipo de segmentação, aplicar diferentes algoritmos de codificação a diferentes objectos da imagem; pode, por exemplo, codificar-se com elevada resolução as pessoas presentes na cena, não dando tanta atenção aos objectos. Esta separação de codificação em diversos graus de resolução é desejável, por exemplo, em sistemas de videotelefone, onde é pretendido que os interlocutores sejam codificados com elevada precisão, sobre um cenário de fundo que não é relevante. Este tipo de segmentação da imagem em regiões não regulares é principalmente usada em sistemas de compressão baseada em objectos. As razões que foram apontadas anteriormente para a não utilização da compressão baseada em objectos (complexidade de implementação e tempo de processamento), aplicam-se igualmente para a justificação da não utilização deste tipo de segmentação.

Pretende-se, portanto, uma segmentação com um desempenho maior do que aquele que é obtido pela segmentação em blocos de dimensão fixa, não podendo, devido às restrições de hardware utilizado, utilizar a segmentação em regiões não regulares. Uma solução intermédia consiste na segmentação de imagem em blocos de dimensão variável. Neste tipo de algoritmos de segmentação, locais da imagem com um elevado índice de actividade são segmentados em blocos de reduzida dimensão; os blocos de maiores dimensões são reservados a zonas de reduzida actividade. A baixa complexidade de implementação deste algoritmo relativamente à segmentação em regiões não regulares, bem como o melhor desempenho em relação à segmentação em blocos de dimensão fixa, satisfaz o compromisso entre desempenho e complexidade que é desejado.

A segmentação da imagem em blocos de dimensão variável é executada tendo em atenção a actividade de cada zona da imagem. Uma possível medida de actividade corresponde à variância do bloco, definida como a diferença entre a média quadrática e o quadrado da média dos elementos do bloco

$$Var = M_{quadrática} - M^2 = \frac{1}{n} \sum_{i=1}^n (x_i)^2 - \left| \frac{1}{n} \sum_{i=1}^n x_i \right|^2$$

Estabelecendo um limiar para a variância, classifica-se um bloco como pouco activo se estiver abaixo desse limite ou como activo se for igual ou ultrapassar o valor definido.

Os algoritmos de segmentação de imagem em blocos de dimensão variável, sendo a divisão da imagem mais apropriada à realidade, conduzem a uma melhor compressão e qualidade da imagem final. Têm, no entanto, a desvantagem de obrigar à transmissão de bits suplementares de descrição da segmentação, para que o receptor tenha conhecimento da dimensão do bloco codificado e assim ser possível a sua correcta localização.

Dos vários algoritmos desenvolvidos para este tipo de segmentação, destaca-se a *segmentação estruturada em árvores quaternárias* [VaG87][VaG92], no qual a imagem é inicialmente dividida em blocos quadrados e recursivamente subdivididos se a medida de actividade ultrapassar um determinado limiar.

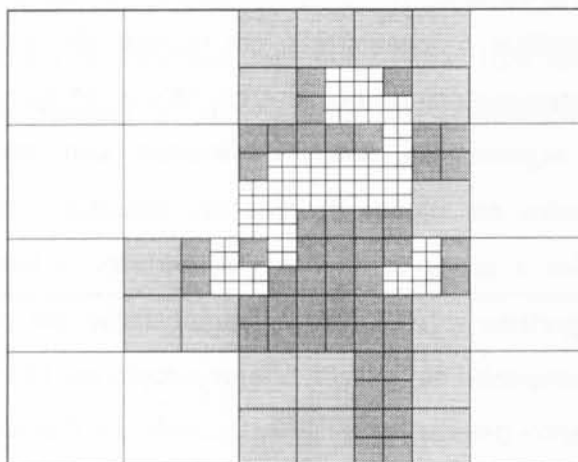


fig. 3.8 - Segmentação estruturada em árvores quaternárias

Outra classe de segmentação corresponde às técnicas não estruturadas, em que são utilizados blocos de dimensões e forma variáveis. De entre estas técnicas, a *segmentação não estruturada em um passo* [CRA90] procura que a um determinado local seja atribuído o maior bloco possível, sem que se atinja o limiar de actividade.

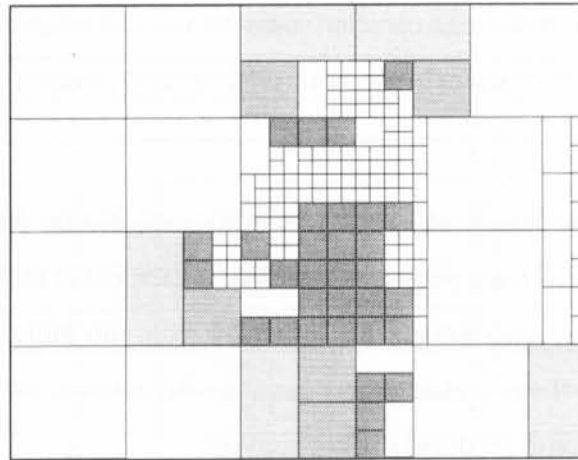


fig. 3.9 - Segmentação não estruturada em um passo

Outra das técnicas não estruturadas é o algoritmo de *segmentação não estruturada em múltiplos passos* [BoL90]. O procedimento consiste em colocar em primeiro lugar os blocos de maiores dimensões, num primeiro varrimento, colocando sucessivamente os blocos menores nas passagens posteriores. Embora com uma visão mais global da imagem e, portanto, com uma melhor distribuição dos tipos de blocos, esta técnica implica obviamente uma maior complexidade de implementação e maior tempo de processamento.

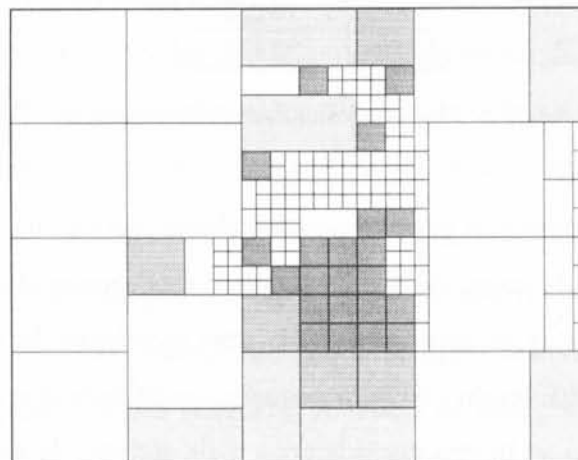


fig. 3.10 - Segmentação não estruturada em múltiplos passos

O algoritmo eleito para o codificador base deste trabalho, foi o da segmentação não estruturada em um passo, devido às características de natural adaptabilidade às diferentes zonas da imagem e grau de complexidade de implementação compatível com o desejado.

Nas imagens segmentadas foram utilizados onze tipos de blocos de diferentes formas e dimensões, desde os blocos mais pequenos de 4 por 4 elementos classificados com maior grau de actividade, até aos macroblocos de 32 por 32, com um índice de actividade muito baixo. Os blocos do tipo 0 são considerados como muito activos; os blocos de 4 por 4 do tipo 1 são classificados como moderadamente activos.

Tipo	Dimensão
0	4x4
1	4x4
2	8x4
3	4x8
4	8x8
5	16x8
6	8x16
7	16x16
8	32x16
9	16x32
10	32x32

tabela 3.1 - Tipos de blocos usados na segmentação

Este conjunto de blocos possíveis permite uma excelente partição da imagem, consoante o nível de actividade contida nesta. Na tabela seguinte está representado a percentagem do número de blocos resultante da segmentação de uma sequência de 32 imagens, com um limiar de actividade de 80, sendo visível a pequena quantidade dos blocos de dimensões mais elevadas em relação aos blocos considerados mais activos, de menores dimensões. É, ainda, apresentada a percentagem da área da imagem que cada tipo de bloco ocupou durante a sequência de teste.

Tipo	nº (%)	Área (%)
0	20.7	11.8
1	62.0	35.5
2	3.3	3.8
3	8.8	10.0
4	1.8	4.1
5	0.8	3.6
6	1.6	7.2
7	0.2	2.0
8	0.3	5.7
9	0.2	3.9
10	0.3	12.4

tabela 3.2 - Resultados da segmentação de 32 imagens da sequência "Miss America"

3.3 Módulo de quantificação vectorial

A codificação da imagem, no algoritmo base que foi considerado, é executada pela quantificação vectorial. Como foi já referido, a quantificação vectorial dos blocos consiste em fazer corresponder a um vector de dados, o vector da tabela de códigos mais próximo, de acordo com uma dada medida de distorção. A tabela de códigos usada foi obtida a partir das sequências de treino "Miss America" e "Claire" usando o algoritmo de Linde-Buzo-Gray.

A utilização de tabelas de códigos estruturadas em árvore foi a opção adoptada neste algoritmo base, uma vez que oferece um bom compromisso entre complexidade de implementação e qualidade obtida [GrA82]. A tabela de códigos é constituída por uma série de andares estruturados em árvore, como se pode observar na figura seguinte.

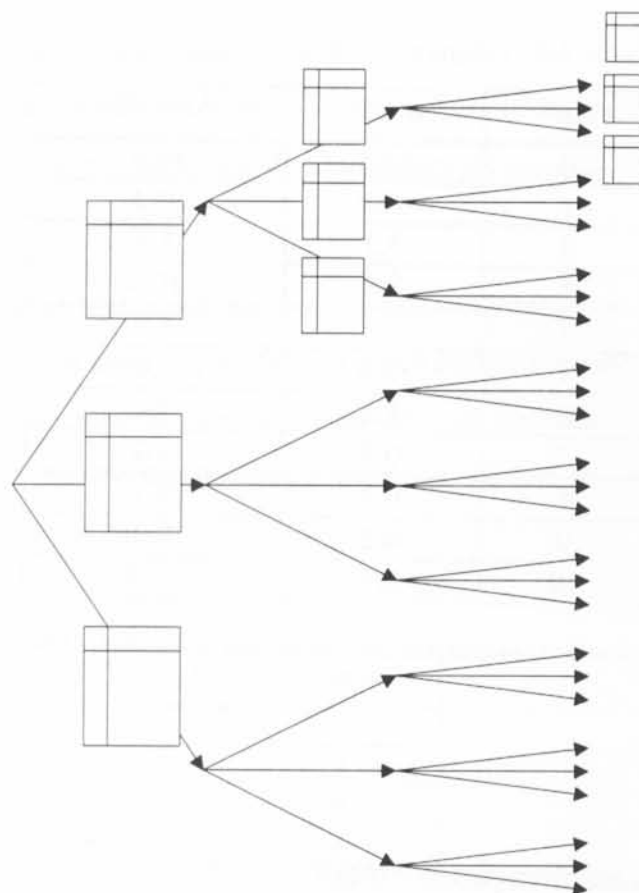


fig. 3.11 - Exemplo de uma tabela de códigos estruturada em árvore

Este método de estruturação permite que, em cada andar da árvore, o vector seja quantificado por uma sub-tabela de códigos, pertencente à tabela de códigos global desse nível da árvore. Este tipo de estruturação das tabelas de código permite que a busca do vector de código se efectue de uma forma simplificada.

A medida de distorção adoptada na procura do vector de código foi a da média do erro quadrático. Utilizou-se o desdobramento desta medida de forma a reduzir o número de operações necessárias. Assim, sendo a formula da média do erro quadrático

$$D = \sum_{j=1}^K (x_j - x_j^*)^2$$

na qual, x_j representa o elemento j do vector de dimensão K a codificar e x'_j o elemento j do vector da tabela de códigos, o desdobramento conduz a

$$D = \sum_{j=1}^K x_j^2 + 2 \cdot \sum_{j=1}^K (x_j \cdot x'_j) + \sum_{j=1}^K x_j'^2$$

A primeira parcela corresponde a uma constante, pois representa a soma do quadrado de cada elemento do vector a codificar. Pode, portanto, ignorar-se esse somatório. Por outro lado, a última parcela corresponde à soma dos quadrados dos elemento de cada vector da tabela de códigos. É, desta forma, possível guardar em memória estes valores, pois são constantes para cada tabela de códigos utilizada. Este desdobramento da fórmula permite reduzir de forma significativa o tempo de cálculo da medida de distorção.

A quantificação vectorial é aplicada a blocos de dimensão e forma variáveis resultantes da segmentação não-estruturada a um passo. Devido à enorme complexidade da implementação de quantificadores vectoriais para cada um dos onze tipos de blocos, foi introduzido uma operação de decimação antes do módulo de quantificação. Esta decimação permite a redução dos onze tipos de blocos a duas classes: a classe dos blocos de 4×4 e a classe dos blocos de 8×4 (ver fig. 3.12)

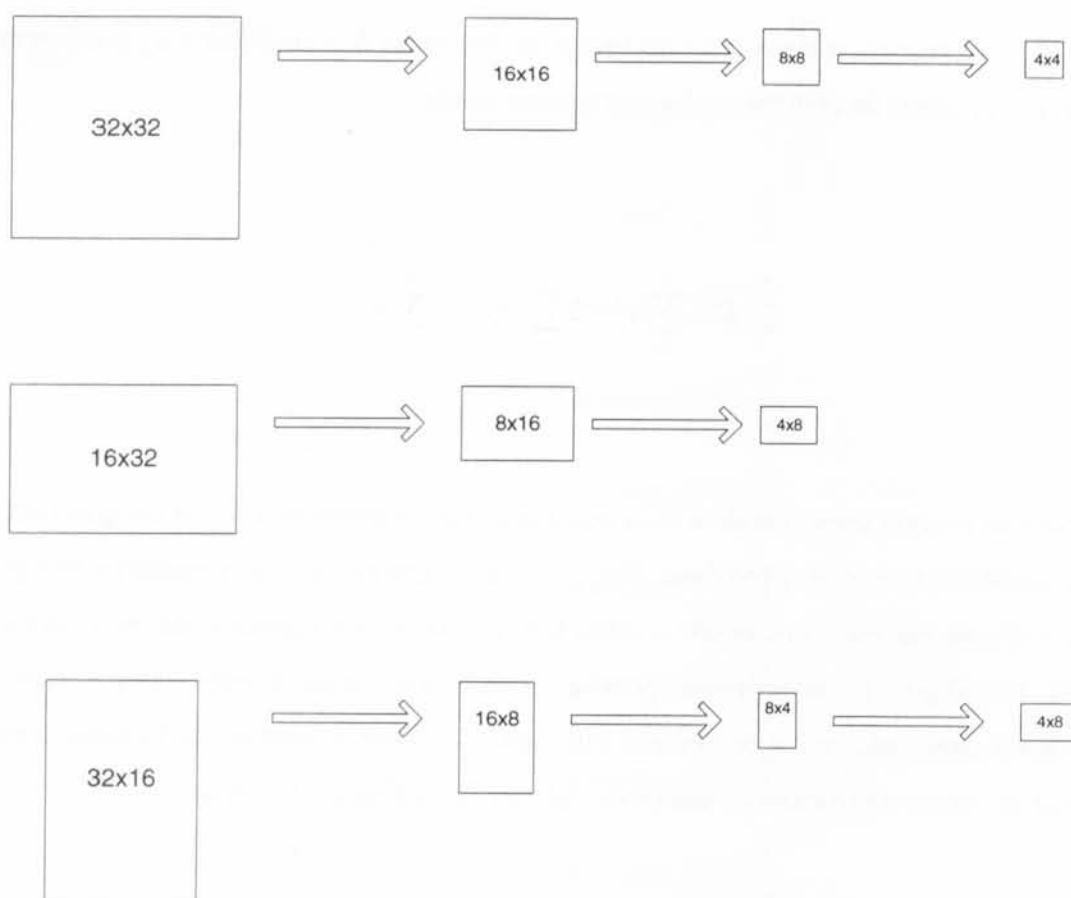


fig. 3.12 - Decimação dos blocos de maiores dimensões

Para restituir a dimensão e forma original do bloco, após a decodificação o bloco é interpolado para a dimensão inicial.

Ficou, desta forma, definido o algoritmo base, algoritmo este que foi o ponto de partida para as melhorias e alterações descritas no próximo capítulo.

Capítulo 4

Técnicas de Codificação de Vídeo Para Baixas Cadências

4.1 Introdução

No capítulo anterior foi descrito o algoritmo que servirá de ponto de partida para os estudos realizados nesta tese. Pretende-se, neste capítulo, explorar outros métodos de codificação, de modo a reduzir a complexidade de implementação, bem como aumentar a taxa de compressão e a qualidade da imagem reconstruída. Esta qualidade é quantificada através da relação sinal-ruído (SNR), mas é dada especial atenção à qualidade subjectiva da imagem. Entende-se por qualidade subjectiva a classificação de maior ou menor distorção da imagem sentida por um observador humano. A importância da qualidade

subjectiva deve-se ao facto de, geralmente, pequenas variações da relação sinal-ruído, não corresponderem de forma precisa às variações sentidas pelo observador.

O primeiro passo foi o de substituir o módulo de quantificação vectorial por um outro tipo de codificação, com o intuito de reduzir a complexidade de implementação. Posteriormente, o aumento da taxa de compressão foi o objectivo perseguido com a introdução de um módulo de compensação de movimento. Na secção devotada ao módulo de identificação de fundos, apresenta-se uma técnica que permite aumentar a taxa de compressão e a qualidade subjectiva da imagem simultaneamente. Na última secção, pretende aumentar-se a qualidade da imagem através da codificação da cor.

4.2 Codificação de imagem por padrões visuais

4.2.1 Introdução

O elevado desempenho dos quantificadores vectoriais, no que respeita à taxa de compressão e qualidade de imagem, foi o ponto de partida do algoritmo base. O atractivo de um método que combinasse as qualidades da quantificação vectorial e, de alguma maneira, superasse os seus pontos fracos, como a elaboração da tabela de códigos e posterior procura, foi o incentivo para a pesquisa de algoritmos alternativos.

Um dos métodos que pareceu viável e que respeitou os requisitos impostos foi a *codificação de imagem por padrões visuais* (*Visual Pattern Image Coding* ou VPIC) descrito em [ChB90]. Esta técnica baseia-se no aproveitamento dos factos conhecidos do sistema visual do observador, tentando obter taxas de compressão elevadas, sem que a complexidade dos algoritmos aumente exponencialmente. Este método de compressão, explora os conhecimentos que têm vindo a ser adquiridos sobre o processamento visual da informação.

Embora o conhecimento do processamento efectuado pelo sistema visual humano ainda seja vago, existem certos fundamentos tidos como certos que, conquanto de “baixo nível”, permitem elaborar um modelo de visão. A subdivisão da imagem em regiões de luminosidade praticamente constante e locais de transições bruscas, está de acordo com o modelo básico de visão. É necessário, então, efectuar uma triagem dos blocos em blocos contínuos ou uniformes e blocos descontínuos ou de transição, aplicando um algoritmo de compressão diferente a cada um deles. Naturalmente, os blocos em que ocorram mudanças bruscas de transição deverão ser tratados de forma a que se preserve a informação neles contida. Desse modo, e de forma similar à quantificação vectorial, será elaborado um conjunto de *padrões* que formará um dicionário. Deste, o codificador efectuará a selecção de um padrão para cada bloco de transição a ser processado.

A construção dos padrões deverá obedecer às características do sistema visual humano. Estando a tratar-se de padrões que contêm variações de luminância, as características importantes para o seu estudo são as que concernem à resposta em frequência dos receptores visuais. A banda a que esses receptores são sensíveis, sendo considerado um olho humano de características “normais”, é medida em ciclos por grau. Considera-se o intervalo de 1-10 ciclos/grau como referência, sendo no entanto a resposta mais elevada entre os 3 e 6 ciclos/grau. Isto significa que um determinado indivíduo, consoante a sua posição em relação à imagem a receber, só poderá distinguir um certo número de variações de luminosidade ocorridas numa zona específica da imagem.

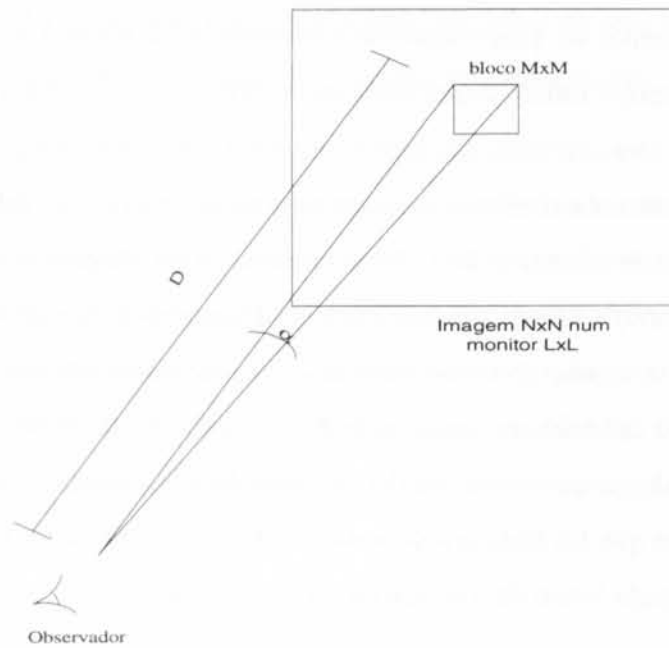


fig. 4.1 - Geometria de visualização

Na figura anterior está representado um bloco a codificar com $M \times M$ elementos de uma imagem de $N \times N$ pixels, apresentada num monitor com as dimensões de $L \times L$ centímetros a um observador situado a uma distância de D centímetros. Se se considerar que D é bastante superior a L , pode afirmar-se que

$$\alpha \approx 180 \cdot \text{tg}^{-1} \left(\frac{LM}{ND} \right)$$

e, afectando valores considerados comuns para cada variável, como sejam $M=4$, $N=512$, $L=21.5$ cm, e $D=200$ cm, obtém-se um valor para α de 0.15° . A este valor de α corresponde, se mantivermos os valores de 1 a 10 ciclos/grau de banda, 0.15-1.5 ciclos [ChB90].

Com a geometria descrita acima, um observador poderá distinguir apenas uma transição de luminosidade ocorrida num bloco com as dimensões de 4×4 pixels. Desta forma, os padrões considerados como blocos de transição apenas irão conter uma mudança de luminosidade, obedecendo ao modelo concebido para o sistema visual humano.

Nesta fase de concepção, apenas serão considerados os blocos de 4 por 4 elementos, efectuando posteriormente a extensão a blocos de maiores dimensões.

4.2.2 Dicionário de padrões

O método da construção dos padrões consiste, ao contrário do que é feito na quantificação vectorial em que o receptor humano não é um factor decisivo na elaboração do conjunto de vectores, no estabelecimento de padrões subjectivamente significativos, de acordo com o modelo de visão considerado. Os blocos de imagem a codificar são divididos em blocos de transição e blocos uniformes; estes últimos são completamente definidos pela extracção da média do bloco.

Definindo a imagem a codificar como uma matriz $I = \{ I_{ij} \}$, a qual é a união dos diversos blocos de 4 por 4

$$b_{i,j} = \{ I_{n,m} : 4i \leq n \leq 4i+3, 4j \leq m \leq 4j+3 \}$$

a média pode ser calculada por

$$Média = \frac{1}{16} \sum_{n=1}^4 \sum_{m=1}^4 I_{n,m} \quad I_{n,m} - \text{pixel do bloco}$$

Como foi já referido, é imposto aos blocos de transição a condição de apenas conterem um variação de luminosidade. Mas, mesmo com esta restrição, os blocos de transição poderão apresentar diversas configurações. É, deste modo, necessário limitar o seu número aos que são considerados mais expressivos.

Uma medida considerada natural para a quantificação da variação ocorrida num bloco de transição é o gradiente. O gradiente é calculado por

$$\nabla I = (\Delta_x I, \Delta_y I)$$

onde cada componente direccional do gradiente, $\Delta_x I$ e $\Delta_y I$, é obtida da seguinte forma

$$\Delta_x I = \{ \Delta_x b_{i,j} \}$$

em que

$$\Delta_x b_{i,j} = \text{Média}\{ I_{n,m} : 4i+2 \leq n \leq 4i+3, 4j \leq m \leq 4j+3 \} - \\ \text{Média}\{ I_{n,m} : 4i \leq n \leq 4i+1, 4j \leq m \leq 4j+3 \}$$

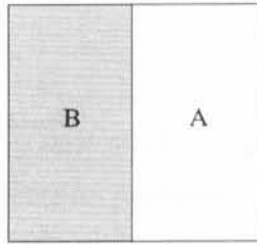
e

$$\Delta_y I = \{ \Delta_y b_{i,j} \}$$

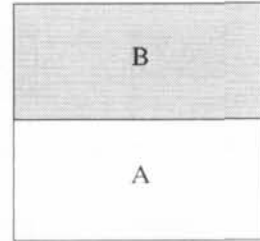
com

$$\Delta_y b_{i,j} = \text{Média}\{ I_{n,m} : 4i \leq n \leq 4i+3, 4j+2 \leq m \leq 4j+3 \} - \\ \text{Média}\{ I_{n,m} : 4i \leq n \leq 4i+3, 4j \leq m \leq 4j+1 \}$$

Esta definição de gradiente direccional corresponde à diferença das médias dos elementos vizinhos, no interior do mesmo bloco, segundo a orientação adequada. Deste modo, $\Delta_x b_{i,j}$ calcula a diferença entre as médias das duas últimas colunas e a média das duas primeiras. O mesmo é feito para $\Delta_y b_{i,j}$, considerando desta vez a diferenças das médias dos dois grupos de linhas.

Componente Horizontal do
Gradiente

$$|\Delta_x| = \text{Média}(A) - \text{Média}(B)$$

Componente Vertical do
Gradiente

$$|\Delta_y| = \text{Média}(A) - \text{Média}(B)$$

fig. 4.2 - Cálculo das componentes direccionais do gradiente

Partindo dos valores de gradiente para um bloco pode calcular-se a magnitude e a orientação da transição nele contida da seguinte forma

$$\text{Magnitude} \quad |\nabla b|_{i,j} = \sqrt{(\Delta_x b_{i,j})^2 + (\Delta_y b_{i,j})^2}$$

$$\text{Orientação} \quad \angle \nabla b_{i,j} = \text{tg}^{-1} \frac{\Delta_y b_{i,j}}{\Delta_x b_{i,j}}$$

O bloco a codificar terá uma magnitude do gradiente e uma orientação da transição que permitirá a sua classificação num dos padrões do conjunto elaborado. Foram considerados quatro tipos de orientações, correspondentes a incrementos de 45 graus, como se pode ver na figura 4.3.

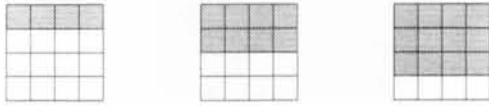
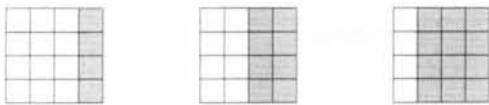
90°45°0°-45°

fig. 4.3 - Padrões quadrados com as orientações de 90°, 45°, 0° e -45°

As restantes quatro orientações de 135°, 180°, 225° e 270° são definidas por uma indicação de polaridade, permitindo obter o inverso de cada padrão. A cada elemento dos padrões, são atribuídos valores de modo a ficarem completamente definidos. Na figura 4.3, a representação dos valores é feita graficamente, correspondendo a cor branca a um valor, e a cor cinzenta a outro. A determinação dos valores é feita tendo em consideração que cada padrão tem média zero e magnitude do gradiente unitária, isto é

$$\sum_n p_i^n = 0 \quad , p_i \in \{ \text{padrões} \}$$

e

$$|\nabla p_i| = 1$$

A imposição destas duas condições permite que se efectue a separação inicial da média do bloco, e da magnitude do gradiente a codificar.

4.2.3 Algoritmo de codificação

O processo de codificação é suficientemente simples para que seja possível a sua implementação em tempo real em hardware pouco poderoso. Ao bloco considerado é extraído a média μ , e são calculados os valores da magnitude do gradiente $|\nabla b|^2$ e o ângulo de orientação $tg(\angle \nabla b)$. A razão do cálculo do quadrado da magnitude e da tangente do ângulo, permite minimizar o tempo dispendido no processamento de codificação. Deste modo, é desnecessária a determinação da raiz quadrada e do arco-tangente, operações essas que não são de trivial implementação em programação.

O quadrado da magnitude do gradiente permite estabelecer uma fronteira entre blocos uniformes e de transição. Se o valor obtido for menor ou igual a um determinado limiar, o bloco será classificado como uniforme e processado como tal.

$$|\nabla b|^2 \leq |\nabla I_{min}|^2 \quad \Rightarrow \text{Bloco uniforme}$$

$$|\nabla b|^2 > |\nabla I_{min}|^2 \quad \Rightarrow \text{Bloco de transição}$$

Com o conhecimento até agora adquirido, pode transmitir-se o tipo de bloco e a sua média e, se este for uniforme, processar o bloco seguinte. No caso de ser um bloco de transição, tem-se seguidamente de o mapear num sub-espaco de codificação. O valor obtido para a tangente do ângulo permite a classificação do bloco num dos sub-espacos com a mesma orientação, como se pode verificar na figura seguinte.

Orientação	$\text{tg} (\angle \nabla b)$
-45°	$[-67.5^\circ, -22.5^\circ] \Rightarrow [-2.41, -0.41]$
0°	$[-22.5^\circ, 22.5^\circ] \Rightarrow [-0.41, +0.41]$
45°	$[22.5^\circ, 67.5^\circ] \Rightarrow [0.41, 2.41]$
90°	$[67.5^\circ, 112.5^\circ] \Rightarrow [2.41, +\infty[$ $\Rightarrow]-\infty, -2.41]$

tabela 4.1 - Mapeamento do valor da tangente na orientação principal correspondente

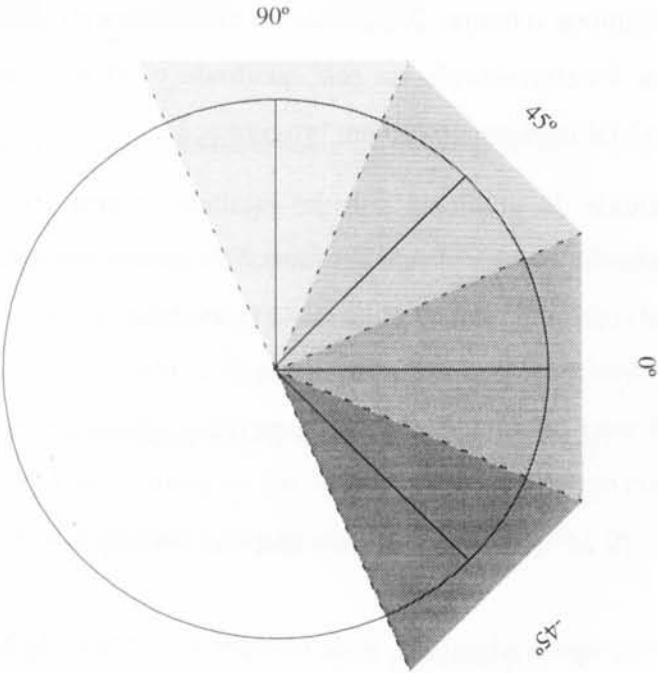


fig. 4.4 - As orientações principais

Os blocos cujo ângulo da transição esteja contido num intervalo de $\pm 22.5^\circ$ centrado nas orientações principais consideradas ($-45^\circ, 0^\circ, 45^\circ, 90^\circ$), são classificados nesse sub-espço. Por esta razão, se se considerar o caso de um bloco cujo valor obtido no cálculo da tangente esteja contido num intervalo entre -2.41 e -0.41 , este corresponderá a uma orientação da transição entre -67.5° e -22.5° ; será, então, classificado como um bloco cujo sub-espço de orientação é de -45° .

No sub-espço seleccionado, existem diversos blocos a serem considerados (ver figura 4.3). O método de escolha do padrão que melhor representa o bloco é feito pela correlação entre o bloco a codificar e os padrões desse sub-espço. Se, para esta função de correlação, um valor elevado corresponde a uma maior semelhança entre o bloco e o padrão, a selecção é realizada procurando o máximo da correlação. O padrão que melhor representa o bloco a codificar é escolhido, sendo transmitido o índice que lhe corresponde. Um outro parâmetro a enviar é o da polaridade, o qual permite obter as orientações complementares, correspondendo a $135^\circ, 180^\circ, 225^\circ$ e 270° . É apenas necessário um bit de informação para notificar o receptor destas orientações. Esta polaridade é obtida verificando se a correlação corresponde a um máximo positivo ou negativo. Se o valor da função de correlação entre o bloco a codificar e o padrão atinge um máximo de magnitude mas de valor negativo, isso significa que o bloco é de orientação complementar da representada pelo padrão.

Está-se, agora, em posição de determinar o número de bits necessários para a codificação de cada tipo de bloco. Concluindo, após a codificação de um bloco, serão gerados

$$n_u = 1 + n_{\mu u} \quad \text{bits}$$

para os blocos do tipo uniforme, em que $n_{\mu u}$ corresponde ao número de bits necessário para a codificação da média e o bit suplementar necessário para indicar o tipo de bloco.

Para blocos de transição, o número de bits resultante da compressão é

$$n_e = 1 + n_{\mu e} + n_p + n_g + 1 \quad \text{bits}$$

Sendo n_{μ} a quantidade de bits para codificação da média, n_p o número de bits para indicar o índice do padrão seleccionado, n_g o número de bits para a codificação da magnitude do gradiente, um bit para a polaridade e outro para classificar o bloco como de transição.

4.2.4 Algoritmo de descodificação

O sistema usado permite uma descodificação simples e com um tempo de processamento muito baixo, pois o descodificador apenas terá que contemplar os dois tipos de blocos existentes, e as operações aritméticas e lógicas efectuadas são de muito baixa complexidade.

Os blocos do tipo uniforme são processados requerendo apenas a recepção da média, dado que esta define completamente aquela classe de blocos. A multiplicação do gradiente pelo padrão a que o índice recebido se refere, com a polaridade devida, e a posterior adição da média que foi transmitida, permite a descompressão dos blocos de transição. Uma simples selecção do padrão através de um índice, uma operação de multiplicação e uma adição é o suficiente para os blocos do tipo mais complexo, ou seja, os de transição.

4.2.5 Análise de complexidade

O algoritmo usado tem como vantagem evidente a baixa complexidade de codificação e descodificação, sendo por isso possível o seu uso em tempo real em hardware pouco poderoso. O primeiro passo neste sistema é o cálculo da média do bloco sendo, para um bloco de 4x4 pixels, necessário 15 adições e uma multiplicação para a efectuar. A obtenção da magnitude do gradiente bem como do valor da tangente pode ser feita em conjunto com a média que, devido a manipulações e simplificações de cálculos, eleva o valor de operações para 20 adições e 3 multiplicações.

Após a quantificação dos valores acima descritos, procede-se à determinação do índice do padrão, mapeando o valor obtido da tangente do ângulo do gradiente numa das quatro classes assumidas, correspondentes às orientações principais (-45°, 0°, 45° e 90°). Este

mapeamento necessita, no pior caso, de três comparações em vírgula flutuante para que o bloco fique classificado. Seguidamente, dentro de cada classe o padrão é definitivamente eleito através da contagem e comparação dos elementos de valor positivo ou negativo do bloco com os dos padrões, conseguindo-se isso com 16 adições e 16 comparações, para o pior caso.

A quantificação do valor da magnitude do gradiente em M níveis necessita de, no máximo, $\log_2 M$ comparações e, deste modo, o total de operações efectuadas na compressão por este sistema pode ser apresentado na tabela 4.2. Devido a que o número de níveis M do gradiente é geralmente muito baixo, como será referido numa secção posterior, as comparações necessárias à sua determinação não foram consideradas, para efeitos do total de operações.

Média, Gradiente e Tangente	20 adições e 3 multiplicações
Índice do Gradiente	$\leq \log_2 M$ comparações
Índice do padrão	3 comparações
Polaridade	16 adições e 16 comparações (pior caso)
Total	36 adições, 3 multiplicações, 19 comparações

tabela 4.2 - Análise de complexidade

Como se pode observar, a complexidade deste algoritmo de compressão é, de facto, muito baixa, permitindo um excelente desempenho do processador no que se refere ao tempo de processamento. Um dos critérios enunciados no início deste capítulo - baixa complexidade de implementação - é atingido com este algoritmo. A condição de uma taxa de compressão elevada pode ser genericamente estipulada, tomando em conta os valores do número de bits requerido para cada tipo de bloco, obtendo-se

$$\frac{128}{n_e} \leq T_{\text{Compressão}} \leq \frac{128}{n_u}$$

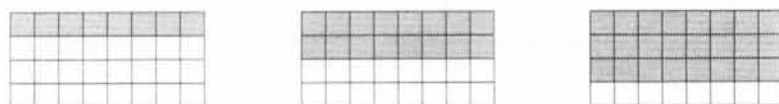
sendo o numerador o número de bits necessário para transmitir um bloco de 4 por 4 pixels sem que se efectue algum tipo de compressão. Dependendo dos valores para cada denominador, a taxa de compressão irá variar, sendo no entanto usual atingir resultados de

$$10 \leq T_{\text{compressão}} \leq 20$$

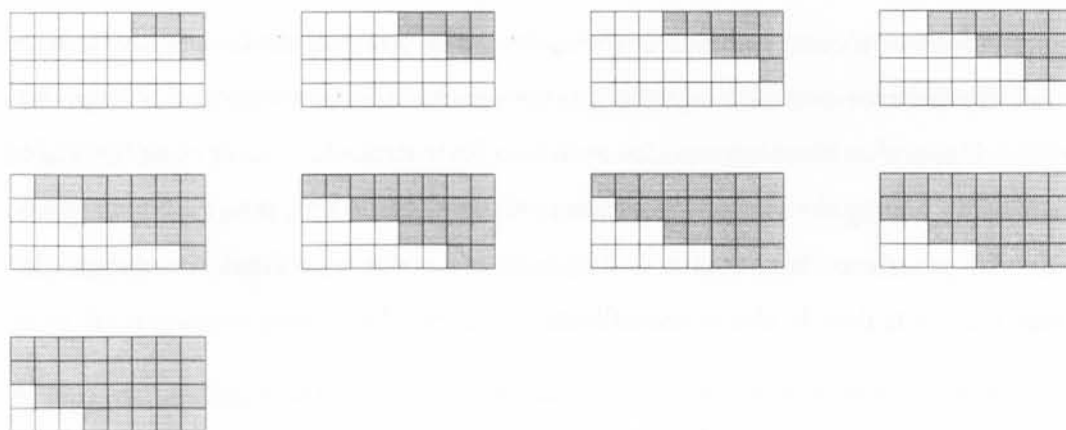
4.2.6 Aplicação do algoritmo a blocos de dimensão e forma variáveis

Embora o algoritmo descrito em [ChB90] apenas contemple blocos de dimensão 4x4, uma extensão a blocos de maiores dimensões permitiria a obtenção de taxas de compressão ainda mais elevadas. Um primeiro alongamento a blocos rectangulares de proporções 4x8 foi elaborado, resultando num novo conjunto de padrões que se representam na figura seguinte.

90°



45°



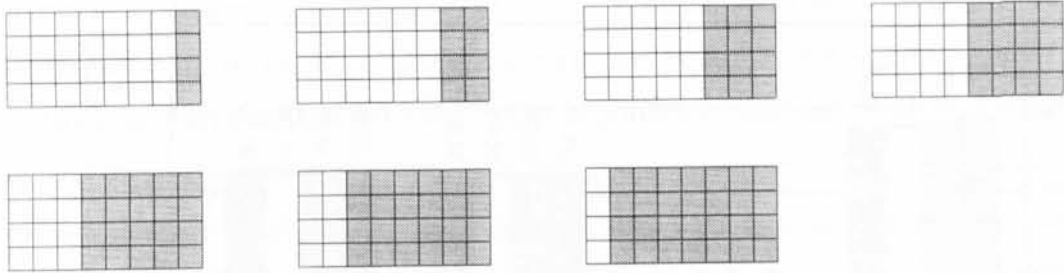
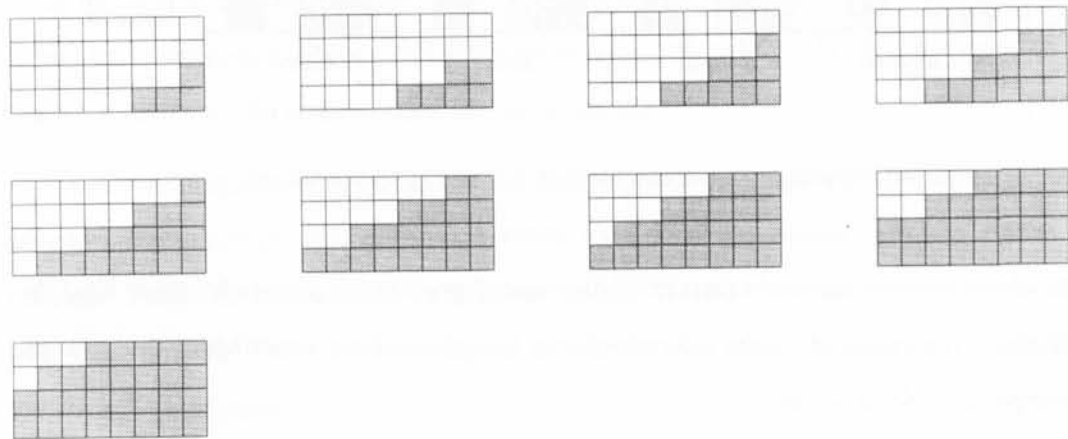
0°  -45° 

fig. 4.5 - Padrões rectangulares

Foi estudada a eventual extensão a blocos de ainda maiores dimensões, efectuando uma decimação e posterior interpolação dos blocos, semelhante à descrita no capítulo 3 quando da apresentação do algoritmo base. A decimação permite o uso de blocos maiores, reduzindo o número dos seus elementos para blocos quadrados de 4×4 , ou rectangulares de 4×8 . O uso da operação de interpolação, restitui à dimensão original, o bloco que foi codificado. Pretende-se a melhoria da taxa de compressão, não comprometendo a qualidade da imagem codificada.

Foram testadas várias sequências de imagens como a sequência “Miss America”, “Alice”, “Claire” e uma outra sequência de treino obtida, denominada “INESC”, permitindo visualizar o desempenho do algoritmo para os vários tipos de blocos.

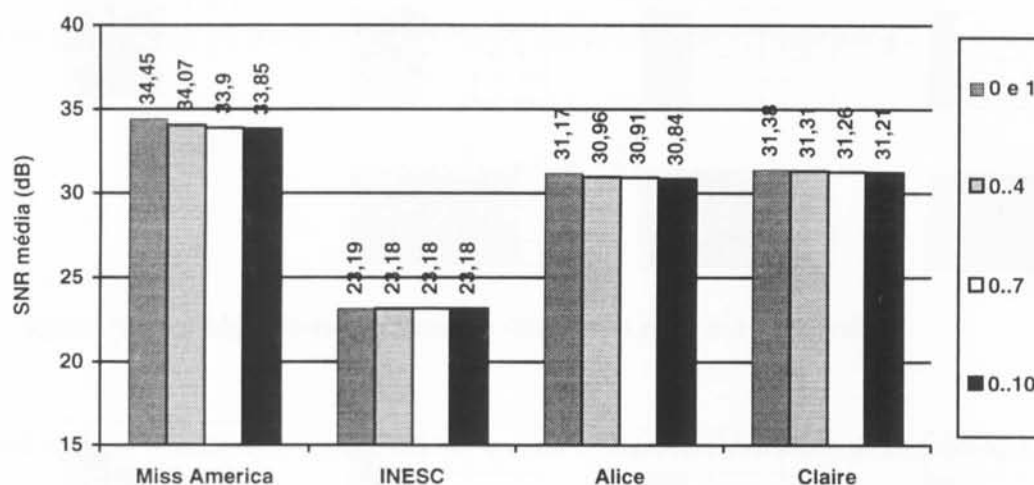


fig. 4.6 - Resultados obtidos para blocos de dimensão e forma variáveis

A pequena variação da relação sinal/ruído indica que é possível a utilização deste tipo de blocos sem que exista uma evidente degradação da imagem e com a vantagem de maiores taxas de compressão do sistema.

4.2.7 Implementação e análise dos resultados obtidos

O módulo do quantificador vectorial existente no codificador básico foi substituído pelo módulo de VPIC, cujo algoritmo foi descrito na secção anterior. O ajustamento de certos parâmetros do algoritmo permitirá a calibração do desempenho deste codificador. Seguidamente, apresenta-se o estudo das variáveis do algoritmo, bem como as implicações ao nível dos resultados obtidos.

4.2.7.1 Estudo da magnitude do gradiente

Inerente ao processo de compressão VPIC, a distinção entre dois grupos de blocos, uniformes e de transição, é necessária para que cada um seja sujeito a uma forma de codificação diferente. Devido à sua maior importância, será dada uma maior atenção aos blocos nos quais existe uma transição brusca de luminância. Estes blocos deverão ser cuidadosamente codificados, pois são os que transportam informação de maior importância para o observador. A imagem, sujeita à segmentação, é aplicada ao módulo de compressão e este executa a classificação dos blocos em uniformes e de transição.

Essa escolha é executada calculando a magnitude do gradiente, comparando-a com um limiar e classificando o bloco como uniforme se o valor do gradiente for menor ou igual ao limiar, e de transição se este valor for ultrapassado.

O valor da magnitude do gradiente é subjectivamente significativo, pois a um valor elevado corresponde uma diferença entre as linhas e colunas de cada bloco também elevada. Pode, desta maneira, determinar-se um elevado número de possíveis transições que ocorram no bloco e classificá-lo em bloco de transição com o seu esquema de codificação adequado.

A determinação do valor mínimo para a magnitude, $|\nabla I|_{min}$, determina o número de blocos de cada tipo e influencia a qualidade da imagem final. Este limiar pressupõe que, valores menores ou iguais da magnitude do gradiente, não são subjectivamente significativos para o receptor, ou seja, o bloco encontra-se numa zona em que a luminância é constante ou que a sua variação é tão suave que não é perceptível.

O limiar mínimo da magnitude do gradiente que é perceptível pelo receptor humano, pode ser obtido por calibração por parte do receptor, ou pode ser calculado com base na relação sinal-ruído e nas estatísticas das sequências de imagem. Importa, pois, avaliar a magnitude do gradiente em sequências de imagens de teste, para ser possível o estabelecimento de um valor de limiar de acordo com estatísticas representativas da realidade.

Para que fosse possível a determinação do melhor valor para o limiar, foi calculada a magnitude do gradiente para cada bloco segmentado. Para o efeito, a segmentação de imagem foi executada em blocos de 4 por 4 elementos. A utilização de blocos de pequenas

dimensões permite determinar com rigor as pequenas mudanças de luminância existentes na imagem. Os resultados obtidos demonstram que, blocos contendo magnitude de gradiente moderada, são em número elevado. A observação do gráfico da figura [seguinte] mostra ainda que, acima do valor 30 para a magnitude do gradiente, a frequência de ocorrência é muito baixa.

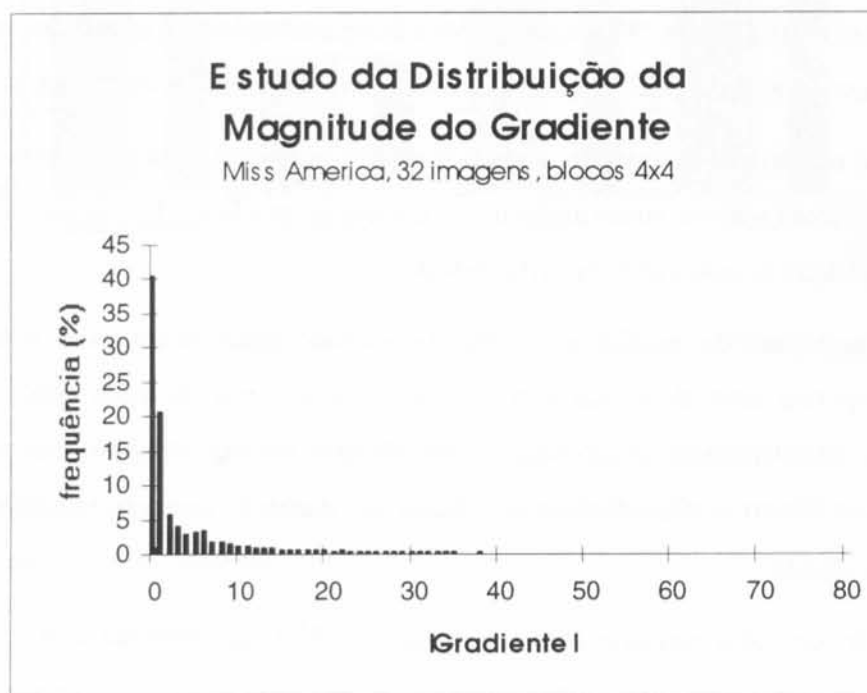


fig. 4.7 - Resultados do estudo da magnitude do gradiente (sequência "Miss America")

A sequência "Miss America" contém vastas zonas em que a intensidade da luminosidade é constante ou sem alterações significativas, contribuindo para que exista um grande número de blocos em que a magnitude do gradiente é zero ou muito baixa. Sequências de imagens em que tal não se verifique, determina um outro resultado, talvez mais de acordo com o que se passa em situações reais. Uma das sequências de treino usada, "INESC", consistindo num cenário de escritório e possuindo movimento moderado, na qual existem variações pequenas mas constantes da iluminação, sendo desta forma bastante ruidosa, permite um estudo mais aproximado de uma situação real. Os resultados proporcionados por essa sequência, na qual a distribuição da magnitude do gradiente aproxima uma

distribuição de Rayleigh, estão de acordo com o que é mencionado em [ChB90], e podem ser observados na figura seguinte.

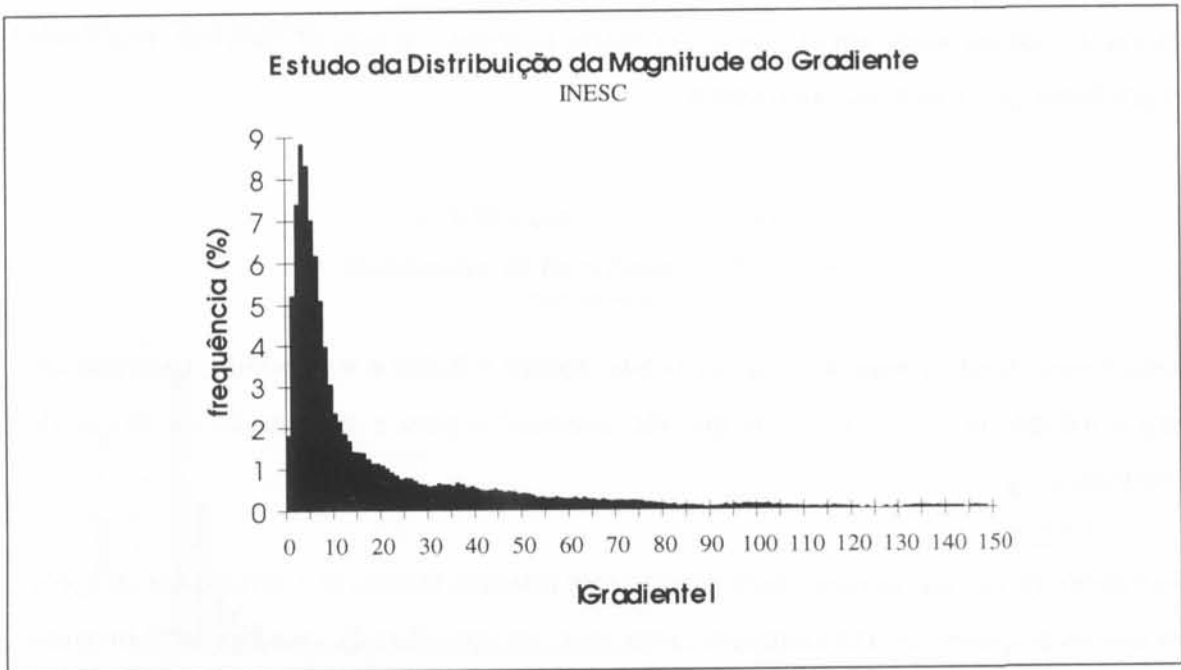


fig. 4.8 - Resultados do estudo da magnitude do gradiente (sequência "INESC")

Como se pode observar na figura 4.8, blocos de magnitude zero ocorrem com baixa frequência, elevando-se esta para magnitudes moderadas e decrescendo para elevadas magnitudes. Ou seja, existem poucos locais da imagem nos quais a luminosidade é constante, existindo um valor máximo da frequência quando a magnitude do gradiente atinge o valor três, decrescendo a frequência a partir deste valor e por uma longa gama de magnitudes.

É bem patente, quer na primeira quer na última sequência de teste, que não tem qualquer utilidade considerar-se valores de magnitude acima de um certo valor máximo, uma vez que o número de blocos em que tal ocorre é extremamente baixo. Isto possibilita que a magnitude seja quantificada num número restrito de níveis, com vista à transmissão do resultado da compressão. Define-se então um limite máximo, $|\nabla I|_{max}$, sendo os valores da magnitude do gradiente acima desse valor truncados para $|\nabla I|_{max}$.

O intervalo das magnitudes do gradiente considerado é então

$$R = |\nabla I|_{\max} - |\nabla I|_{\min}$$

em que R é quantificado em M níveis. Por outras palavras, existem M níveis da magnitude do gradiente que podem ser codificados,

$$|\nabla I|_i \quad i=0,\dots,M-1$$

Desta forma, sendo a magnitude quantificada, apenas o índice é transmitido, contribuindo para a redução do número de bits que são necessários para a codificação de blocos de transição.

No método de cálculo da magnitude do gradiente referido, realiza-se a divisão das imagens em blocos pequenos de 4x4 elementos, sem levar em consideração relações inter-imagens no plano temporal. O resultado da segmentação é transmitido ao módulo de compressão VPIC e este, conhecida a posição do bloco a codificar, procede à determinação do gradiente, retirando os valores da imagem originada pelo sistema de aquisição.

A vantagem de se considerar para bloco de codificação a diferença entre a imagem actual e a precedente consiste na redução da redundância temporal, sendo esperado valores médios mais baixos para magnitude do gradiente. Neste caso, após a aquisição da imagem a codificar, a diferença entre a nova imagem e a imagem anterior será objecto de codificação. Do mesmo modo, a operação de segmentação é executada sobre a imagem diferença. Na descodificação, a imagem reconstruída é adicionada à imagem anterior que foi já descodificada.

A diferença das imagens permitirá diminuir a redundância existente no plano temporal, diminuindo também o intervalo R entre o valor para o qual a magnitude do gradiente é considerado mínimo e o valor assumido como limite máximo. Esta redução de variância é desejável pois permite a quantificação da magnitude do gradiente num menor número de níveis.

Nas figuras seguintes, podem constatar-se as vantagens da utilização das imagens diferença, pois é visível a translação de valores, que anteriormente eram elevados, para mais baixas magnitudes.

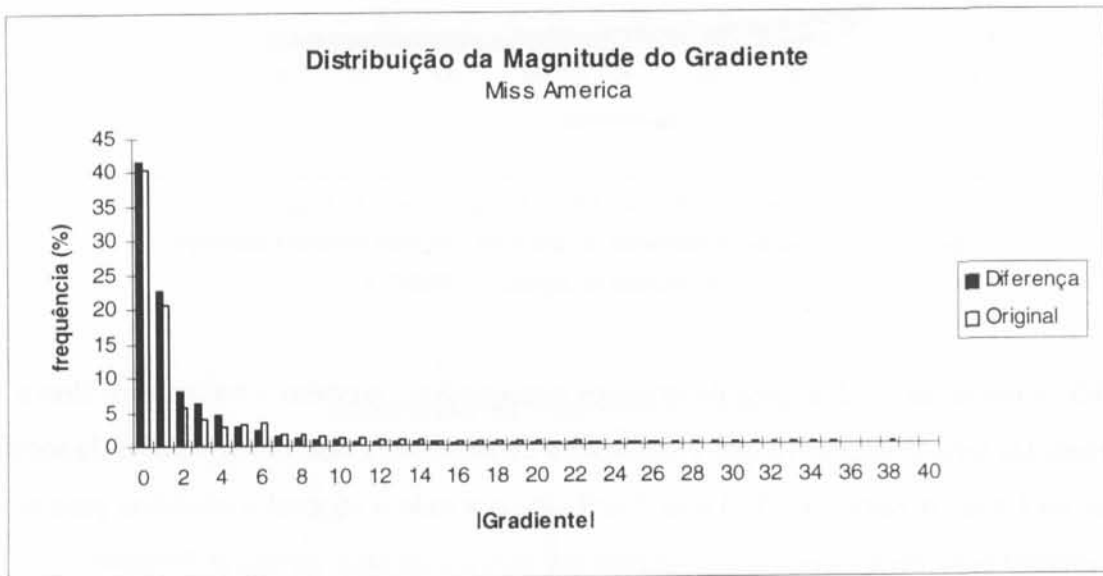


fig. 4.9 -Distribuição da magnitude do gradiente - imagens original e diferença
(32 imagens da sequência "Miss America")

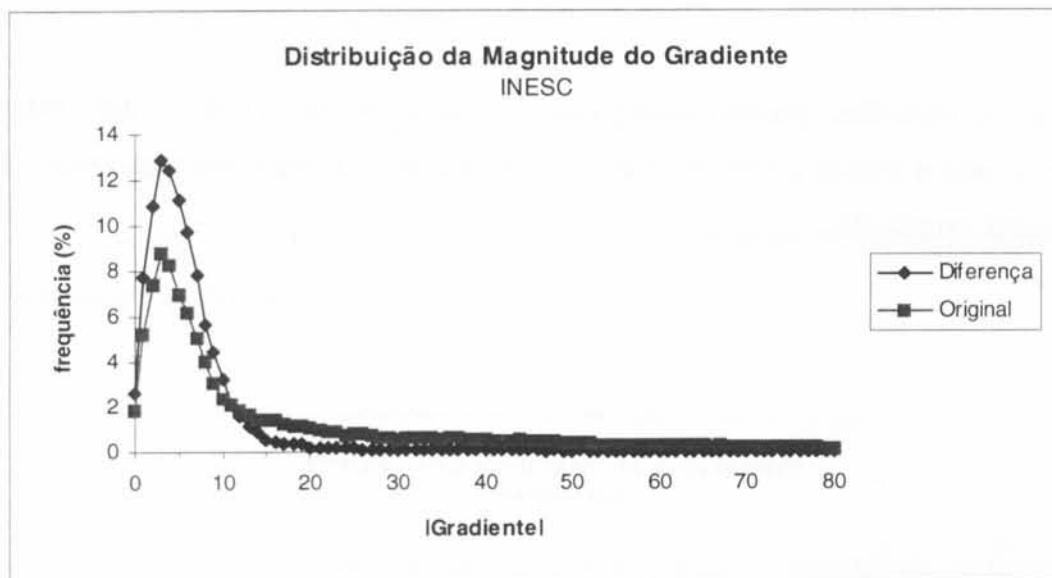


fig. 4.10 - Distribuição da magnitude do gradiente - imagens original e diferença
(100 imagens da sequência "INESC")

Devido à operação de diferença de imagens consecutivas, também a média dos blocos é alterada. De uma situação em que a frequência de ocorrência dos vários valores da média possíveis (entre o valor 0 e 255) está distribuída por todo o intervalo, evolui-se para uma distribuição próxima da gaussiana, centrada em zero e com uma variância diminuta.



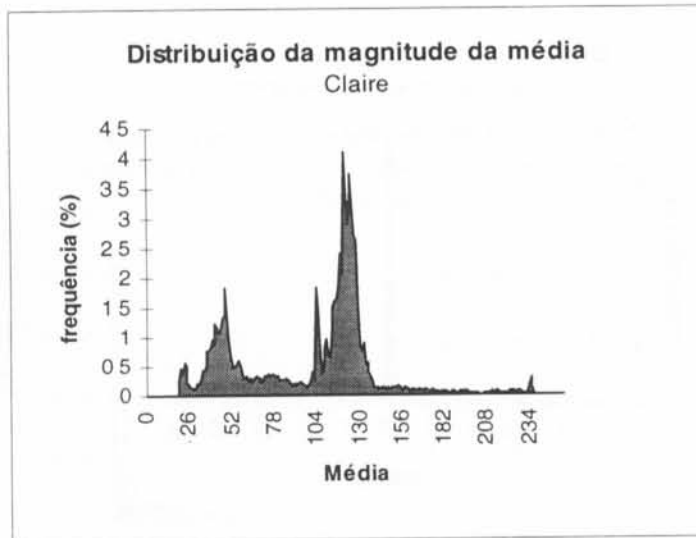
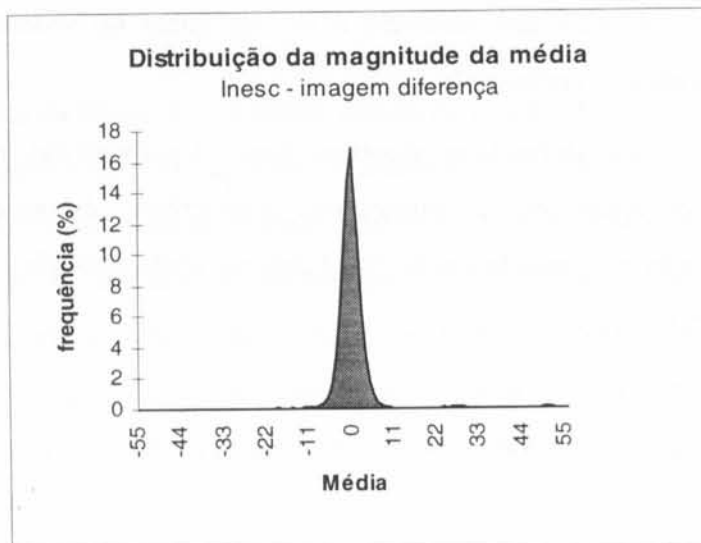


fig. 4.11 - Distribuição da média para imagens originais



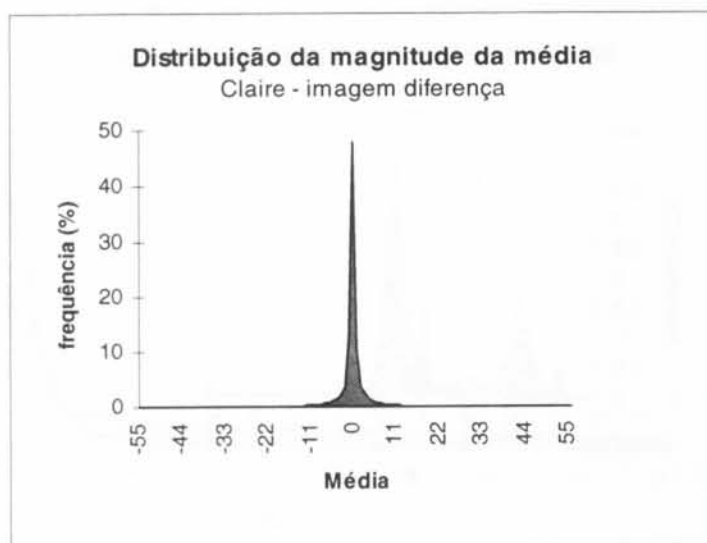


fig. 4.12 - Distribuição da média para imagens diferença

A diminuição da gama na qual a média estava contida é igualmente proveitosa, dado que a codificação da média é outra das variáveis a ter em conta no sistema, contribuindo positivamente para a taxa de compressão.

A aplicação da segmentação definida no algoritmo base - a partição das imagens em onze tipos básicos de blocos quadrados e rectangulares já descritos - é considerada uma boa solução, e foi estudado o comportamento do algoritmo nessa situação, relativamente à magnitude do gradiente.

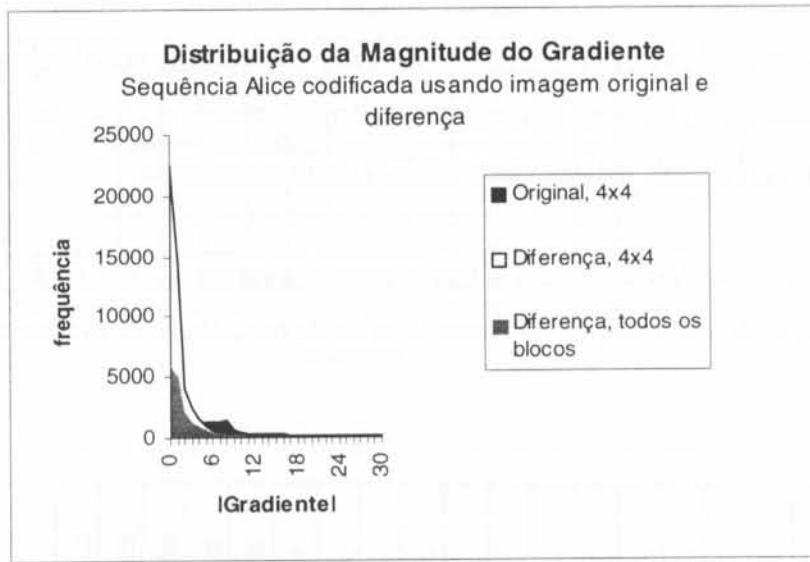


fig. 4.13 - Distribuição da magnitude do gradiente para vários tipos de segmentação

É visível no gráfico da figura 4.13 o ganho obtido pela redução da redundância através da diferença de imagens consecutivas sobre a codificação de imagens originais. Embora a frequência dos blocos da imagem original seja transitoriamente inferior, para magnitudes mais elevadas torna-se, sem dúvida, bastante maior, contribuindo para uma vasta gama de magnitudes a quantificar. Outra das conclusões a que se pode chegar é a vantagem evidente do uso da segmentação de imagem com blocos de forma e dimensão variável, conseguindo atingir-se metade do número de blocos em relação à segmentação com blocos de proporções fixas.

A diminuição da redundância devido à utilização da diferença das imagens e consequente menor amplitude do intervalo R dos limites do gradiente, assim como a redução da amplitude da média, sugerem a aplicação da codificação sobre imagens diferença, no sentido de serem obtidos melhores resultados para a taxa de compressão do sistema.

Embora a escolha do valor de $|\nabla I|_{\max}$ possa ser realizada observando atentamente os gráficos, já a determinação de $|\nabla I|_{\min}$ terá de ser mais cuidadosa, devido às fortes

implicações na qualidade da imagem reconstruída, assim como no número de blocos de cada tipo daí resultantes. Um $|\nabla I|_{min}$ baixo resultará numa maior quantidade de blocos de transição, aumentando assim o número de bits necessários à reconstrução da imagem, pois, como foi já referido anteriormente, o número de bits necessário à codificação de blocos de transição é significativamente superior à dos blocos do tipo uniforme.

Na figura seguinte pode observar-se a relação entre o número de blocos de cada tipo - uniforme e de transição - em função do limite inferior da magnitude do gradiente.

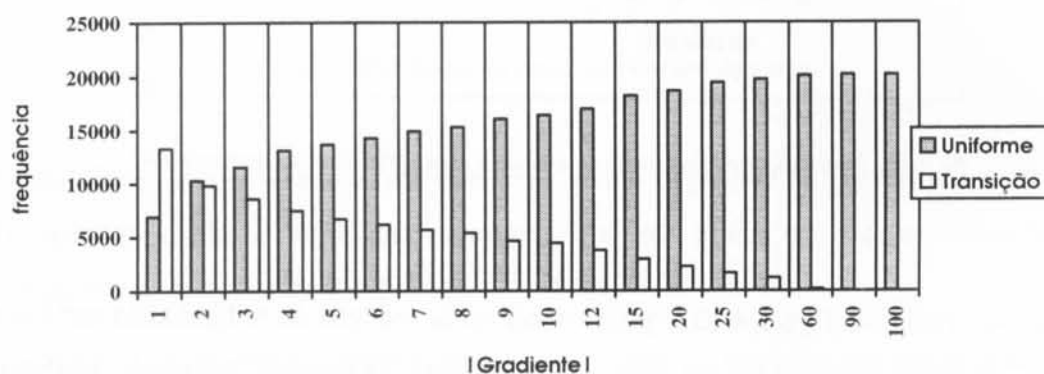


fig. 4.14 - $|\nabla I|_{min}$ vs. número de blocos uniformes e de transição

Adicionalmente, o valor de $|\nabla I|_{min}$ afectará a relação sinal-ruído da imagem final. A relação sinal-ruído aumenta quando existe um maior número de blocos de transição, devido a uma melhor representação de pequenas transições existentes nos blocos da imagem.

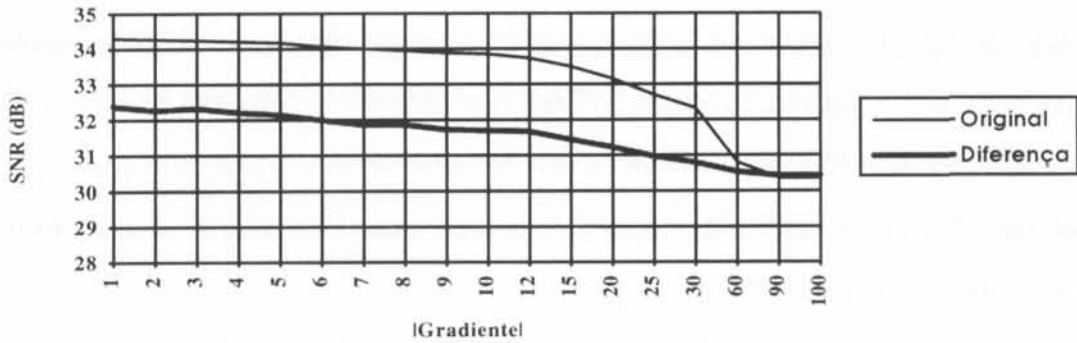


fig. 4.15 - $|\nabla I|_{\min}$ vs. relação sinal-ruído (sequência "Miss America")

A discrepância dos valores da relação sinal-ruído obtidos para as imagens original e diferença, observável na figura 4.15, deve-se ao facto de, na imagem diferença, existir um acumular de erros de quantificação na reconstrução, erros esses devidos essencialmente aos blocos de transição, nos quais são mais elevados. No caso em que o elevado valor de $|\nabla I|_{\min}$ não permite a classificação de nenhum bloco a codificar como bloco de transição, a relação sinal-ruído atinge um valor mínimo devido a só existirem blocos do tipo uniforme, aproximando-se a codificação de imagens originais à codificação de imagens diferença.

A escolha de $|\nabla I|_{\min}$ e $|\nabla I|_{\max}$ permite, portanto, influenciar várias características do funcionamento do sistema, como a taxa de compressão, uma vez que o limite mínimo influencia o número de blocos de cada tipo e o valor máximo define o intervalo de quantificação da magnitude do gradiente, e a qualidade da imagem reconstruída que é determinada pelo valor de $|\nabla I|_{\min}$, impondo uma determinada relação sinal-ruído.

O uso da imagem diferença nos blocos a codificar parece ser o mais adequado a este esquema de compressão, dada a menor amplitude de R , tendo sido assim utilizada nos estudos a seguir realizados. A obtenção do valor da taxa de compressão só é completamente definida estipulando o número de bits para cada parcela da codificação dos dois tipos de blocos, sendo este o objectivo do estudo seguinte.

4.2.7.2 Codificação dos blocos de transição e uniformes

Devido à menor dimensão do intervalo $R = |\nabla I|_{\max} - |\nabla I|_{\min}$, no caso de se utilizarem imagens diferença, é possível quantificar a magnitude do gradiente com um menor número de níveis do que é necessário para codificar directamente as imagens originais. Esta técnica traz vantagens do ponto de vista da taxa de compressão.

Foram definidos vários valores de limite e obtidos os valores da relação sinal-ruído para cada quantificador do gradiente

nº bits do Gradiente	nº níveis	$ \nabla I _{\min}$	$ \nabla I _{\max}$	SNR (dB)
0	1	5		32.72
		10		32.02
1	2	5	30	32.76
		10	30	32.29
2	4	5	30	32.81
		10	30	32.42
3	8	5	30	32.92
		10	30	32.60
5	32	5	30	32.92
		10	30	32.60

tabela 4.3 - Número de bits de codificação do gradiente vs. SNR

Na tabela 4.3, é visível que o número de níveis de quantificação da magnitude do gradiente não é de capital importância para o desempenho global do sistema, embora o seu aumento contribua para uma pequena melhoria da relação sinal-ruído. Este facto é referido em [ChB90], onde o número de bits para a codificação do gradiente é citado como de segundo plano em relação ao processo de codificação.

Para se determinar a taxa de compressão dos dois tipos de blocos, é necessário ainda avaliar o impacto da quantificação da média de cada bloco na relação sinal-ruído, de forma a concluir qual o número de bits necessário para que a qualidade da imagem reconstruída seja aceitável e com uma taxa de compressão elevada. À quantificação da média, no caso de blocos uniformes, deverão ser atribuídos mais bits do que à quantificação dos blocos de transição. A razão desta diferença, deve-se ao facto de o sistema visual humano ser menos

sensível a distorções em zonas com variações bruscas de luminosidade. Deste modo, em locais em que a luminância é constante, se não existir uma quantificação cuidadosa, resultará o chamado *efeito de bloco*. Este efeito consiste na variação perceptível do valor de luminosidade de blocos vizinhos em zonas consideradas uniformes.

Na figura seguinte apresentam-se os resultados obtidos em termos de relação sinal-ruído, para dois intervalos de quantificação e para vários valores do número de bits da média.

Sequência	Intervalo	nº bits média bloco uniforme	nº bits média bloco transição	média SNR (dB)
INESC	[-127;+128]	5	5	22.63
	[-63;+64]	5	5	22.67
		5	4	22.23
		4	4	22.14
		5	3	21.06

Sequência	Intervalo	nº bits média bloco uniforme	nº bits média bloco transição	média SNR (dB)
Claire	[-127;+128]	5	5	28.03
	[-63;+64]	5	5	28.64
		5	4	28.20
		4	4	27.47
		5	3	26.82

Sequência	Intervalo	nº bits média bloco uniforme	nº bits média bloco transição	média SNR (dB)
Alice	[-127;+128]	5	5	29.03
	[-63;+64]	5	5	29.08
		5	4	28.67
		4	4	27.98
		5	3	27.50

tabela 4.4 - Codificação da média em dois intervalos distintos de quantificação

Foi considerada uma quantificação uniforme da média em intervalos de $[-63,+64]$ e $[-127,+128]$, com 8, 16 e 32 níveis. Embora o último intervalo possibilite uma mais rápida adaptação a variações bruscas de intensidade, como é o caso das primeira e segunda imagens, o intervalo de menores dimensões possibilita uma maior relação sinal-ruído e melhor qualidade subjectiva.

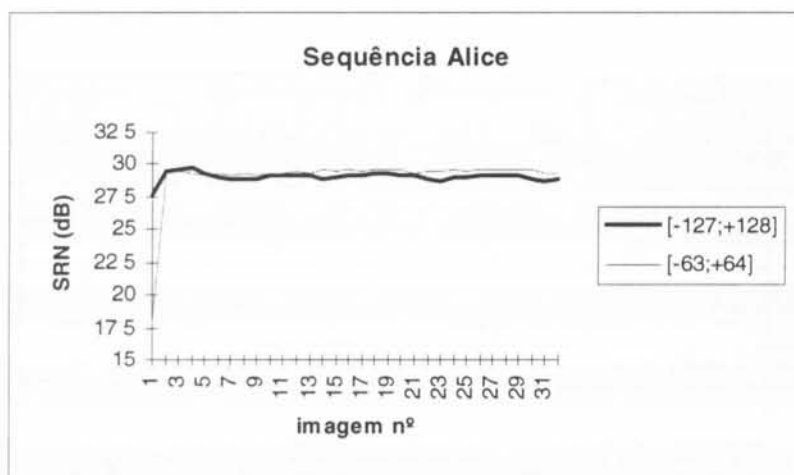


fig. 4.16 - Relação sinal-ruído para dois intervalos de quantificação

Aquele resultado é natural, pois um intervalo de maiores dimensões conduz a que o valor da média do bloco a codificar seja mais elevado e, portanto, é mais indicado para zonas da imagem em que ocorram maiores flutuações daquele valor. O interesse da sua utilização decai, no entanto, se atendermos à degradação da representação dos pequenos valores da média que, como é notório na figura 4.12, constituem a maioria das ocorrências. Concluindo, um intervalo de quantificação de menores dimensões permite atingir valores de relação sinal-ruído melhores tendo, por contrapartida, a desvantagem na quantificação de variações maiores. O caso das variações maiores acontece apenas nas primeiras imagens ou em cortes de cena.

Foi considerado como um resultado de qualidade subjectiva boa os resultantes da codificação da média de blocos uniformes e de transição com, respectivamente, 5 e 4 bits. A quantificação com valores inferiores, como (4,4) e (5,3), tem por resultado imagens finais com um elevado efeito de bloco, não sendo por isso aceitável.

Após o estudo precedente, são conhecidos todos os elementos necessários para eleger um possível sistema de codificação que utilize VPIC. O número de bits para a codificação dos parâmetros deste sistema foram estipulados permitindo, assim, o cálculo da taxa de compressão final. Essa quantidade é, no entanto, variável. Está dependente da sequência de imagens utilizada e, por conseguinte, do número de ocorrências de cada um dos onze tipos de blocos quadrados e rectangulares gerados pelo segmentador, bem como do número de blocos classificados como uniformes ou de transição. Uma forma de se quantificar, de um modo geral, a taxa de compressão resultante, consiste em calcular a compressão alcançada para cada um dos onze tipos de blocos do segmentador.

Valores(bits)	tipo	média	gradiente	índice padrão	polaridade	Total (bits)
Bloco (tipo)						
Uniforme	1	5	-	-	-	6
Transição	1	4	0	4	1	10

tabela 4.5 - Atribuição do número de bits a cada tipo de bloco

Bloco	Dimensões	nº elementos	Codificação Bloco Uniforme (bits/pixel)	Codificação Bloco de Transição (bits/pixel)	Taxa de Compressão
0,1	4x4	16	0.375	0.625	12,8:1 < Tx < 21,3:1
2,3	8x4 , 4x8	32	0.1875	0.3125	25,6:1 < Tx < 42,6:1
4	8x8	64	0.09375	0.15625	51,2:1 < Tx < 85,3:1
5,6	16x8 , 8x16	128	0.046875	0.078125	102,4:1 < Tx < 170,6:1
7	16x16	256	0.0234375	0.0390625	204,8:1 < Tx < 341,3:1
8,9	32x16 , 16x32	512	0.01171875	0.01953125	409,6:1 < Tx < 682,6:1
10	32x32	1024	0.00585937	0.00976562	819,2:1 < Tx < 1365:1

tabela 4.6 - Taxa de compressão obtida

Uma possível atribuição de bits a cada valor a ser quantificado está descrito na tabela 4.6, onde se verifica que o número de bits necessários à codificação de um bloco uniforme é de seis, sendo dez os bits que um bloco de transição requer. A taxa de compressão final está

sumariada na tabela 4.6, na qual os blocos de maiores dimensões conduzem, naturalmente, a uma maior compressão. No entanto, este tipo de blocos são os que ocorrem com menor frequência. A taxa de compressão final deverá aproximar-se da gama obtida para o caso dos blocos mais pequenos - blocos de dimensão 4×4 , 4×8 e 8×4 .

É apresentado na figura 4.17 a imagem original e resultante da codificação, com a atribuição de bits referida: 6 bits para os blocos uniformes e 10 bits para os blocos de transição.



imagem original



imagem codificada

fig. 4.17 - Imagem original e imagem codificada com 6 bits para os blocos uniformes e 10 bits para os blocos de transição (8ª img da sequência "Miss America")

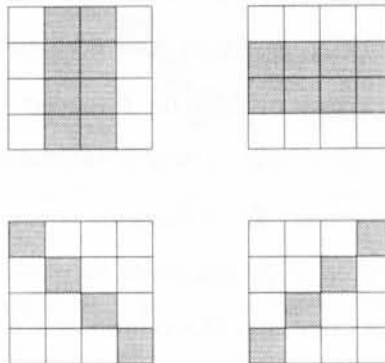
4.2.7.3 Resolução dos casos especiais

A utilização da imagem diferença como entrada no módulo de codificação, sendo vantajosa do ponto de vista da taxa de compressão, aumenta a sensibilidade do sistema aos erros propagados de imagem para imagem. Como no processo de codificação o valor dos elementos é substituído por uma representação, a diferença entre o valor real dos pixels e o da sua representação implicará um erro, erro esse que terá de ser compensado nas imagens seguintes.

De um modo geral, essas pequenas diferenças são atenuadas, pelo facto de o segmentador detectar os locais onde a desigualdade entre o valor da imagem real e o da imagem resultante da compressão é relevante. A detecção é conseguida porque esta desigualdade

resulta numa medida de actividade mais elevada e, devido ao algoritmo do segmentador, a essa zona é afectada um bloco de dimensão mais pequena. Na imagem seguinte, o bloco codificado tentará corrigir o erro que existia nas imagens precedentes, de tal forma que anule as alterações introduzidas pela deficiente representação dos elementos. O bloco detectado pelo segmentador, classificado agora com um tipo correspondente a um bloco activo e portanto de dimensões reduzidas, é transmitido ao codificador que o classifica em bloco uniforme ou de transição.

Se os erros de representação são tais, que o VPIC, no cálculo da magnitude do gradiente, considere que foi ultrapassado o limite de gradiente mínimo, o bloco é de transição e irá ser tratado como tal. Este tipo de bloco permite ao codificador corrigir os erros existentes devido à utilização da imagem diferença no processo de codificação. Poderão surgir problemas, no entanto, quando o bloco contém transições que não são detectáveis pela medida do gradiente, sendo classificado como bloco uniforme. Este tipo de bloco não conseguirá atenuar os erros em forma de contorno, permitindo apenas um ajuste da média do bloco.



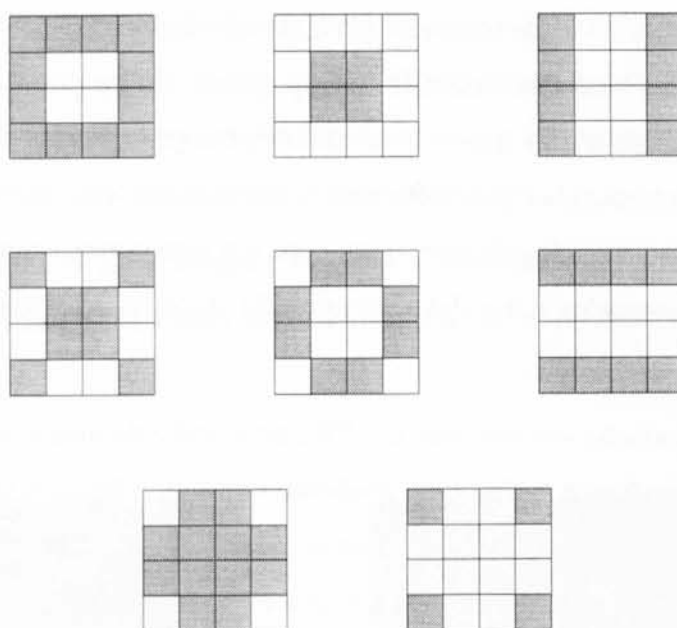


fig. 4.18 - Exemplos de blocos de erros não detectáveis

Na figura 4.18 está representado um conjunto de blocos que não são detectáveis pela medida do gradiente. Esta impossibilidade deve-se à particularidade de a diferença das médias na direcção horizontal, medida utilizada para o cálculo do gradiente parcial nessa direcção, bem como a diferença das médias na direcção vertical, correspondendo ao gradiente parcial vertical, terem valor nulo. Consequentemente, o valor da magnitude do gradiente, que é igual à hipotenusa do triângulo cujos catetos são os gradientes direccionais, é zero. Não sendo o valor do gradiente superior ao do limiar do gradiente, o bloco é, erradamente, considerado como um bloco uniforme e tratado como tal.

Uma possível solução para este problema de acumulação de erros, consiste na codificação de uma imagem original com o inevitável aumento do número de blocos e todas as desvantagens já referidas. Esta solução deve ser aplicada em intervalos de tempo regulares, ou quando a relação sinal-ruído atinge um valor considerado muito baixo. Resolve-se, desta forma, os erros cometidos em imagens anteriores. O decodificador, em sincronia

com o codificador, ou mesmo notificado por ele, descarta a imagem anterior, permitindo assim uma inicialização do sistema.

A transmissão de pixels com o valor verdadeiro é outra solução, se o débito binário de transmissão dos dados da imagem codificada ainda não atingiu o valor da capacidade do canal. Um processo que determine o débito de transmissão necessária para a transmissão da imagem codificada, permite estabelecer quantos destes pixels, cujo valor é o da imagem original, podem ser transmitidos. Este método está, contudo, dependente da taxa de compressão conseguida, a qual varia de imagem para imagem, não sendo garantido que em momentos de grande actividade possam ser transmitidos esses pixels.

A resolução adoptada permite, com o acréscimo de um reduzido número de operações, a detecção dos casos especiais mais graves, como sejam a ocorrência de transições nas linhas e colunas internas do bloco e nas diagonais principais.

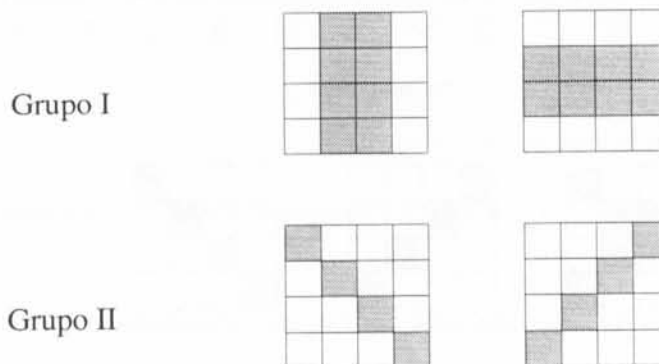


fig. 4.19 - Casos especiais contemplados

Na figura 4.19 estão representados os dois grupos contemplados para resolução. O primeiro corresponde a transições de linhas e colunas; no segundo grupo estão os casos em que as transições ocorrem nas diagonais principais. Estes casos não poderão ser resolvidos imediatamente pelo VPIC, pois os blocos contêm duas transições. Como foi referido no capítulo dedicado à introdução do VPIC, o dicionário de padrões apenas contempla os casos em que ocorra uma só transição.



Na figura 4.19, os blocos do primeiro grupo são facilmente classificados pela comparação da média das duas linhas centrais com a média global. Se, após a determinação do gradiente, o bloco é considerado uniforme, procede-se ao cálculo das médias acima referidas e, se a diferença dessas duas quantidades for superior a um valor pré-determinado, o bloco será tratado como um caso especial de bloco de transição. De igual forma são tratados os blocos do segundo grupo, caso em que o cálculo da média das diagonais é determinante para a qualificação de bloco de transição especial.

A escolha destes dois grupos para a resolução de casos especiais deve-se ao facto de que todos os outros casos podem ser resolvidos, de uma forma análoga, pela simples adição daqueles blocos com a polaridade apropriada. É, pois, possível tratar blocos nos quais ocorrem contornos e cuja magnitude do gradiente não permite a sua classificação como blocos de transição, pela codificação apropriada dos dois grupos especiais escolhidos. Naturalmente, devido ao facto de determinados blocos só poderem ser tratados com a adição de dois blocos especiais, a atenuação dos efeitos indesejáveis será conseguida nas imagens seguintes.

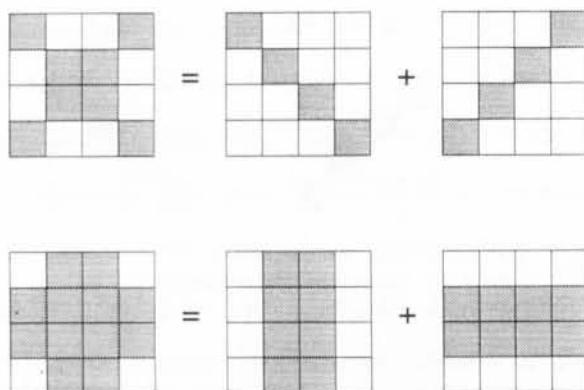


fig. 4.20 - Exemplo da resolução de outros blocos especiais, por meio da adição dos grupos escolhidos

A melhoria da relação sinal-ruído obtida pela introdução deste processamento paralelo à codificação é constatada no gráfico seguinte, alcançando-se uma melhoria de cerca de 5% relativamente à compressão onde os casos especiais não são contemplados. A degradação no tempo de processamento é desprezável uma vez que as operações a efectuar são

diminutas, tornando esta pequena alteração ao algoritmo um ponto a considerar em sistemas futuros.

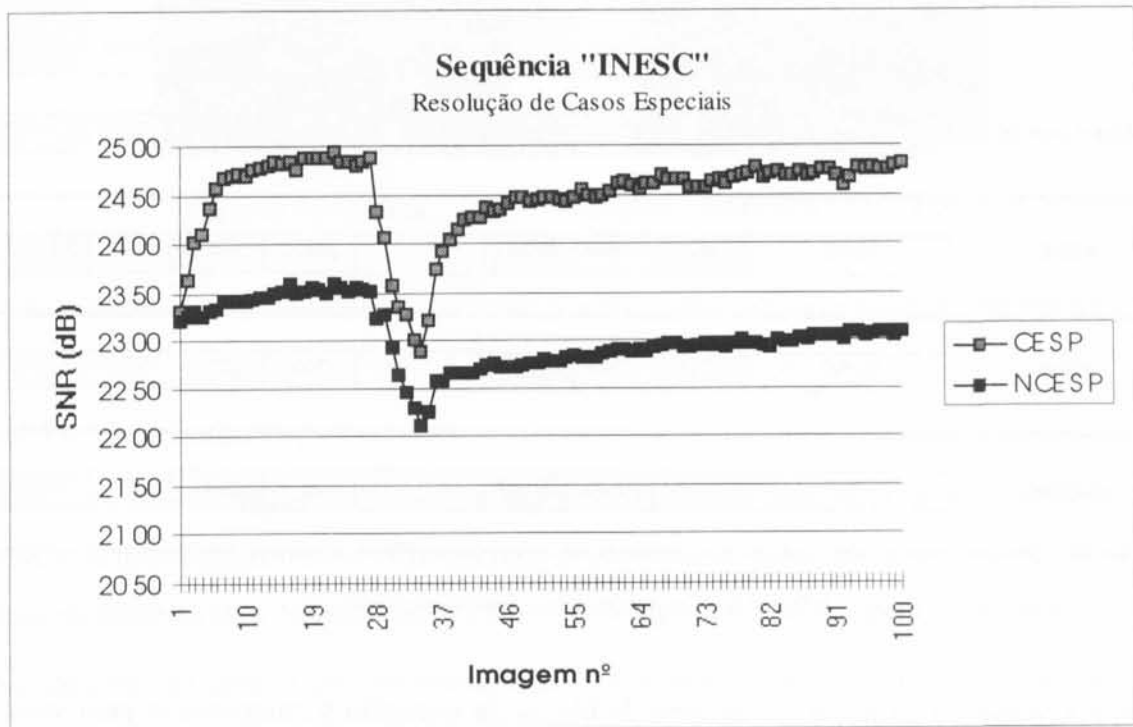


fig. 4.21 - Sequência "INESC" com resolução de Casos Especiais (CESP) e Sem Casos Especiais (NCESP)

Como é evidente, as qualidades de melhor adaptação às características da imagem só se fazem sentir quando os erros acumulados dos blocos de difícil resolução são tais que o processo de codificação dos casos especiais é utilizado, demarcando-se o sistema de forma notável do esquema de compressão clássico.

A desvantagem resultante do processamento dos casos especiais é o aumento do número de blocos que são classificados como de transição, diminuindo, por conseguinte, a taxa de compressão final. Foi realizado um estudo em relação a este óbice, contabilizando o número de blocos de cada tipo para cada uma das compressões. Os resultados são apresentados na tabela.

	CESP sem				com			
	Blocos nº unif	nº edge	Total	SNR (dB)	Blocos nº unif	nº edge	Total	SNR (dB)
Miss America	16442	4446	20888	31 699	12001	7569	19570	32 478
INESC	36132	5812	41944	23 361	34820	6938	41758	24 488
Alice	22495	1248	23743	30 519	13636	6293	19929	31 830
Claire	17542	3101	20643	29 846	12691	7221	19912	31 014

tabela 4.7 - Casos especiais vs. número de blocos

Como é visível, o aumento do número de blocos de transição é compensado pelo valor mais elevado da relação sinal-ruído e pela diminuição do número total de blocos a codificar. A redução do número de blocos a codificar deve-se a que, no processo clássico, as regiões correspondentes a blocos que o codificador não consegue tratar convenientemente são, na imagem seguinte, segmentadas em blocos pequenos. Se forem contemplados os casos especiais, esses blocos são codificados de forma eficiente, contribuindo para que o segmentador classifique as áreas respectivas como blocos de maiores dimensões.

Na figura 4.22 apresentam-se duas imagens, uma referente à imagem codificada pelo processo clássico e outra utilizando a resolução dos casos especiais.



imagem codificada



imagem codificada com CESP

fig. 4.22 - Imagem codificada sem e com a resolução dos casos especiais (10ª img da sequência "Miss America")

4.2.7.4 Conclusões

Foram estudadas, do ponto de vista de implementação e de resultados obtidos, as diversas variáveis que o algoritmo de VPIC contém. A alteração do valor destas variáveis permite a determinação da taxa de compressão bem como da qualidade da imagem codificada.

No decorrer da análise dos parâmetros do codificador, ficou claro que a utilização da imagem diferença possibilita uma maior taxa de compressão. Esta melhoria da eficiência de compressão deve-se à diminuição do número de blocos a codificar. A relação entre o limiar da magnitude do gradiente e o número de cada tipo de blocos, bem como da relação sinal-ruído, foi o assunto seguinte deste estudo. Concluiu-se a necessidade de uma escolha criteriosa dos limites a impôr, dadas as consequências que deles advêm. Adicionalmente, foi estudado o impacto na relação sinal-ruído do número de bits utilizados para a codificação da média e da magnitude do gradiente, chegando-se a uma conclusão para o melhor valor destas variáveis. A resolução de casos difíceis foi o último caso considerado. Nessa secção foram clarificados os problemas decorrentes da codificação de imagens diferença, obtendo-se resultados que poderão ser utilizados em sistemas de compressão que utilizem VPIC - usando imagens diferença ou imagens originais - , com o intuito de aumentar a qualidade da imagem codificada.

Recentemente, foram descritos novos desenvolvimentos deste algoritmo de codificação em [SBC93], com uma orientação mais específica para codificação de sequências de imagens.

Neste artigo são apresentados codificadores do tipo hierarquizado, como o HVPISC (*Hierarchical Visual Picture Image Sequence Coding*), um algoritmo de codificação em pirâmide AVPISC (*Adaptive VPISC*) e um método onde se explora ainda mais as características do sistema visual humano, o FVPISC (*Foveal VPISC*). Mais recentes são os trabalhos de Chan e Chan [ChC94], nos quais os princípios da geometria fractal são introduzidos no esquema de codificação, de modo a aumentar ainda mais a taxa de compressão. Embora estes algoritmos permitam obter resultados interessantes, a sua implementação requer um aumento considerável de complexidade e da memória de processamento, o que fez com que não tenham sido considerados, uma vez que era pretendido um sistema versátil e de baixa complexidade.

4.3 Compensação de movimento

4.3.1 Introdução

O interesse da utilização da compensação de movimento foi referido na secção 2.3.4 [*Compensação de Movimento*], bem como os métodos mais usados nos sistemas de compressão de vídeo em tempo real. A vantagem da inclusão dessa técnica de compressão, consiste na diminuição da redundância no plano temporal, existente nas sequências de imagens. Esta diminuição de redundância provocará uma diminuição dos valores de actividade local, o que levará o segmentador a atribuir blocos de maiores dimensões a estas zonas da imagem.

O algoritmo base de compressão que tem sido o ponto de partida, beneficiará com a introdução de um módulo de compensação de movimento, uma vez que a segmentação gerará um maior número de blocos de maiores dimensões, contribuindo assim positivamente para o aumento da taxa de compressão. O envio dos vectores de deslocamento, no caso da técnica de emparelhamento de blocos, deverá ser levado em conta de modo a que, um aumento da taxa de compressão num ponto não seja apenas

transitório. Nesta técnica, a apreciação dos ganhos devido à diminuição do número de blocos segmentados, deverá considerar a necessária transmissão de um número suplementar de bits relativo aos vectores de deslocamento. Na técnica baseada em algoritmos recursivos, o aumento da complexidade de implementação e do tempo de processamento será uma desvantagem a ter em conta.

Este módulo poderá ser introduzido em diversos pontos como, por exemplo, imediatamente antes da segmentação da imagem (ver fig 4.23). Deste modo tenta-se que a imagem diferença à entrada do segmentador tenha baixos índices de actividade, resultando num aumento do número de blocos de grande dimensões.

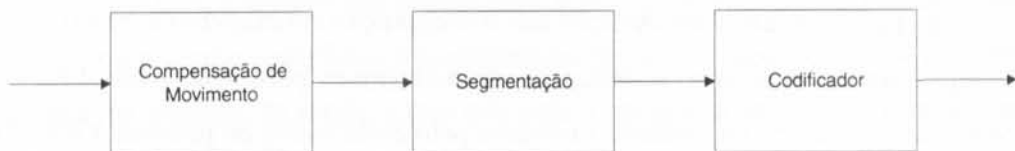


fig. 4.23 - Compensação de movimento antecedendo a segmentação

Outra implementação possível consiste na realização da operação de segmentação conjuntamente com a compensação de movimento (ver fig. 4.24). Desta forma, o segmentador, tentando atribuir o maior bloco possível a uma área sem que o limiar de actividade local seja ultrapassado, poderá ainda contar com a compensação de movimento para reduzir a actividade da região, contribuindo, assim, para a atribuição a essa área de um bloco de maiores dimensões.



fig. 4.24 - Segmentação e compensação de movimento realizadas conjuntamente

Uma terceira opção para o posicionamento de um módulo de compensação de movimento, consiste na sua introdução após a segmentação e imediatamente antes do codificador (ver fig 4.25). Esta opção tem em consideração não a diminuição do número de blocos de cada imagem segmentada, mas sim a tentativa de contribuir para que cada bloco seja classificado como sendo de intensidade uniforme pelo codificador de padrões VPIC. Desta forma, resultará uma melhoria da taxa de compressão, pois o número de bits necessário para a codificação de blocos uniformes é inferior ao que é requerido pelos blocos do tipo transição.

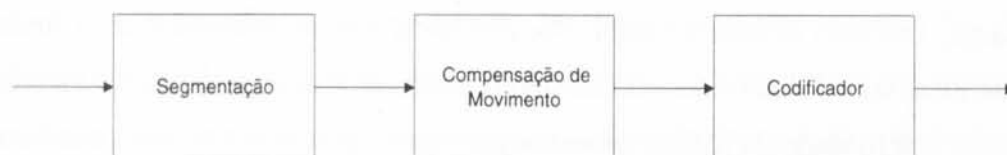


fig. 4.25 - Compensação de movimento precedendo o bloco de codificação

Foram executados ensaios e estimativas das três alternativas expostas com o objectivo de determinar qual a mais vantajosa. A última opção, a compensação com vista à diminuição de blocos contendo transições, embora seja válida teoricamente, revelou-se ineficaz, pois a diferença existente entre o número de bits exigido pela codificação dos dois tipos de blocos é diminuta para efeitos da taxa de compressão final. Era indispensável que o número de blocos de transição fosse reduzido de tal forma que, pelo menos, equilibrasse o acréscimo de tempo de processamento exigido pela compensação de movimento. Esta,

visto o ganho em bits de codificação ser precário, deveria ser do tipo recursivo, dificultando ainda mais o saldo obtido da taxa de compressão e tempo de processamento. As desvantagens foram de tal maneira preponderantes, que foi eliminada a hipótese de esta via de compensação de movimento ser a escolha acertada para os objectivos em consideração.

Restaram duas opções para a implementação do módulo de compensação de movimento, a compensação antes da segmentação ou conjuntamente com a segmentação. O intuito de ambas consiste na redução do número de blocos constituintes de cada imagem através da atribuição de blocos de maiores dimensões. Esta finalidade é atingida fazendo baixar o valor da actividade local por meio da operação de compensação de movimento. Ambas as formas de atingir esse objectivo são válidas mas, como se pretende neste trabalho obter um sistema flexível e pouco complexo, é aconselhável garantir a separação total dos módulos constituintes do sistema, de modo a que não sejam agregadas técnicas diferentes num só bloco. Resta assim a hipótese de colocar o compensador de movimento antes da segmentação.

As duas técnicas de compensação de movimento mais utilizadas, o emparelhamento de blocos e os algoritmos recursivos, oferecem uma compensação de movimentos de translação dos objectos. Devido ao maior grau de complexidade dos algoritmos do último tipo, um aumento do tempo de processamento é inevitável, facto que foi preponderante na escolha do método utilizado. Assim, foi procurado o método que, não sobrecarregando em demasia o sistema de compressão, contribuísse para um maior desempenho global da codificação. Deste ponto de vista, a técnica de emparelhamento de blocos é a que melhor satisfaz os requisitos, sendo esta a eleita para cumprimento dos objectivos propostos.

Para a técnica de emparelhamento de blocos é necessário seleccionar a dimensão dos blocos a compensar, a dimensão da janela de busca e o algoritmo de procura. Quanto à dimensão dos blocos, os blocos de menores dimensões são naturalmente mais indicados para pequenos objectos que sofram de deslocação. Em contrapartida, se a imagem a compensar for segmentada em pequenos blocos, um maior número deles terá de ser compensado, gerando por isso um elevado número de bits suplementares para a codificação dos vectores de deslocamento. A utilização de blocos de pequenas dimensões

na compensação de movimento poderá, deste modo, não resultar numa redução da taxa de compressão global. Por outro lado, blocos excessivamente grandes poderão incluir elementos da imagem cujos vectores de deslocamento não são idênticos, tendo como consequência uma má representação da translação real.

A dimensão da janela da imagem anterior, na qual é procurado o bloco a compensar, é igualmente importante, uma vez que se for exageradamente grande implicará um acréscimo do tempo de processamento que não é de todo desejável. Os vectores de deslocamento encontrados numa janela deste tipo podem, no entanto, representar com uma maior veracidade translações de elevado valor, sendo aconselhado o seu uso para objectos com uma velocidade de deslocamento elevada. De salientar que a dimensão da janela influencia de sobremaneira o número de bits suplementares a serem enviados, uma vez que o vector de deslocamento terá de cobrir todas as possíveis posições do bloco compensado na área de busca. Se se optar por uma janela de menores dimensões, dever-se-á levar em linha de conta que esta determinará a translação máxima permitida e o número de bits imposto para a codificação dos vectores de deslocamento.

Estando seleccionadas as dimensões dos blocos a compensar e da área de busca, resta escolher um algoritmo que execute a procura e calcule os vectores de deslocamento. Diversos métodos foram apresentados na secção 2.3.4.1. [*Técnica de emparelhamento de blocos*], desde a busca exaustiva que garante a determinação do vector óptimo mas é computacionalmente pesada, até algoritmos que visam a redução do número de procuras no interior da janela e a diminuição da complexidade de processamento, mas com a desvantagem de, por vezes, convergirem para mínimos locais. A escolha do algoritmo de procura pode determinar a exequibilidade de um módulo de compensação de movimento, devido às limitações do hardware utilizado para aplicações em tempo real.

4.3.2 Implementação e análise dos resultados obtidos

A realização da compensação de movimento, com vista à redução do número de blocos codificados e consequente diminuição de bits exigidos para a transmissão, fica completamente determinada com a atribuição de valores à dimensão dos blocos, à janela

de procura e com a escolha do algoritmo de busca.

Foi com base na norma H.261 do CCITT, que foram utilizados blocos cujas dimensões são 16×16 pixels, ficando ligeiramente acima da média da dimensão dos blocos utilizados na segmentação. Sendo 176×144 a área da imagem a codificar, pode determinar-se o número de blocos para os quais se efectuará a compensação de movimento. Deste modo, o número de blocos existente na imagem é calculado por

$$\frac{176 \times 144}{16 \times 16} = 99$$

Os blocos de dimensão 16×16 permitem um bom compromisso entre o número de blocos a compensar, a quantidade de vectores de deslocamento existentes, e a acuidade do vector de deslocamento representativo da translação real. As imagens são segmentadas em 99 blocos de 16 por 16 elementos, sendo cada um compensado em movimento, gerando 99 vectores de deslocamento.

Para a área de procura foi estabelecida uma dimensão que, não sendo exageradamente grande, permita uma correcta determinação do vector de deslocamento. Concluiu-se que uma janela de dimensões $W=\pm 8$ é suficiente para a maior parte das translações ocorridas, contribuindo para uma taxa de compressão elevada, pois o número de bits necessários para a codificação dos vectores de deslocamento é tanto menor quanto mais reduzida for a área de procura. Deste modo, 8 bits é o máximo necessário para a transmissão dos vectores sem que exista qualquer compressão destes dados.

Um dos pontos de principal interesse neste tipo de compressão, reside no algoritmo de procura do bloco de emparelhamento na janela de busca. O objectivo é um método que, elegendo o melhor ou uma boa aproximação do melhor vector de deslocamento real, seja de baixa complexidade de implementação, permitindo assim um rápido processamento. O método de busca exaustiva, como já foi referido, não é aceitável pois impõe um longo tempo de processamento. Outros métodos, sumariados na tabela 4.8, tentam reduzir o tempo necessário de processamento com a redução dos pontos de pesquisa no interior da janela de busca. Quanto menor for o número de pontos, menor é o tempo de

processamento. A redução desses pontos não deve ser tão drástica que não permita uma boa escolha do vector de deslocamento.

Algoritmo	Máximo número de pontos pesquisados	W		
		4	8	16
Busca Exaustiva	$(2W+1)^2$	81	289	1089
Logaritmico bi-dimensional	$2+7 \log_2 W$	16	23	30
Três passos	$1+8 \log_2 W$	17	25	33
Estimação de movimento modif.	$1+6 \log_2 W$	13	19	25
Direcções conjugadas	$3+2W$	11	19	35
Procura Ortogonal	$1+4 \log_2 W$	9	13	17
Procura em cruz	$5+4 \log_2 W$	13	17	21

tabela 4.8 - Análise de complexidade de diversos algoritmos de procura

Da análise da tabela anterior é perfeitamente nítido o interesse da escolha de tais algoritmos, podendo constatar-se as vantagens destes em relação ao método de busca exaustiva, para várias dimensões da janela de procura. O método adoptado foi o da procura em cruz (“*Cross-Search Algorithm*” ou *CSA*). Embora não sendo o que apresenta a menor complexidade de implementação, o *CSA* é o melhor do ponto de vista de desempenho, uma vez que o algoritmo de procura ortogonal não contempla a fronteira da janela de busca [Gha90].

O algoritmo de procura em cruz utiliza como medida de distorção, a média do erro absoluto

$$mae(i, j) = \frac{1}{N} \sum_{m=1}^N \sum_{n=1}^N |f(m, n) - f(m+i, n+j)|$$

$$-W \leq i, j \leq W$$

em que o mínimo valor da função corresponde ao melhor emparelhamento de blocos. O algoritmo de procura é descrito da seguinte forma:

1. O bloco a compensar é comparado com o bloco na imagem anterior cujas coordenadas são $(0,0)$ e, se a função de distorção é inferior a um limiar, então esse bloco é considerado como estacionário e a procura termina.
2. Inicializar a posição de mínimo (m,n) com $m=0$ e $n=0$ e atribuir o passo de busca com metade da amplitude máxima de deslocamento $p=W/2$.
3. As coordenadas do bloco a comparar são agora iguais ao mínimo resultante, ou seja, $(m,n)=(i,j)$.
4. Encontrar as coordenadas de mínimo (m,n) , dos ponto (i,j) , $(i-p,j-p)$, $(i-p,j+p)$, $(i+p,j-p)$, $(i+p,j+p)$.
5. Se o passo $p=1$, ir para 6. Se não, fazer $p=p/2$ e ir para 3.
6. Se a posição de mínimo (m,n) é (i,j) , $(i-1,j-1)$, ou $(i+1,j+1)$, ir para 7. De outro modo ir para 8.
7. Procurar a posição de mínimo em (m,n) , $(m-1,n)$, $(m,n-1)$, $(m+1,n)$, $(m,n+1)$, ou seja, a cruz Grega (+). Terminar a procura.
8. Procurar a posição de mínimo em (m,n) , $(m-1,n-1)$, $(m-1,n+1)$, $(m+1,n-1)$, $(m+1,n+1)$, ou seja, a cruz de S.André (x). Terminar a procura.

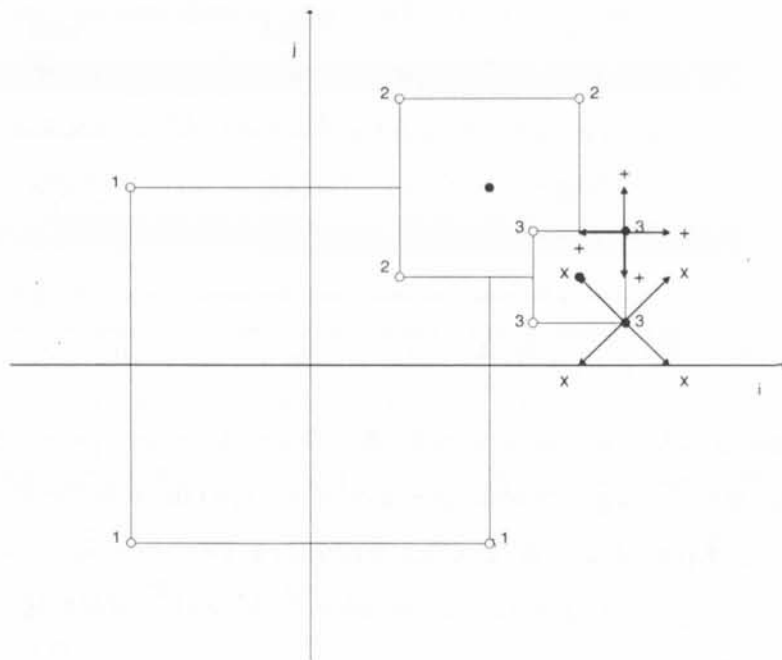


fig. 4.26 - Exemplo do algoritmo CSA para $W=8$

Este algoritmo foi implementado e foram realizados testes a 32 imagens de várias sequências, obtendo-se como resultados o número de blocos das diversas dimensões gerados pelo segmentador, bem como o número de blocos uniformes e de transição que resultaram da classificação de VPIC, após a compensação de movimento (ver tabela 4.9). Os ganhos evidentes em termos de taxa de compressão e de qualidade da imagem final e a enorme redução do número de blocos, permitem afirmar que este sistema de compressão, ao qual foi adicionado um módulo de compensação de movimento, é claramente superior ao sistema em que apenas existia o módulo de codificação VPIC.

<i>Sequência</i>	<i>Compensação Movimento</i>	<i>Blocos Uniformes</i>	<i>Blocos Transição</i>	<i>Total</i>
Miss America	sem	16442	4446	20888
	com	10187	3402	13589
INESC	sem	36132	5812	41944
	com	28290	5636	33926
Alice	sem	22495	1248	23743
	com	5073	653	5726
Claire	sem	17542	3101	20643
	com	9481	2228	11709

tabela 4.9 - Resultados comparativos entre a utilização ou não utilização da compensação de movimento

Pode facilmente concluir-se que a redução do número total de blocos só pode significar que as imagens que são segmentadas após a compensação de movimento são constituídas na sua maior parte por blocos de grandes dimensões. Isto pode ser observado na tabela 4.10, onde se compara a quantidade de blocos de cada dimensão sem ou com a compensação de movimento.

Sequência	C.M.	Tipo	0	1	2	3	4	5	6	7	8	9	10
Miss America	sem		7112	7168	897	3703	512	199	864	89	103	109	132
	com		3640	4118	595	3172	383	159	934	72	115	168	233
INESC	sem		25584	11896	967	1491	519	125	328	31	1	2	0
	com		15852	8864	1978	5382	105	181	553	53	0	10	0
Alice	sem		9887	7073	982	3698	720	366	730	33	61	66	127
	com		1337	881	202	1745	140	93	527	71	115	160	455
Claire	sem		7380	7292	595	3537	472	224	679	61	94	160	149
	com		3000	3076	473	3143	323	143	932	66	105	175	273

tabela 4.10 - Número de blocos de cada dimensão obtidos com e sem compensação de movimento

O êxito da compensação de movimento depende do saldo entre a redução do número de blocos e o número de bits exigido para transmissão adicional dos vectores de deslocamento. Como foi considerada uma janela com a dimensão de $W=\pm 8$, necessita-se de no máximo 8 bits para a codificação do deslocamento ocorrido. Devido à partição da imagem num número fixo de blocos, existe uma quantidade igualmente fixa de número de vectores cuja codificação tem de ser enviada. Assim, efectuando a compensação sobre 99 blocos, é necessário transmitir 792 bits/imagem para codificar os vectores de deslocamento. Pode, então, efectuar-se os cálculos necessários para determinação dos ganhos obtidos, tendo em conta a codificação efectuada pelo módulo de VPIC, o qual utiliza 6 bits para a codificação dos blocos uniformes e 10 para os blocos de transição.

Na tabela 4.11 apresentam-se os resultados finais obtidos. São descritos o número de blocos uniformes e de transição, assim como o número de bits por imagem, para os casos da utilização ou não utilização da compensação de movimento. O ganho obtido pelo uso da compensação de movimento varia entre os 8.5% e os 57.7%, consoante a sequência de teste utilizada. Este ganho significa a diminuição do número de bits por imagem quando é utilizada a compensação de movimento, em relação a um sistema que não a utilize.

Sequência	Comp. Movim.	Blocos		Codificação (bits/imagem)			Total (bits/img)	Ganho (%)
		Uniform	Transição	Uniform	Transição	Comp. Movim.		
Miss	sem	16442	4446	3082.9	1389.4		4472.3	
America	com	10187	3402	1910.06	1063.125	792	3765.18	15.8%
INESC	sem	36132	5812	6774.75	1816.25		8591	
	com	28290	5636	5304.375	1761.25	792	7857.62	8.5%
Alice	sem	22495	1248	4217.8	390		4607.8	
	com	5073	653	951.18	204.06	792	1947.25	57.7%
Claire	sem	17542	3101	3289.125	969.06		4258.18	
	com	9481	2228	1777.68	696.25	792	3265.93	23.3%

tabela 4.11 - Resultados finais do módulo de compensação de movimento

Observando a tabela 4.11 pode-se chegar à conclusão que, embora a utilização da compensação de movimento necessite de transmissão suplementar de bits para a indicação dos vectores de deslocamento, este número de bits adicional é plenamente justificado pelo ganho devido à diminuição do número de blocos. A redução do número de blocos tem como consequência o aumento da taxa de compressão global do sistema, sugerindo a utilização da compensação de movimento como um método de aumentar a eficiência da compressão.

Na figura seguinte apresenta-se uma imagem compensada em movimento, utilizando o algoritmo de procura descrito.



fig. 4.27 - Imagem compensada em movimento (8ª imagem da sequência "Miss America")

4.4 Identificação de fundos

4.4.1 Introdução

A especificidade da codificação de sequências de imagens vídeo, situação na qual existe, frequentemente, um cenário e diversos objectos que o percorrem, por exemplo, em sequências de videovigilância, permitiu já a divisão da imagem em blocos segundo dois critérios: o da intensidade luminosa e o de movimento de translação. No caso da intensidade luminosa, foi feita a distinção entre blocos nos quais a intensidade é considerada constante e outros em que ocorrem transições bruscas; este foi o caso tratado na secção sobre o VPIC. A divisão da imagem em blocos nos quais o movimento é considerado, fazendo-se a distinção entre blocos com e sem deslocamento, foi explorada na secção de compensação de movimento.

Podendo a sequência de imagens ser interpretada como um cenário sobre o qual existem objectos que transitoriamente o preenchem, foi concebido um algoritmo que tirasse proveito dessa particularidade. Esse algoritmo teria como objectivo a classificação dos objectos em dois grupos distintos:

⇒ Objectos considerados como fundo

⇒ Objectos considerados como não fundo ou com movimento

No primeiro grupo de objectos considerados como fundo estão incluídos todo o conjunto de peças e corpos que, pelas suas características, não estejam animados de movimento. Pode, por exemplo, classificar-se uma sala com toda a sua mobília como fazendo parte de um fundo ou cenário e, portanto, pertencente ao primeiro grupo sobre o qual se desenrola uma sequência de acções. A vantagem de tal organização de objectos é evidente, pois os blocos da imagem que são considerados fundo não necessitam de actualização por parte do decodificador, sendo desse modo redundante a sua transmissão, o que contribui para um aumento da taxa de compressão.

O módulo de identificação de fundos terá por missão a classificação dos blocos nos dois grupos referidos, tendo que efectuar medidas de actividade sobre todos os blocos da imagem, memorizando quais são os que possuem as características que os colocam no grupo de blocos estacionários, ou seja, de fundo. Estes blocos deverão permanecer inalteráveis por um certo intervalo de tempo, para que sejam correctamente eleitos como fundo. Desse modo, é permitida a construção dinâmica de uma imagem de fundo, característica indispensável para que o sistema reconheça e se adapte às alterações do cenário, que irão decorrendo na evolução da sequência no plano temporal.

Após a identificação de um bloco de fundo, quer o codificador quer o sistema de decodificação, ambos em sincronia de processamento, consideram-no como bloco pertencente à imagem de fundo. Este bloco, incluído agora na imagem de fundo, possibilita que o codificador no futuro possa enviar um código de fundo, de modo a que o bloco fique completamente codificado, dado o decodificador ter já em sua posse uma representação sua. Isto mesmo se pode observar na figura 4.28, onde está representado a codificação e decodificação de um bloco de fundo.

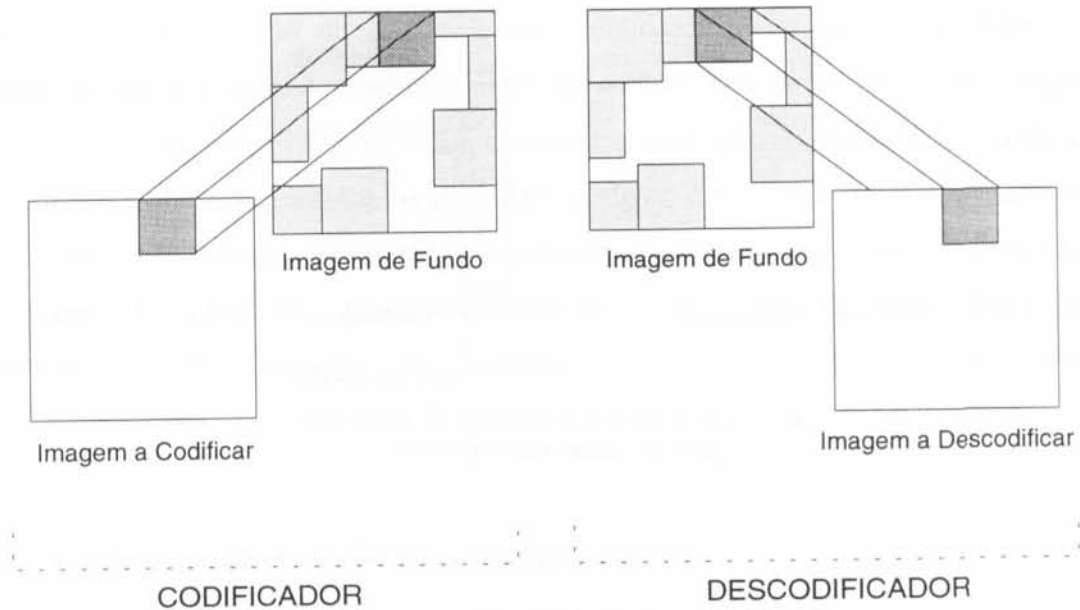
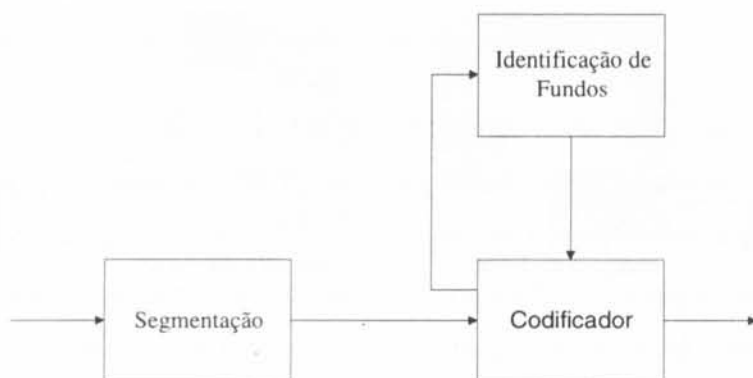
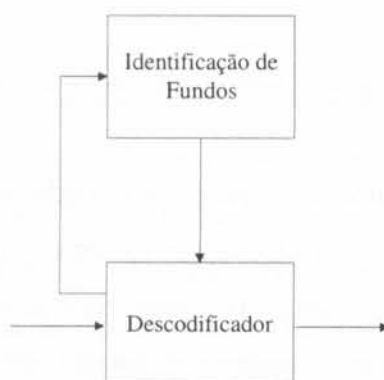


fig. 4.28 - Imagem de fundo no codificador e decodificador

No caso de um objecto ocupar uma zona que é classificada como fundo, o codificador não poderá enviar o código de fundo dos blocos que foram ocultos. Por esta razão, os blocos que contêm o objecto serão processados de forma usual pelo módulo de codificação. Se, no entanto, esse objecto permanecer inalterável durante um certo intervalo de tempo, o módulo de identificação de fundos procederá à sua inclusão na imagem de fundo. A alteração da imagem representativa de fundo não é mais do que o envio de um bloco classificado como pertencente ao fundo.

Os módulos constituintes do sistema de compressão e descompressão estão representados nas figura seguintes, onde é incluído um bloco de identificação de fundos.

*fig. 4.29 - Sistema de compressão**fig. 4.30 - Sistema de descodificação*

Com a adição de processamento, é esperada não só uma diminuição do número de blocos que será processado, mas também uma melhor utilização dos conhecimentos adquiridos até um determinado momento sobre as características próprias da sequência. O reconhecimento de locais dentro da imagem que pertençam ao que se pode apelidar de cenário, permite ao sistema de compressão codificar um bloco com apenas um código de fundo. O receptor, em sincronia com o sistema de compressão, descodifica o bloco utilizando a sua imagem de fundo. É com isto em mente que esperamos obter uma melhoria significativa da taxa de compressão, pois o simples envio de um código permitirá a descodificação de um bloco que anteriormente teria de ser processado.

A utilização de um módulo de identificação de fundos permite que, após a passagem de um objecto pelo fundo, este será prontamente repostado pelo sistema. Esta reposição do fundo, deve-se ao facto de o codificador reconhecer imediatamente os blocos que constituem a sua imagem de fundo. Deste modo, o codificador poderá enviar os códigos de fundo dos blocos para o receptor, e este efectuará a restituição do fundo. Desta forma tenta resolver-se o *problema dos fundos descobertos*, típico dos codificadores convencionais e que se deve à codificação deficiente do cenário que vai sendo descoberto pelo objecto em movimento. Este problema gera normalmente um certo “rasto” do objecto nas imagens subsequentes. A resolução do problema dos fundos descobertos contribui significativamente para o aumento da qualidade subjectiva da imagem reconstruída.

4.4.2 Implementação e análise dos resultados obtidos

A elaboração de um sistema que identifique zonas da imagem que são estacionárias e que, por isso, poderão ser classificadas como fundo, e detecte objectos que sobre ele se desloquem, coloca um primeiro problema a resolver que consiste em eleger uma medida tal, que permita a distinção entre esses dois grupos. Uma medida de semelhança entre o bloco a processar e o bloco correspondente na imagem de fundo, permite efectuar a discriminação entre blocos estacionários e blocos onde ocorre movimento. Esta medida de semelhança foi implementada utilizando a média do erro absoluto, medida esta descrita em capítulos anteriores. É, no entanto, necessário um complemento a essa medida de semelhança, pois uma zona só deverá ser classificada como fundo se permanecer imóvel durante um certo intervalo de tempo. Só após esse intervalo é que o bloco em consideração poderá integrar a construção ou actualização da imagem de fundo. A utilização de contadores que quantifiquem o número de imagens que uma dada zona permaneceu inalterável permite, a resolução deste problema. Cada contador efectuará o controlo de cada bloco de dimensão 4x4 da imagem a processar. Deste modo, é possível a construção do fundo com blocos de qualquer dimensão de entre os onze tipos considerados, pois estes podem ser reduzidos à agregação dos blocos de quatro por quatro.

A conjunção das condições de imobilidade e de permanência nesse estado por um tempo determinado, permitem classificar um dado bloco na categoria de bloco de fundo. Se o bloco integrar a imagem de fundo, essa zona da imagem poderá, a partir desse momento, ser plenamente identificada e utilizada para codificações futuras. Na figura seguinte está representada a imagem de fundo e a implementação dos contadores, constituintes do módulo de identificação de fundos.

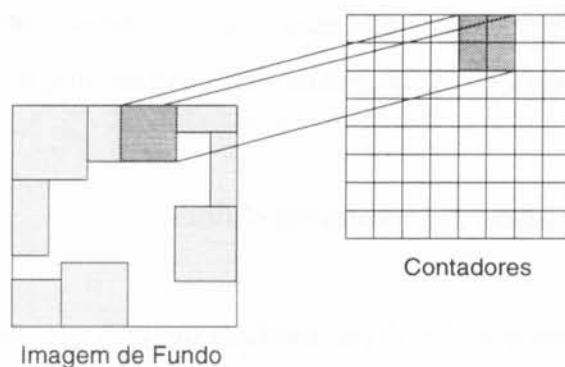


fig. 4.31 - Imagem de fundo e contadores

Existem, pois, duas vias para cada bloco da imagem: ou pertence à imagem de fundo e, por esta razão, é codificado como fundo, ou não faz parte da imagem de fundo e é processado pelo módulo de codificação de modo usual (por quantificação vectorial ou VPIC). A opção é tomada com base na determinação da medida de semelhança. Se o módulo de identificação de fundos considerar que o bloco considerado difere do bloco da imagem de fundo na mesma posição, dá indicação ao codificador que este bloco não constitui parte do fundo considerado até ao momento, e deverá ser codificado de forma usual. Se, por outro lado, a medida de semelhança determinar que o bloco a processar já existe na imagem de fundo, o módulo de identificação de fundos reconhece-o, e a codificação é efectuada através do envio do código de fundo.

A inexistência do bloco a processar na imagem de fundo, indica três hipóteses possíveis: ou se está perante o caso de um objecto móvel sobre o fundo, ou não está definida a imagem de fundo nessa zona, ou o fundo foi alterado. Na primeira hipótese pretende-se que, logo após a passagem do objecto sobre o fundo, este seja repostado. Deste modo, a

imagem de fundo deve permanecer inalterável para utilização posterior. No caso de indefinição da imagem de fundo nessa zona, os contadores associados a esse bloco deverão ser incrementados e, para cada contador em que se ultrapasse o limiar, o sub-bloco de 4x4 correspondente deverá integrar a imagem de fundo. Na última hipótese, a alteração do fundo, deverá de igual modo incrementar os contadores e proceder à actualização da imagem de fundo se for caso disso.

Na figura 4.32, está representado o caso de um bloco a codificar, em que apenas parte dos seus sub-blocos de 4x4, foram integrados na imagem de fundo. Estes sub-blocos correspondem às posições onde os contadores ultrapassaram o limiar.

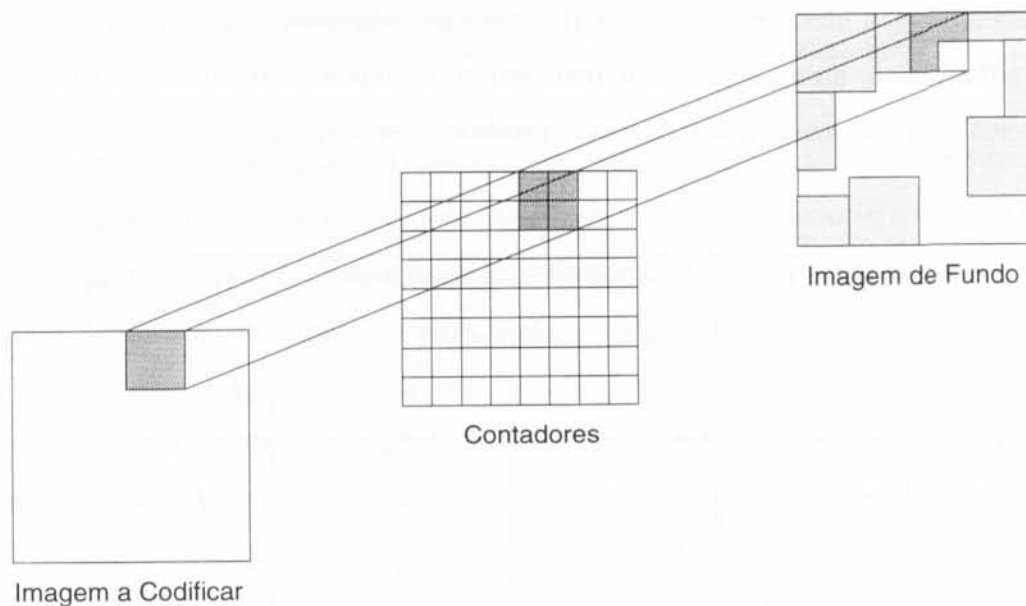


fig. 4.32 - Actualização dos contadores e da imagem de fundo

Após a actualização da imagem de fundo, os contadores que tenham ultrapassado o limiar serão inicializados. Deste modo, qualquer bloco a codificar que não pertença à imagem de fundo, fica desde já em posição de actualizar a imagem de fundo. O inicializar e incrementar os contadores permite a actualização de forma dinâmica da imagem de fundo, adaptando-se esta às alterações do cenário.

Existem dois valores de limites a atribuir: um que determina se o bloco pertence à imagem de fundo, e outro que define o valor máximo dos contadores.

Os testes realizados sobre 32 imagens de cada uma das sequências abaixo referidas incidiram sobre as variáveis de implementação já descritas, tendo sempre em atenção o objectivo de redução do número de bits necessário para descrever uma imagem codificada e o consequente aumento da taxa de compressão, proposto no início.

A tabela 4.12 dá uma indicação da influência no número de blocos do limiar de semelhança, limiar este que determina se o bloco a processar já existe na imagem de fundo, ou se é processado pelo codificador de forma usual. Um valor elevado do limiar permite uma maior utilização dos blocos da imagem de fundo, pois não é necessário que a semelhança entre o bloco a processar e o bloco da imagem de fundo seja elevada. Em contrapartida, se se atribuir um valor mais baixo, os blocos são sujeitos a uma maior semelhança com o bloco de fundo de modo a poderem ser utilizados.

<i>Sequência</i>	<i>nº Bloco Utilizados como fundo</i>		
	<i>Limiar de Semelhança</i>		
	<i>10</i>	<i>12</i>	<i>20</i>
Miss America	4253	4453	4669
INESC	6138	6837	7114
Alice	4879	5188	5439
Claire	3917	3954	3981

tabela 4.12 - Número de blocos utilizados como fundo em função do limiar de semelhança

A tabela 4.13 permite a comparação entre o número de blocos que são codificados como fundo (para o pior caso), e o número de blocos que são usados para actualização da imagem de fundo. Uma percentagem excessivamente grande de actualizações indica uma constante alteração da imagem de fundo. Em contrapartida, se o número de actualizações for diminuto, poderá significar que a imagem de fundo não se está a adaptar convenientemente às alterações do cenário. Foi observado, em sequências de imagens de

teste, que a relação obtida entre o número de actualizações e o número de blocos utilizados como fundo (entre os 11% e os 20%), permitiu uma construção estável e utilização eficiente da imagem de fundo.

<i>Sequência</i>	<i>nº Blocos</i>	
	<i>Utilizados</i>	<i>de Actualização</i>
Miss America	4253	512
INESC	6138	1241
Alice	4879	716
Claire	3917	456

tabela 4.13 - Número de blocos utilizados e de blocos de actualização da imagem de fundo

Um outro valor importante a ser considerado é o do limiar dos contadores. Esse limiar tem enorme influência no comportamento do algoritmo, pois determina o intervalo de tempo necessário para que um bloco, que permaneça inalterável, seja classificado como fundo. Um valor demasiadamente baixo poderá captar “falsos” fundos, como é o caso de um objecto animado de movimento lento. Sendo menor o tempo que é exigido para qualificação como fundo, existirá, consequentemente, um aumento de actualizações da imagem de fundo. Um limiar elevado, por outro lado, implica um igualmente elevado tempo de espera para que a imagem de fundo possa ser utilizada, bem como uma mais rígida condição de imobilidade por um determinado período.

<i>Sequência</i>	<i>n° blocos</i>	<i>Limiar Contadores</i>	
		<i>10</i>	<i>20</i>
Miss America	Actualiz	737	512
	Utiliza	8581	4453
INESC	Actualiz	1900	1241
	Utiliza	14291	6837
Alice	Actualiz	1014	716
	Utiliza	10582	5188
Claire	Actualiz	652	456
	Utiliza	8127	3954

tabela 4.14 - Limiar dos contadores vs. número de blocos de actualização e utilização

Em seguida, foi estudada a relação existente entre a dimensão e forma do bloco e o número de actualizações e utilizações efectuadas com este método. Os blocos foram divididos em dois grupos: um contendo os blocos de menores dimensões (os 4x4 até aos rectangulares de 4x8 e 8x4), e outro com os blocos de grandes proporções.

Quando apenas os blocos de maiores dimensões são candidatos a actualizar o fundo, devido ao menor número nas imagens segmentadas, ocorrerão poucas actualizações. Isto conduz, de modo geral, a uma menor área de representação de fundo, decrescendo a sua utilização. Em contrapartida, a actualização do fundo com blocos de dimensões mais pequenas, aumenta significativamente o número de alterações, mas não se consegue, como é natural, alcançar o número de utilizações efectuadas quando são considerados todos os blocos para actualização.

A tabela seguinte apresenta de forma numérica o que se acabou de afirmar; a linha denominada “Área” fornece uma indicação da percentagem da imagem de fundo que foi

construída. Deste modo, a uma pequena percentagem estão associadas vastas zonas da imagem onde não existe ainda fundo definido e, por essa razão, não poderão ser codificados como fundo blocos nessas posições.

<i>Sequência</i>		<i>Blocos de Actualização de dimensão</i>		
		<i>grande</i>	<i>pequena</i>	<i>todos</i>
Miss America	Actual	53	671	512
	Utiliz	1200	3854	4453
	Área	50 %	57.7 %	84.2 %
INESC	Actual	28	1330	1241
	Utiliz	734	6534	6837
	Área	15.1 %	94.2 %	95.6 %
Alice	Actual	50	758	716
	Utiliz	787	4415	5188
	Área	45.4 %	60.6 %	97.1 %
Claire	Actual	51	515	456
	Utiliz	920	3258	3954
	Área	45.3 %	56.5 %	84.5 %

tabela 4.15 - Relação entre as dimensões dos blocos de actualização e o número de actualizações e utilizações, bem como a área construída da imagem de fundo

O estudo referente ao tipo de blocos usados na actualização pode ser igualmente realizado para os blocos utilizados para repor o fundo. Os resultados obtidos não são surpreendentes pois, devido ao maior número de blocos de dimensões mais pequenas, é de esperar que estes sejam mais utilizados que os de maiores dimensões.

Sequência		Blocos de Utilização de dimensão		
		grande	pequena	todos
Miss America	Actual	510	510	512
	Utiliz	318	4027	4453
INESC	Actual	1241	1241	1241
	Utiliz	115	6586	6837
Alice	Actual	713	716	716
	Utiliz	393	4493	5188
Claire	Actual	455	456	456
	Utiliz	335	3492	3954

tabela 4.16 - Relação entre a dimensão dos blocos de utilização e o número de utilizações e actualizações

Depois de todas estas considerações acerca do controlo das diversas variáveis deste módulo de identificação de fundos, resta apenas a apresentação das considerações finais sobre os ganhos obtidos. É desejável a redução do número de bits que o codificador tem de transmitir ao receptor, elevando-se, desse modo, a taxa de compressão final. O número de blocos que utilizam a imagem de fundo determina o ganho obtido, pois só é necessário o envio de um código para que o bloco seja codificado.

A tabela seguinte resume, de forma explícita, os ganhos obtido pela técnica de identificação de fundos; neste sentido, são apresentadas as percentagens do uso de cada tipo de blocos.

<i>Sequência</i>	<i>Blocos (%)</i>		
	<i>Uniformes</i>	<i>Transição</i>	<i>Fundo</i>
Miss America	58.36	20.17	21.47
INESC	71.01	12.55	16.44
Alice	73.01	5.21	21.78
Claire	66.19	14.87	18.94

tabela 4.17 - Percentagem de cada tipo de bloco

A elevada percentagem de uso de blocos de fundo, sendo mesmo superior à dos blocos de transição, permite concluir que a utilização de uma técnica deste tipo é recomendável pois possibilita o aumento da taxa de compressão. De referir ainda que a qualidade das imagens reconstruídas, no que respeita à relação sinal-ruído, é da mesma ordem de grandeza da obtida sem este módulo de identificação de fundos; essa relação é apresentada na tabela 4.18.

<i>Sequência</i>	<i>Relação Sinal-Ruído</i>	
	<i>Identificação de Fundos</i>	
	<i>com</i>	<i>sem</i>
Miss America	31.63	31.70
INESC	23.35	23.36
Alice	30.42	30.52
Claire	29.8	29.85

tabela 4.18 - Resultados da relação sinal-ruído resultante do módulo de identificação de fundos

Embora as relações sinal-ruído sejam praticamente idênticas, a qualidade subjectiva da imagem reconstruída utilizando o módulo de identificação de fundos é significativamente superior. A melhoria da qualidade subjectiva, que se pode observar quando é codificada

uma sequência de imagens, justifica por si só a introdução do módulo de identificação de fundos.

Na figura seguinte apresenta-se a imagem original e a imagem de fundo obtida pelo módulo de identificação de fundos, bem como a imagem codificada resultante.



imagem original



imagem de fundo



imagem codificada

*fig. 4.33 - Imagem original, imagem de fundo e imagem codificada
(22ª imagem da sequência "Miss America")*

4.5 Compressão de sequências de imagens a cores

Embora neste trabalho a preocupação primordial tenha sido o aumento da taxa de compressão e a redução da complexidade do codificador, também se procurou melhorar a qualidade da imagem final. Nesta secção pretende-se avaliar a melhoria da qualidade subjectiva das imagens através da introdução da cor.

De facto, uma imagem a cores, devido à sua maior aproximação à realidade tende a ser considerada de maior qualidade pelos observadores. Por este facto, foi decidido tratar a cor como parte integrante da imagem, concebendo um sistema de codificação que a contemplasse.

As imagens adquiridas são de formato QCIF, sendo constituídas por 176x144 elementos de luminância e 88x72 pixels para a crominância de área de codificação

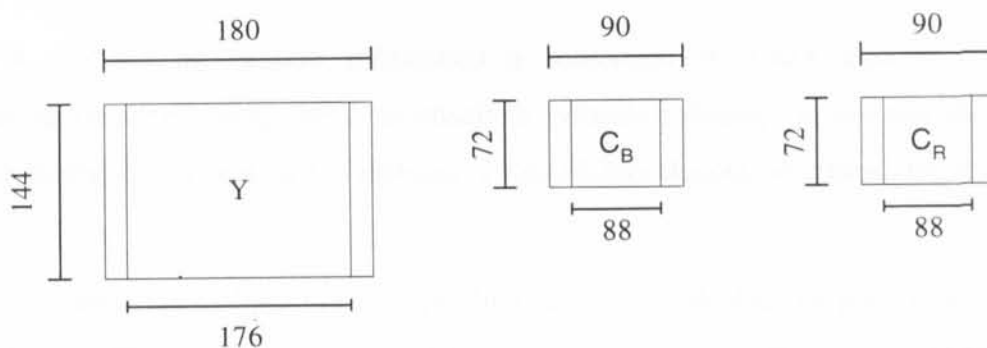


fig. 4.34 - A luminância e as duas componentes da crominância (formato QCIF)

O que se pretende é a codificação da luminância como a que se tem vindo a realizar e, adicionalmente, a codificação de cada componente da crominância. O sistema de codificação adoptado consiste na codificação da luminância e da crominância através de

VPIC. Cada componente da cromaticidade é segmentada e os seus blocos são processados pelo bloco de codificação de igual modo ao que é efectuado para a luminância.

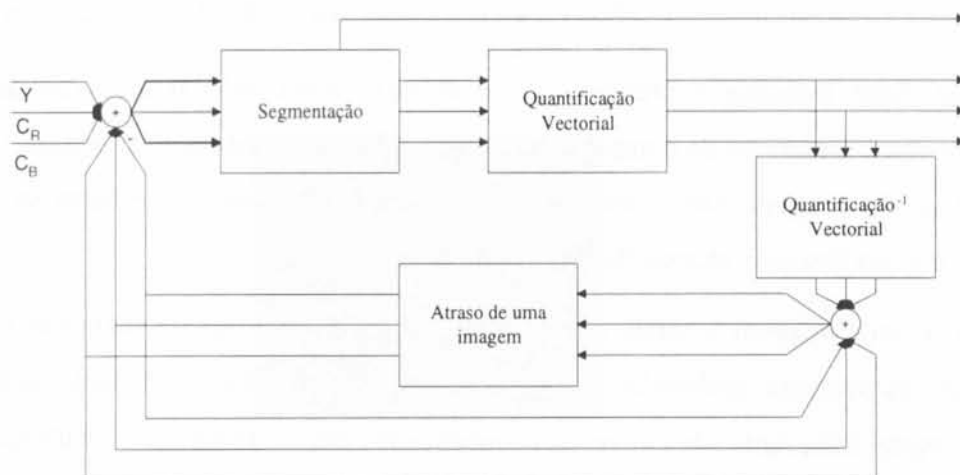


fig. 4.35 - Sistema de compressão da cor

Devido ao facto de cada componente da cromaticidade representar 1/4 da área da luminância, perfazendo no total 1/2, o tempo de processamento deverá ser proporcional a essa quantidade, sendo portanto esperado um aumento de cerca de 50%.

Podem aplicar-se aqui todos os resultados e conclusões obtidas no capítulo 4.2 [Codificação de imagem por padrões visuais], dedicado ao VPIC, pois a compressão da cor é executada pela aplicação daquele codificador à luminância e às duas componentes da cromaticidade.

Não aumentando a complexidade do sistema, a codificação da cor implica um aumento de forma significativa do número de bits que é necessário transmitir. Em contrapartida, assiste-se a uma melhoria evidente da qualidade da imagem reconstruída.

A redução do número de bits resultante da codificação da cor, pode ser obtida aplicando, de uma forma simples, a compensação de movimento às imagens. A componente da luminância é compensada em movimento, utilizando-se o vector de deslocamento resultante para compensar as duas componentes da cromaticidade. Desta forma, não é

necessário compensar em movimento a cromaticidade. Consegue, assim, reduzir-se o número de bits de uma imagem codificada, com um pequeno acréscimo de processamento.

Capítulo 5

Conclusões

5.1 Conclusões

Foram descritas, de forma sucinta, as técnicas de compressão de imagem mais utilizadas. Na apresentação destas técnicas, procurou reflectir-se sobre as vantagens e desvantagens de cada uma delas, tendo principalmente em atenção a complexidade de implementação, a qualidade da imagem e a taxa de compressão obtida.

O ponto de partida deste trabalho foi um sistema de compressão, em que é utilizada a quantificação vectorial de blocos previamente segmentados com dimensão e forma variáveis. O objectivo consistiu na diminuição da complexidade da implementação e no aumento da qualidade subjectiva das imagens, através da alteração ou introdução de

módulos alternativos de codificação que, ainda assim, conduzissem a elevadas taxas de compressão.

Com o intuito de substituir o módulo de quantificação vectorial (VQ), foi descrito um codificador baseado em padrões visuais (VPIC). Esta técnica, de mais baixa complexidade de implementação do que a introduzida pela quantificação vectorial, permitiu a obtenção de imagens de boa qualidade com taxas de compressão comparáveis às resultantes de um módulo de VQ.

A técnica de VPIC que foi apresentada, utilizando padrões que são subjectiva e visualmente significativos, permite que os blocos da imagem sejam codificados de forma eficiente tendo em consideração o sistema visual humano. Deste modo, são preteridas as medidas de distorção comuns, como a relação sinal-ruído, dando-se especial atenção à qualidade subjectiva da imagem codificada.

Este algoritmo de codificação, com um diminuto número de operações de adição e multiplicação reduz de forma significativa a complexidade de implementação. A procura de um padrão numa tabela extensa de códigos, operação realizada na quantificação vectorial, não é necessária, uma vez que o cálculo da tangente do ângulo do gradiente fornece informações precisas do sub-espço de codificação. A descodificação de VPIC é comparável à descodificação no caso da quantificação vectorial, sendo suficiente uma operação de adição e outra de multiplicação para que o bloco seja descodificado.

Foi executada uma rigorosa análise dos valores das variáveis que o algoritmo contém, tendo em conta a complexidade da implementação, a taxa de compressão e a qualidade da imagem resultante observada.

Para aumentar a taxa de compressão do sistema foi introduzido um módulo de compensação de movimento. Foram feitas algumas considerações que determinassem a melhor forma de a implementar. Das três opções apresentadas para a possível inclusão deste módulo de compensação: antes da segmentação, conjuntamente com a segmentação e imediatamente após a segmentação, foi adoptada a introdução da compensação de movimento antes da operação de segmentação, uma vez que permite a redução do índice de actividade da imagem e, como consequência, a partição da imagem em blocos de maiores dimensões. A utilização de métodos expeditos de busca, como o do algoritmo de

procura em cruz, permitiu a diminuição do tempo de processamento, de modo a que não se sobrecarregasse em demasia o sistema de compressão. A escolha da dimensão dos blocos a compensar, que segue a norma H.261 do CCITT com blocos de 16x16 elementos, e a determinação da área de procura estabeleceu o número de vectores de deslocamento bem como a quantidade de bits necessários à sua codificação. Com a dimensão dos blocos utilizada são necessários 99 vectores de deslocamento para a compensação de movimento de uma imagem. Exigindo a dimensão da janela 8 bits de codificação para cada vector, a compensação da imagem perfaz um total de 792 bits necessários. Este valor foi de longe superado pelos ganhos obtidos pela diminuição do número de blocos a codificar, resultando um saldo positivo devido ao uso da compensação de movimento, o que sugere a sua utilização como meio de alcançar maiores taxas de compressão. O objectivo proposto de melhoria da taxa de compressão foi atingido através da diminuição do número de blocos por imagem que o módulo de codificação tem que processar. De facto, o aumento do número de blocos de maiores dimensões em cada imagem permitiu a codificação com um menor número de bits por imagem, elevando deste modo a taxa de compressão.

Em seguida, foi desenvolvido um algoritmo com o objectivo de melhorar a qualidade subjectiva das imagens e, ao mesmo tempo, permitir um acréscimo da taxa de compressão. Este módulo, denominado de “identificação de fundos”, é complementar do módulo principal de codificação e explora o facto de vastas zonas da imagem poderem ser consideradas como estacionárias. A construção de uma imagem que representasse um cenário de fundo pareceu interessante, pois permitiria a codificação dos blocos com um reduzido número de bits e resolveria o problema dos fundos descobertos. Foi concebido um algoritmo que resolvesse o problema proposto, com um grau de complexidade mínima e, portanto, passível de implementação em tempo real. Os bons resultados obtidos revelam as vantagens da implementação de um módulo de identificação de fundos, à custa de um pequeno acréscimo da complexidade de implementação.

Finalmente, foi considerado o caso da codificação de sequências a cor. Foi apresentada a solução da compressão da luminância e da croma, separadamente, pelo módulo de VPIC. Embora implicando um acréscimo do tempo de processamento, a ausência de

aumento significativo da complexidade do sistema e a melhoria da qualidade subjectiva das imagens codificadas, fazem com que a codificação da cor deva ser considerada.

Foram apresentadas diversas técnicas alternativas para o sistema de compressão inicial, permitindo assim a concepção de sistemas adaptados às características das aplicações. Deste modo, é possível a substituição do módulo de quantificação vectorial pelo módulo de VPIC, com a intenção de reduzir a complexidade de implementação. Se, por outro lado, for desejado um aumento da taxa de compressão, a inclusão dos módulos de compensação de movimento ou de identificação de fundos são opções a considerar. Em sistemas em que a qualidade subjectiva é importante, a codificação da cor e a identificação de fundos devem ser consideradas.

5.2 Perspectivas de trabalho futuro

A implementação de um sistema que fizesse uso simultâneo das diversas técnicas descritas, permitiria o aumento da qualidade da imagem sem a consequente diminuição da taxa de compressão. Deste modo, poder-se-ia compensar a redução da taxa de compressão introduzida por uma técnica destinada à melhoria da qualidade das imagens, com o aumento desta taxa por parte de uma outra técnica cujo objectivo fosse a melhoria da eficiência da compressão.

A utilização de técnicas de pré e pós-processamento permitiria a resolução de alguns problemas de complexidade e da qualidade das imagens obtidas. O uso de códigos de comprimento variável, como os códigos aritméticos ou os de Huffman, à saída do codificador, possibilitaria a redução do número de bits por imagem codificada, elevando deste modo a taxa de compressão obtida. A resolução do problema do efeito de bloco, à custa de técnicas de filtragem pós-processamento, permitiria o aumento da qualidade subjectiva da imagem. As restrições impostas pela utilização de hardware com capacidade limitada, não possibilitaram a implementação das técnicas referidas, bem como das modernas técnicas de codificação baseada em objectos. Estas últimas técnicas, embora promissoras do ponto de vista da taxa de compressão, são de elevada complexidade de

implementação, impossibilitando o seu uso na área da compressão de imagens em tempo real utilizando hardware pouco potente.

Referências

- [AHS89] K. Aizawa, H. Harashima, T. Saito, "Model-based analysis synthesis image coding (MBASIC) system for a person's face", *Signal Processing: Image Communication*, vol.1, nº2, 1989
- [ALR78] A. Arking, R. C. Lo, A. Rosenfeld, "A Fourier approach to cloud motion estimation", *American Meteorological Soc., Journal of Applied Metereology*, nº17, 1978
- [Ber71] T. Berger, "Rate distortion theory", *Prentice Hall*, 1971
- [Bol90] J. L. Boxerman, H. J. Lee, "Variable block-sized vector quantization of grayscale images with unconstrained tiling", *Proc. of IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 1990

- [Buz80] A. Buzo, A.H. Gray Jr., R. M. Gray, J. D. Markel, "Speech coding based upon vector quantization", *IEEE Trans. on Acoustics Speech and Signal Processing*, ASSP-28, 1980

- [CGR84] D. Y. Cheng, A. Gersho, B. Ramamurthi, Y. Shoham, "Fast search algorithms for vector quantization and pattern matching", *Proc. International Conference on Acoustics Speech and Signal Processing*, vol.1, 1984

- [ChB90] D. Chen, A. C. Bovik, "Visual pattern image coding", *IEEE Trans. on Communications*, vol.38, nº12, 1990

- [ChC94] K. W. Chan, K. L. Chan, "Variable block and multi-level extended visual pattern image coding", *IEEE*, 1994

- [ChG85] Chang-Da Bei, R. M. Gray, "An improvement of the minimum distortion encoding algorithm for vector quantization", *IEEE Trans. on Communication*, vol. COM-33, 1985

- [ChP86] S. S. Chen, M. Penna, "Shape and motion of nonrigid bodies", *Computer Graphics, Vision and Image Proc.*, nº36, 1986

- [Cla85] R. J. Clarke, "Transform coding of images", *Academic Press*, 1985

- [CRA90] L. Corte-Real, A. P. Alves, "Vector quantization of image sequences using variable size and variable shape blocks", *Electronics Letters*, vol. 26, nº18, 1990

- [EnW81] R. M. Endlich, D. E. Wolf, "Automatic cloud tracking applied to GOES and METEOSAT observations", *J. Appl. Meteorology*, nº20, 1981

-
- [FoK89] R. Forchheimer, T. Kronander, "Image coding - from waveforms to animation", *IEEE Trans. Acoust. Speech Signal Process.*, vol.37, nº12, 1989
- [Gha90] M. Ghanbari, "The cross-search algorithm for motion estimation", *IEEE Trans. on Communications*, vol.38, nº7, 1990
- [GrA82] R. M. Gray, H. Abut, "Full search and tree searched vector quantization of speech waveforms", *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, vol.1, 1982
- [Gra84] R. M. Gray, "Vector quantization", *IEEE ASSP Magazine*, 1984
- [Gra90] R. M. Gray, "Source coding theory", *Kluwer Academic Publishers, Boston*, 1990
- [HaB80] U. L. Haass, T. A. Brubaker, "Estimation of cloud motion from satellite pictures", *Proc. of the IEEE Int. Conf. Acoustics, Speech and Signal Proc.*, ICASSP 80, 1980
- [Jai81] Anil K. Jain, "Image data compression: a review", *Proc. of the IEEE*, vol. 69, nº 3, 1981
- [JaJ81] J. R. Jain, A. K. Jain, "Displacement measurement and its application in interframe image coding", *IEEE Trans.*, COM-12, 1981
- [KaR83] S. Kappagantula, K. R. Rao, "Motion compensated predictive coding", *Proc. Int. Tech. Symp. SPIE*, 1983
-

- [Kog81] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, T. Ishiguro, "Motion compensated interframe coding for video conferencing", *Proc. Nat. Telecommun. Conf.*, 1981
- [LBG80] Y. Linde, A. Buzo, R. M. Gray, "An algorithm for vector quantizer design", *IEEE Trans. on Communications*, COM-28, 1980
- [Llo57] S. P. Lloyd, "Least squares quantization in PCM", *Bell Laboratories Technical Note*, 1957
- [Mar92] M. Isabel Martins, L. Corte-Real, N. Vasconcelos, A. Pimenta Alves, "Low bit rate vector quantization of image sequences with reduced complexity", *RECPAD92*, 1992
- [NeR79] A. Netravali, J. Robbins, "Motion compensated television coding - Part I", *The Bell System Technical Journal*, vol. 58, n°3, 1979
- [Net80] A. N. Netravali, "Picture coding: a review", *Proc. of the IEEE*, vol. 68, n°3, 1980
- [PHS87] A. Puri, H. M. Hang, D. L. Schilling, "An efficient block-matching algorithm for motion compensated coding", *Proc. IEEE ICASSP*, 1987
- [SBC93] P. L. Silsbee, A. C. Bovik, D. Chen, "Visual Pattern Image Sequence Coding", *IEEE Trans. on Circuits and Systems for Video Technology*, vol.3 n°4, 1993
- [Sha48] C.E. Shannon, "A mathematical theory of communication", *Bell System Tech. J.*, 27:379-423,623-656, 1948

-
- [SoM89] M. R. Soleymani, S. D. Morgera, "A fast encoding technique for vector quantization", *IEEE Trans. on Communication*, vol. COM-37, 1989
- [SrR84] R. Srinivasan, K. R. Rao, "Predictive coding based on efficient motion estimation", *IEEE Int. Conf. Commun.*, 1984
- [TeI88] Texas Instruments: "Third-Generation TMS320" User's Guide, *Digital Signal Processor Products*, 1988
- [TuH94] L. Torres, J. Huguet, "An improvement on codebook search for vector quantization", *IEEE Trans. on Communications*, vol.42, n° 2/3/4, 1994
- [VaG87] D. J. Vaisey, A. Gersho, "Variable block-size image coding", *Proc. of IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 1987
- [VaG92] D. J. Vaisey, A. Gersho, "Image compression with variable block size segmentation", *IEEE Trans. on Signal Processing*, vol. SP-40, 1992
- [VeB93] R. Veldhuis, M. Breeuwer, "Source coding", *Prentice Hall*, 1993
- [Veg89] J. F. Vega-Riveros, K. Jabbour, "Review of motion analysis techniques", *IEE Proceedings*, vol. 136, n°6, 1989
- [WHE77] D. E. Wolf, D. J. Hall, R. M. Endlich, "Experiments in automatic cloud tracking using SMS-GOES data", *J. Appl. Meteorology*, n°16, 1977
-

- [WSW90] W. J. Welsh, S. Searby, J. B. Waite, "Model-based image coding",
British Telecom Technol. J., vol.8, nº3, 1990





FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

BIBLIOTECA



0000004451

