



INSTITUTO
SUPERIOR
TÉCNICO

UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Compressão de Vídeo para Aplicações Submarinas

Jorge Augusto Castro Neves Barbosa
(Mestre)

Dissertação para obtenção do Grau de Doutor em
Engenharia Electrotécnica e de Computadores

Orientador: Doutor Victor Alberto Neves Barroso

Júri:

Presidente: Reitor da Universidade Técnica de Lisboa

Vogais: Doutor Armando José Formoso de Pinho, UA
Doutor Victor Alberto Neves Barroso, IST, UTL
Doutor Arnaldo Joaquim Castro Abrantes, ISEL, IPL
Doutor Mário Alexandre Teles de Figueiredo, IST, UTL
Doutor Pedro Manuel Quintas Aguiar, IST, UTL

Fevereiro de 2004

Resumo

A compressão de vídeo a baixas taxas é tratada neste trabalho. O objectivo principal é o de transmitir, em tempo real, sequências de vídeo, adquiridas em ambientes submarinos, através do canal acústico submarino. Este é um problema muito exigente dadas as fortes limitações impostas pelas propriedades físicas desse canal. Em particular, os sistemas acústicos submarinos de comunicação de dados, eficientes e confiáveis, só conseguem acomodar baixas taxas de transmissão. O objectivo é por conseguinte a construção de algoritmos de compressão de vídeo específicos, com transmissão em tempo real a taxas de dados limitadas superiormente entre os $20 - 30Kbps$.

Este problema foi motivado por uma aplicação prática: a detecção visual remota de actividade hidrotermal no fundo submarino. Os utilizadores finais são cientistas interessados nos fenómenos de natureza geológica e biológica relacionados com esta actividade, que podem ser detectados pela observação da emissão de bolhas de gás e pelo seu movimento ascendente na direcção da superfície do oceano. Estas emissões produzem imagens vídeo com propriedades muito específicas: baixo contraste, alta transparência e uma relativa alta velocidade ascensional das bolhas. Estas propriedades impedem uma utilização directa das técnicas *standard* para compressão de vídeo a baixas taxas.

Aqui, um novo método de compressão é proposto, desenvolvido e avaliado. Essencialmente, consiste em dois passos: **(i)** detecção das colunas de bolhas ascendentes e segmentação das imagens, isolando aqueles blocos onde está presente somente o movimento ascendente; e **(ii)** compressão deste tipo de blocos de imagem.

O primeiro passo é implementado usando-se filtros espaço/temporais de Gabor. Esta abordagem resulta num algoritmo de segmentação eficiente, de baixa complexidade e paralelo.

O segundo passo, compressão dos blocos de imagem seleccionados, é baseado em técnicas de quantização vectorial. A principal novidade é que é usado um novo

método de quantização vectorial que permite a compressão de uma sequência temporal de blocos de imagem usando apenas uma entrada do *codebook*, resultando num algoritmo de compressão *3D*.

As experiências realizadas, usando sequências de vídeo reais, mostram que os algoritmos desenvolvidos cumprem as restrições da taxa de dados especificadas. A qualidade das sequências de vídeo reconstruídas foram avaliadas quantitativamente usando como medida o PSNR. Como consequência das altas taxas de compressão obtidas pelos métodos propostos, os resultados obtidos mostram valores do PSNR aceitáveis. Também a qualidade visual qualitativa das imagens reconstruídas é perfeitamente compatível com os requisitos do utilizador final: identificar o *background*, visualizar as emissões de bolhas e poder estimar as suas dimensões e velocidade.

Palavras-chave: Compressão de Vídeo, Quantização Vectorial, Filtros de Gabor, Transmissão Acústica, Veículos Submarinos.

Abstract

Low bit rate video compression is considered in this work. The main objective is to transmit real time video sequences, acquired in underwater environments, through the underwater acoustic channel. This is a very demanding problem given the strong limitations imposed by the physical properties of that channel. In particular, efficient and reliable data communication underwater acoustic systems can only accommodate low transmission rates. The goal is therefore to design specific video compression algorithms compatible with real time transmission at data rates upper bounded in the range $20 - 30Kbps$.

This problem was motivated by a practical application: remote visual sensing of hydrothermal activity on the ocean bottom. The final users are scientists interested in geological and biological phenomena related with that activity, which can be detected by observing the emission of bubbles of gas and their ascending movement in the direction of the ocean surface. These emissions produce video images with very specific properties: low contrast, high transparency, and relatively high velocity of the ascending movement of the bubbles. These properties preclude direct use of standard low bit rate video compression techniques.

Here, a new compression method is proposed, developed, and evaluated. Essentially, it runs in two steps: **(i)** detection of the ascending bubble columns and segmentation of the images by isolating those blocks where strict ascending movement is present; and **(ii)** compression of this class of image blocks.

The first step is implemented using space/time Gabor filters. This approach results in a low complexity, parallel, and efficient segmentation algorithm.

The second step, compression of the selected image blocks, is based on vector quantization techniques. The main novelty is that we use a new vector quantization method that allows the compression of a time sequence of image blocks using just one entry of the codebook, resulting in what can be denoted as a 3D compression algorithm.

The experiments performed using real video sequences show that the algorithms developed cope with the specified data rate constraints. The quality of the reconstructed video sequences was quantitatively evaluated using the PSNR measure. As a consequence of the high compression rates given by the proposed methods, the results obtained show acceptable values of the PSNR. Also, the qualitative visual quality of the reconstructed images is perfectly compatible with the final user requirements: identifiable background, bubble emission visibility, and assessment to bubbles' dimensions and velocity.

Keywords: Video Compression, Vector Quantization, Gabor Filters, Acoustic Transmission, Underwater Vehicle.

Dedicatória

Para a Nanda,
para o Renato,
para o Miguel,

verdadeiras mãos invisíveis na elaboração desta tese.

Agradecimentos

Este trabalho não teria sido possível sem a colaboração e a amizade de muita gente. A todos os que contribuíram para a sua concretização o meu obrigado.

No entanto, não poderia deixar de referir em particular os que mais de perto me acompanharam e, principalmente, me estimularam ao longo de todo este tempo.

Aos meus colegas do Laboratório de Processamento de Sinal e Imagem e do Laboratório de Visão do ISR/IST, em especial ao João Xavier, João Pedro Gomes, Nuno Gracias, Etienne Grossmann, César Silva, João Maciel e José Gaspar, o meu obrigado e até sempre.

Aos meus colegas do DEIS, em especial ao Acácio Amaral, Amâncio Santos, Anabela Gomes, Carlos Pereira, Francisco Duarte, Jorge Bernardino, José Marinho, Tiago Figueira e Viriato Marques, agradeço a amizade. Ao Álvaro Santos agradeço a amizade e também as "dicas" na programação Windows.

Aos Profs. Dr^o. João Paulo Costeira e Mário Figueiredo pelo interesse e disponibilidade sempre manifestada para debater ideias. À Prof. Dr^a. Isabel Lourtie pela amizade.

Ao Prof. Dr^o. Victor Barroso por ter sentido que para além de meu orientador científico foi também um amigo.

Foi um prazer para mim ter estado todos estes anos a desenvolver investigação num dos mais prestigiados laboratórios de investigação portugueses: o Instituto de Sistemas e Robótica - Pólo do IST. Queria agradecer, na pessoa do seu director Prof. Dr^o. João Sentieiro, todas as facilidades concedidas. Pudera eu também ter contribuído um pouco para o prestígio do ISR/IST.

Ao ISEC e ao IPC pelo apoio concedido, respectivamente, nas pessoas dos seus presidentes, Mestre Eng^o João Benjamim e Prof. Dr^o. Torres Farinha.

Finalmente à minha família o meu obrigado. Não foram concerteza poucas as vezes em que, por causa deste trabalho, não correspondi como deveria. Aos meus sogros um especial obrigado.

Índice

Resumo	i
Abstract	iii
Dedicatória	v
Agradecimentos	vii
Lista de Figuras	xi
Lista de Tabelas	xv
1 Introdução	1
1.1 Compressão de Dados	1
1.2 Formulação do Problema	3
1.3 Compressão de Imagens Submarinas com Actividade Hidrotermal	5
1.4 Principais Contribuições	7
1.5 Organização da Dissertação	7
2 Compressão de Imagem	9
2.1 Introdução	9
2.2 A Teoria da Informação e a Compressão de Dados	10
2.3 Medição da Qualidade	13
2.4 Compressão de Imagens	15
2.4.1 Compressão de Imagens Sem Perda	15
2.4.2 Compressão de Imagens Com Perda	17
2.5 Compressão de Vídeo	32
2.5.1 Normas para a Compressão de Vídeo	33
2.5.2 Compressão para Transmissão de Vídeo a Muito Baixa Cadência	37
2.6 Sumário	41
3 Detecção de Movimento em Sequências de Imagens Submarinas	43
3.1 Introdução	43
3.2 Transmissão e Reconstrução das Imagens	44
3.3 Detecção dos Blocos de Imagens com Actividade Hidrotermal	46
3.3.1 Percepção do Movimento em Sequências de Imagens	46

3.3.2	Filtros Direccionais Sensíveis ao Movimento	51
3.3.3	Detecção de Movimento com Filtros de Gabor	53
3.3.4	Aplicação de Filtros de Gabor na Detecção do Movimento nas Imagens Submarinas	56
3.4	Sumário	61
4	Compressão de Vídeo Submarino	63
4.1	Introdução	63
4.2	Compressão de Sequências de Imagens com Quantização Vectorial Multibloco	64
4.3	Compressão de Imagens com Quantização Vectorial com <i>n-codebooks</i> .	69
4.3.1	Algoritmo para Compressão VQ2.5DN	70
4.3.2	Resumo do Algoritmo VQ2.5DN	75
4.3.3	Algoritmo para Compressão VQ3DN	77
4.3.4	Resumo do Algoritmo VQ3DN	80
4.3.5	Quantização Vectorial com <i>n-codebooks versus</i> Qualidade das Imagens Reconstruídas	82
4.4	Compressão de Imagens com Quantização Vectorial com <i>i-codebooks</i> .	85
4.4.1	Obtenção dos <i>codebooks</i>	86
4.4.2	Algoritmo para Compressão VQIND	89
4.4.3	Resumo do Algoritmo VQIND	93
4.4.4	Quantização Vectorial com <i>i-codebooks</i> de Índices <i>versus</i> Qua- lidade das Imagens Reconstruídas	95
4.4.5	Escalabilidade da Compressão	97
4.5	Compressão de Imagens com Quantização Vectorial e Lempel-Ziv . .	98
4.5.1	Algoritmo para Compressão VQZIP	98
4.5.2	Resumo do Algoritmo VQZIP	100
4.5.3	Quantização Vectorial com Lempel-Ziv <i>versus</i> Qualidade das Imagens Reconstruídas	102
4.6	Sumário	104
5	Simulações e Análise de Resultados	107
5.1	Introdução	107
5.2	Aspectos da Implementação	108
5.3	Simulações	110
5.3.1	Sequências Utilizadas	110
5.3.2	Segmentação das Imagens	111
5.3.3	Compressão dos Blocos	114
5.3.4	Reconstrução das Imagens	118
5.3.5	Informação Lateral a Enviar	120
5.4	Análise de Resultados	121
5.4.1	Desempenho do Algoritmo na Segmentação	122
5.4.2	Desempenho do Algoritmo na Compressão Total	123
5.4.3	Qualidade Visual das Imagens Reconstruídas	123
5.4.4	Qualidade das Imagens Relativamente ao PSNR	125

5.4.5	Qualidade das Imagens <i>versus</i> Metodologia Empregue	131
5.5	Aplicação a Outros Tipos de Imagem	132
5.5.1	Aplicação a Sequências Padrão	132
5.5.2	Aplicação a Sequências de Tráfego	139
5.6	Estabilidade da Imagem de Referência e Estabilidade dos <i>Codebooks</i> <i>versus</i> Taxa de Compressão	143
5.6.1	Refrescamento da Imagem de Referência	144
5.6.2	Refrescamento dos <i>Codebooks</i>	148
5.6.3	Compressão Média Total	150
5.7	<i>Codebooks</i> para a Quantização	152
5.8	Sumário	154
6	Conclusões	157
6.1	Resumo	157
6.2	Principais Contribuições	158
6.3	Trabalho Futuro	160
	Bibliografia	162

Lista de Figuras

2.1	Imagens de erro	14
2.2	Sequência de imagens MPEG	35
2.3	Compressão baseada no sinal	39
2.4	Codificação baseada em objectos	40
2.5	Compressão baseada na semântica	41
3.1	Deslocamento tri-dimensional	48
3.2	Divisão das imagens em blocos tridimensionais e aplicação dos filtros	58
3.3	Classificação dos Blocos - Sequência Açores 1	60
3.4	Classificação dos Blocos - Sequência Açores 2	61
3.5	Classificação dos Blocos - Sequência Açores 3	61
4.1	Estrutura da divisão da sequência	66
4.2	Etapa 1 do algoritmo de compressão	68
4.3	Etapa 2 do algoritmo de compressão	68
4.4	Disposição dos blocos b_t^n e dos <i>pixels</i> $X_{t,r}^n$ na sequência.	71
4.5	Determinação dos índices para os 3 blocos - Variante VQ2.5DN	72
4.6	Compressor multi-bloco baseado em <i>n-codebook</i> 2.5D	75
4.7	Descompressor multi-bloco baseado em <i>n-codebook</i> 2.5D	76
4.8	Disposição dos blocos b_t^g e dos <i>pixels</i> $Y_{t,z}$, $z = n(k \times k)$ na sequência.	78
4.9	Determinação dos índices para os 3 blocos - Variante 3DN	79
4.10	Compressor multi-bloco baseado em <i>n-codebook</i> 3D	81
4.11	Descompressor multi-bloco baseado em <i>n-codebook</i> 3D	82
4.12	Reconstrução da Sequência Açores 1 - Com VQ3DN	83
4.13	Reconstrução da Sequência Açores 3 - Com VQ3DN	83
4.14	Reconstrução da Sequência Açores 5 - Com VQ3DN	84
4.15	Movimento rectilíneo de um padrão	86
4.16	Exemplo de obtenção dos índices para a determinação do <i>i-codebook</i>	88

4.17	Codificador com <i>i-codebook</i>	90
4.18	Descodificador com <i>i-codebook</i>	91
4.19	Compressor multi-bloco baseado em VQIND	92
4.20	Descompressor multi-bloco baseado em VQIND	93
4.21	Reconstrução da Sequência Açores 1 - Com VQIND	95
4.22	Reconstrução da Sequência Açores 3 - Com VQIND	95
4.23	Reconstrução da Sequência Açores 5 - Com VQIND	96
4.24	Codificador com VQ e Lempel-Ziv	99
4.25	Descodificador com VQ e Lempel-Ziv	99
4.26	Compressor multi-bloco baseado em VQZIP	100
4.27	Descompressor multi-bloco baseado em VQZIP	101
4.28	Reconstrução da Sequência Açores 1 - Com VQZIP	103
4.29	Reconstrução da Sequência Açores 3 - Com VQZIP	103
4.30	Reconstrução da Sequência Açores 5 - Com VQZIP	103
5.1	Sequência sintetizada para teste	109
5.2	Interface da aplicação desenvolvida	110
5.3	Segmentação - Sequência Açores 1	112
5.4	Segmentação - Sequência Açores 3	112
5.5	Segmentação - Sequência Açores 5	112
5.6	Sequência Açores 2	113
5.7	Blocos classificados como 'com movimento'	114
5.8	Menu de selecção do tipo de compressão	115
5.9	Resultados da compressão com VQIND - Açores 1	116
5.10	Resultados da compressão com VQ3DN - Açores 3	117
5.11	Reconstrução - Sequência Açores 1	119
5.12	Reconstrução - Sequência Açores 3	119
5.13	Reconstrução - Sequência Açores 5	120
5.14	PSNR absoluto - Compressão com VQ3DN da Sequência Açores 2 . .	126
5.15	PSNR absoluto - Compressão com VQ3DN da Sequência Açores 3 . .	127
5.16	PSNR absoluto - Compressão com VQ3DN da Sequência Açores 5 . .	128
5.17	Blocos 'com movimento' - Sequência Açores 1	128
5.18	Blocos 'com movimento' - Sequência Açores 3	129
5.19	Blocos 'com movimento' - Sequência Açores 5	129
5.20	PSNR relativo - Compressão com VQ3DN da Sequência Açores 2 . .	129

5.21	PSNR relativo - Compressão com VQ3DN da Sequência Açores 3 . . .	130
5.22	PSNR relativo - Compressão com VQ3DN da Sequência Açores 5 . . .	130
5.23	Segmentação - Sequência Claire	133
5.24	Segmentação - Sequência Miss América	134
5.25	Segmentação - Sequência Salesman	134
5.26	Reconstrução - Sequência Claire	136
5.27	Reconstrução - Sequência Miss América	136
5.28	Reconstrução - Sequência Salesman	136
5.29	PSNR absoluto - Compressão com VQ3DN da Sequência Claire . . .	137
5.30	PSNR absoluto - Compressão com VQ3DN da Sequência Miss América	138
5.31	PSNR absoluto - Compressão com VQ3DN da Sequência Salesman .	138
5.32	PSNR relativo - Compressão com VQ3DN da Sequência Claire	138
5.33	PSNR relativo - Compressão com VQ3DN da Sequência Miss América	139
5.34	PSNR relativo - Compressão com VQ3DN da Sequência Salesman . .	139
5.35	Segmentação - Sequência Tráfego 1	140
5.36	Segmentação - Sequência Tráfego 2	140
5.37	Reconstrução - Sequência Tráfego 1	141
5.38	Reconstrução - Sequência Tráfego 2	141
5.39	PSNR absoluto - Compressão VQ3DN da Sequência Tráfego 1	142
5.40	PSNR absoluto - Compressão VQ3DN da Sequência Tráfego 2	142
5.41	PSNR relativo - Compressão VQ3DN da Sequência Tráfego 1	142
5.42	PSNR relativo - Compressão VQ3DN da Sequência Tráfego 2	143
5.43	PSNR Açores 1 - Diferentes refrescamentos da imagem de referência .	145
5.44	PSNR Açores 3 - Diferentes refrescamentos da imagem de referência .	145
5.45	PSNR Açores 5 - Diferentes refrescamentos da imagem de referência .	145
5.46	PSNR Açores 1 - Diferentes refrescamentos dos <i>codebooks</i>	149
5.47	PSNR Açores 3 - Diferentes refrescamentos dos <i>codebooks</i>	149
5.48	PSNR da composição de 3 sequências - Açores 1, 2 e 3	150
5.49	PSNR de Açores 1 - Reconstruída com <i>codebook</i> de Açores 3 e 5 . . .	151
5.50	PSNR de Açores 3 - Reconstruída com <i>codebook</i> de Açores 1 e 5 . . .	151
5.51	PSNR de Açores 5 - Reconstruída com <i>codebook</i> de Açores 1 e 3 . . .	151

Lista de Tabelas

5.1	Blocos classificados como 'com movimento' - Sequências Açores . . .	114
5.2	Taxas de compressão obtidas com compressão VQIND	116
5.3	Taxas de compressão obtidas com compressão VQ2.5DN	117
5.4	Taxas de compressão obtidas com compressão VQ3DN	118
5.5	Taxas de compressão obtidas com compressão VQZIP	118
5.6	Taxa de compressão obtida na segmentação com filtros de Gabor . . .	122
5.7	Comparação das taxas de compressão - Sequência Açores 1	123
5.8	Classificação visual - Sem conhecer original - Açores	124
5.9	Classificação visual - Conhecendo o original - Açores	124
5.10	PSNR absoluto médio - Compressão com VQ3DN - Sequências Açores	127
5.11	PSNR relativo médio - Compressão com VQ3DN - Sequências Açores	130
5.12	Blocos classificados como 'com movimento' - Sequências Padrão . . .	134
5.13	Taxas de compressão nominal e de transmissão - Sequências Padrão .	135
5.14	Classificação visual - Sem conhecer original - Padrão	136
5.15	Classificação visual - Conhecendo Original - Padrão	137
5.16	PSNR absoluto e relativo - Compressão VQ3DN - Sequências Padrão	139
5.17	Blocos classificados como 'com movimento' - Sequências Tráfego . . .	140
5.18	Taxas de compressão nominal e de transmissão - Sequências Tráfego .	141
5.19	PSNR absoluto e relativo - Compressão VQ3DN - Sequências Tráfego	143
5.20	Compressão nominal <i>versus</i> refrescamento imagens base - Açores 1 . .	146
5.21	Compressão nominal <i>versus</i> refrescamento imagens base - Açores 3 . .	146
5.22	Compressão nominal <i>versus</i> refrescamento imagens base - Açores 5 . .	147
5.23	Compressão efectiva <i>versus</i> refrescamento imagens base - Açores 1 . .	147
5.24	Compressão efectiva <i>versus</i> refrescamento imagens base - Açores 3 . .	147
5.25	Compressão efectiva <i>versus</i> refrescamento imagens base - Açores 5 . .	147

Capítulo 1

Introdução

1.1 Compressão de Dados

À medida que aumenta a interdependência entre os sistemas informáticos e os sistemas de telecomunicações aumenta também a quantidade de informação produzida e que é necessário transmitir. O tipo de dados a transmitir evoluiu de simples caracteres de texto, fáceis de transmitir, para tipos mais complexos de dados. Actualmente, é usual ter-se necessidade de transmitir informação composta por vários sub-tipos de informação, tais como texto, imagem e som, no contexto de aplicações designadas por multimédia, vídeo-telefone, vídeo-conferência, etc..

Este tipo de informação, com origem em diferentes meios, diversificou os problemas inerentes à sua manipulação. Por um lado, a grande quantidade de informação envolvida obriga ao recurso a poderosos sistemas informáticos que possibilitem o seu tratamento em tempo útil. Por outro lado, levantam-se problemas de armazenamento e/ou de transmissão.

A necessidade de se compatibilizar esta grande quantidade de informação com a relativa baixa capacidade dos meios existentes para o respectivo armazenamento e com as limitações de largura de banda dos canais de transmissão, levou ao desenvolvimento de técnicas de compressão de dados de forma a tornar mais eficiente o seu armazenamento e/ou transmissão.

A grande diferença entre o armazenamento e a transmissão dos dados comprimidos prende-se com a simetria das operações de compressão e de descompressão.

Geralmente, no armazenamento de dados as operações são assimétricas, no sentido em que a compressão é efectuada uma só vez e a descompressão várias. Um exemplo paradigmático desta situação é a gravação em CD de um filme. O filme é comprimido e é produzido um CD matriz a partir do qual são feitas inúmeras cópias as quais, por sua vez, podem ser decodificadas muitas vezes. No caso da transmissão dos dados, as operações de compressão/descompressão são normalmente simétricas, isto é, a informação é comprimida, transmitida e em seguida descomprimida.

Por outro lado, é também necessário que os codificadores/descodificadores utilizados nos sistemas de transmissão possam trabalhar quase em tempo real. É óbvio, que esta última premissa depende não só da compressão a efectuar, nomeadamente da taxa de compressão que se pretenda a qual, normalmente, quanto maior for obriga a um maior processamento e conseqüente aumento do tempo necessário a esse processamento. Também depende do próprio canal de transmissão que, se tiver um débito muito baixo requer uma grande taxa de compressão, pode obrigar a um atraso na recepção dos dados.

Do ponto de vista do utilizador, a compressão de dados permite que os meios de armazenamento sejam encarados como tendo capacidades de armazenamento bastante maiores da que na realidade possuem, ou que o canal de transmissão que está a utilizar tenha uma largura de banda muito superior à que efectivamente tem.

Deste modo, quando a informação disponível aumenta, a necessidade de métodos mais eficientes para representar essa informação também aumenta. O objectivo da compressão de dados é pois o de disponibilizar a maneira mais eficiente de representar informação. Este objectivo é acompanhado pelo desenvolvimento de técnicas para explorar diferentes tipos de estruturas existentes nos dados. Diferentes formas de informação têm também os seus tipos específicos de estrutura. Um método de compressão, ao explorar a estrutura própria de determinado tipo de informação, pode revelar-se bastante bom para esse tipo de informação mas menos adequado para outros tipos.

A compressão de dados poderá ser feita sem ou com perda de informação. Os

métodos conhecidos para a compressão de dados sem perda de informação não permitem obter factores de compressão muito grandes, pelo que é vulgar recorrer-se à compressão com perda quando há necessidade de se obter grandes taxas de compressão. No caso da compressão de texto, de dados ou de programas informáticos, não são admissíveis perdas de informação e portanto usa-se de preferência o primeiro tipo de métodos. No caso da voz e também no caso de imagens já é mais admissível fazer-se compressão com perda. No caso da voz esta perda resulta, normalmente, da supressão da informação presente em frequências não perceptíveis pelo ouvido humano. No caso de imagens resulta da supressão de redundância. No caso particular do vídeo, o qual é constituído por sequências de imagens amostradas no tempo, actualmente é mesmo impossível, caso se queiram obter elevadas taxas de compressão que possibilitem a sua transmissão por canais de banda estreita, fazer-se a sua compressão sem perda de resolução e de qualidade das sequências de vídeo.

1.2 Formulação do Problema

Neste trabalho, estudam-se e propõem-se técnicas de compressão de vídeo para o caso particular da compressão de sequências de imagens submarinas onde se regista actividade hidrotermal. Esta actividade manifesta-se pela emissão de gases os quais se deslocam para a superfície em colunas de bolhas. Estes algoritmos de compressão foram desenvolvidos de modo a possibilitar a transmissão deste tipo de sequências para a superfície através do canal acústico submarino.

O cenário de utilização de uma aplicação baseada nestes algoritmos seria o de um veículo autónomo submarino, AUV (*Autonomous Underwater Vehicle*), o qual faz a captação de imagens vídeo de zonas hidrotermais e as transmite para a superfície, através do canal acústico submarino. As imagens captadas e assim transmitidas permitiriam guiar o AUV para a zona de emissão dessas bolhas. Dado o relativamente baixo débito deste canal, da ordem dos $20 - 30 Kbps$, é necessário que as sequências sejam comprimidas com altas taxas de compressão, pelo que é necessário fazer uma

massiva compressão das sequências de imagens.

Na missão preconizada para o veículo submarino, este irá operar no fundo do mar em zonas onde há uma forte actividade sísmica. Relacionada com esta actividade sísmica existem concentrações de bancos submarinos emissores de bolhas gasosas provenientes de actividade hidrotermal e que interessa analisar.

Há duas componentes vídeo neste veículo. Numa, o veículo deverá, atingida a zona de emanações gasosas, proceder a filmagens do meio ambiente, armazenando essa informação para posterior tratamento à chegada à superfície. A outra componente será utilizada para guiar o veículo para estas zonas onde existam emanações gasosas. As imagens desta última componente devem ser enviadas para a superfície para se poder orientar o veículo para as zonas de interesse, isto é, zonas de actividade hidrotermal.

O veículo submarino irá operar sem qualquer ligação física com a superfície, efectuando-se todas as comunicações por meio do canal acústico submarino, o que, atendendo à baixa taxa de transferência de dados passível de ser atingida neste canal, coloca grandes dificuldades do ponto de vista prático à transmissão destas imagens.

Estas sequências são constituídas por imagens cujos objectos que se deslocam dentro do campo das imagens se movimentam a grande velocidade e não produzem um contraste significativo. A isto acresce o facto de haver necessidade de se ter uma muito grande taxa de compressão, de modo a tornar viável a sua transmissão através do canal acústico submarino ou seja a uma muito baixa cadência. Deste modo, tem-se necessidade de, por um lado, se utilizar um detector de movimentos que consiga nestas imagens de muito baixo contraste extrair, por segmentação, a informação respeitante às bolhas das formações gasosas que se deslocam para a superfície e, por outro lado, efectuar uma compressão muito alta dos blocos segmentados.

Nas imagens a transmitir o movimento das bolhas é essencialmente no sentido vertical e as alterações no cenário do fundo são relativamente pequenas durante períodos de tempo bastante razoáveis. Preconiza-se por isso uma técnica em que

se utilizam filtros que permitam captar o movimento no sentido vertical e, desse modo, isolar determinadas zonas das imagens correspondentes ao movimento das bolhas. De seguida comprimem-se e transmitem-se só os blocos de imagem correspondentes a esse movimento. *A posteriori*, faz-se a reconstrução da sequência com base numa imagem de referência à qual se acrescentam sucessivamente os blocos correspondentes a zonas de movimento.

1.3 Compressão de Imagens Submarinas com Atividade Hidrotermal

Apresentavam-se à partida algumas dificuldades técnicas que foi preciso superar, nomeadamente o facto destas imagens apresentarem muita textura e as bolhas gasosas serem, devido à sua natureza, praticamente transparentes o que punha dificuldades a nível do seu contraste com o resto da imagem. Existem sequências em que a presença destas bolhas gasosas praticamente só é notada nas imagens vistas em sequência, sendo praticamente indistinguíveis numa análise isolada de cada imagem. A acrescentar a isto, saliente-se ainda o facto de a velocidade ascensional das bolhas ser bastante grande. Isto põe grandes problemas quando se pretende fazer a segmentação das imagens com base no movimento pois, como referido, a velocidade é grande e o contraste quase nulo.

Uma condicionante no desenvolvimento de uma técnica de compressão de imagens para utilização no canal acústico submarino é que se pretende enviar sequências de imagens, vídeo, e não somente imagens paradas do fundo submarino. Esta condicionante é muito forte dado a enorme quantidade de dados necessária para a transmissão de vídeo e o baixo débito deste canal.

Também era desejável que o desempenho do algoritmo permitisse a sua utilização em tempo quase real, de modo a permitir a condução do veículo submarino para as zonas onde se detectem emanações gasosas.

Os algoritmos desenvolvidos mostraram-se, de um modo geral, capazes de satisfazer estes requisitos. Permitem obter altas taxas de compressão e manter uma

boa qualidade das imagens reconstruídas, quer quando esta qualidade é analisada do ponto de vista da qualidade visual, quer quando é analisada do ponto de vista da qualidade objectiva baseada na relação sinal-ruído (de pico), PSNR (*Peak Signal Noise Ratio*).

Nestes algoritmos podem considerar-se duas fases complementares: numa primeira fase, segmentam-se as imagens em zonas onde exista movimento devido às bolhas gasosas e zonas onde não exista este movimento, *background*, utilizando filtros espaço-temporais direccionais de uma classe particular, os filtros de Gabor; numa segunda fase, para se atingirem as taxas de compressão necessárias à transmissão, faz-se a compressão dos blocos seleccionados. Para esta segunda fase desenvolveram-se estratégias alternativas que, complementando a segmentação efectuada na primeira fase, permitiram obter as altas taxas de compressão necessárias.

Estas estratégias foram baseadas em Quantização Vectorial, a qual foi adaptada de modo a permitir a compressão em simultâneo de mais do que um bloco de imagem. As alternativas baseiam-se na utilização da Quantização Vectorial com *codebooks* com múltiplos planos, VQN, e na Quantização Vectorial com dois *codebooks* distintos, um de níveis de cinzento e outro de índices, VQIND. No caso da quantização vectorial com *codebooks* com múltiplos planos, atendendo ao modo como esses *codebooks* com múltiplos planos são gerados, fizeram-se duas variantes: quantização vectorial com múltiplos planos em 2.5D, VQ2.5DN, e quantização vectorial com múltiplos planos em 3D, VQ3DN.

Para se poder fazer uma comparação de resultados implementou-se também uma versão que consiste na utilização da Quantização Vectorial, seguida de compressão baseada no algoritmo de Lempel-Ziv aplicada à compressão dos índices obtidos da quantização vectorial. Designou-se esta técnica por compressão VQZIP. Todas estas técnicas serão exaustivamente descritas ao longo desta dissertação.

Exceptuando a VQZIP, qualquer uma das outras 3 técnicas de compressão, associada à segmentação efectuada na primeira fase, permitiram obter bons resultados

numa perspectiva de taxa de compressão elevada *versus* qualidade das imagens reconstruídas.

Destas técnicas, a que permitiu obter a taxa de compressão mais elevada e a melhor qualidade das imagens foi a baseada na Quantização Vectorial com *codebooks* com múltiplos planos em 3D, VQ3DN.

1.4 Principais Contribuições

As contribuições desta tese podem ser consideradas de dois pontos de vista : contribuições ligadas à segmentação de imagens de sequências de vídeo e contribuições mais directamente relacionadas com a compressão em si.

Assim, podemos referir:

- A adaptação dos filtros de Gabor à segmentação selectiva deste tipo particular de imagens;
- Desenvolvimento de estratégias de compressão com Quantização Vectorial Multi-Bloco, as quais possibilitam a compressão, em simultâneo, de mais do que um bloco de imagem. Neste contexto foram desenvolvidas as seguintes estratégias:
 - Compressão Vectorial com *n-codebooks*:
 - * Compressão VQ2.5DN.
 - * Compressão VQ3DN.
 - Compressão Vectorial com *i-codebooks*, VQIND.

1.5 Organização da Dissertação

Esta dissertação está organizada em seis capítulos. No capítulo 2, faz-se uma introdução ao tema da compressão de dados, com especial relevância para a compressão de imagem, estática ou em sequência, abordando-se algumas das principais técnicas conhecidas. Referem-se as relações entre a Teoria da Informação e a

compressão de dados e pormenorizam-se algumas técnicas gerais de compressão de imagem e vídeo.

As estratégias particulares, desenvolvidas para a segmentação e a compressão das sequências de imagens submarinas com registo de actividade hidrotermal, serão descritas nos capítulos 3 e 4. Faz-se a formulação do problema em causa e analisam-se as particularidades inerentes à transmissão e reconstrução destas imagens, fazendo-se também uma abordagem à percepção do movimento e aos filtros direccionais (de Gabor) sensíveis ao movimento. Assim, a segmentação deste tipo particular de imagens submarinas com registo de actividade hidrotermal é descrita em detalhe no capítulo 3. A compressão e a descompressão são abordadas no capítulo 4. Aqui, são descritos os algoritmos propostos baseados em Quantização Vectorial multibloco que permitem a compressão/descompressão simultânea de vários blocos de imagem.

No capítulo 5, faz-se a análise dos resultados de simulações efectuadas utilizando-se uma aplicação desenvolvida com base nos algoritmos propostos. Para além de se apresentarem resultados obtidos para a segmentação e reconstrução das imagens submarinas, mostram-se também, para fins de comparação, resultados obtidos com imagens de outros tipos: sequências padrão e de tráfego.

Finalmente, no capítulo 6, são descritas as principais conclusões a que se chegou. Apontam-se direcções de trabalho com vista a um eventual desenvolvimento das técnicas agora apresentadas e indicam-se possíveis alternativas de aplicação das técnicas propostas a outro tipo de imagens e de cenários de utilização.

Capítulo 2

Compressão de Imagem

2.1 Introdução

As imagens são diferentes de outras fontes, pois muitas dessas outras fontes contêm algum mecanismo de suporte que induz estrutura. Por exemplo, toda a fala é gerada por mecanismos similares, cuja mecânica tem limitações físicas, que fornecem a estrutura particular da sua saída. A linguagem é regulada por muitas regras, as quais fornecem à fonte de texto uma significativa quantidade de estrutura. As linguagens de computador são sempre mais restritas e conseqüentemente têm mais estrutura. As imagens, no entanto, não partilham destas características. De um modo geral, as imagens não são geradas por nenhum tipo de mecanismo restritivo. Uma imagem de uma pintura abstracta, uma imagem de abertura sintética de radar, SAR (*Synthetic Aperture Radar*) e um retrato têm muito poucas similaridades estruturais. Por causa desta falta de características unificadoras, foram desenvolvidas diferentes maneiras de se caracterizarem as imagens.

Quando falamos de técnicas de compressão ou algoritmos de compressão estamos na realidade a referir-nos a dois algoritmos distintos: um é o algoritmo de compressão propriamente dito, o qual a partir de uma entrada \mathcal{X} , por exemplo uma imagem, gera uma sua representação \mathcal{X}_c , que requer poucos *bits* para ser representada; o outro algoritmo será o algoritmo de descompressão ou reconstrução, que opera sobre a representação comprimida \mathcal{X}_c para gerar a imagem reconstruída \mathcal{Y} . Por uma questão de convenção, referir-nos a ambos os algoritmos por algoritmos

de compressão, se bem que com o sentido acima referido.

2.2 A Teoria da Informação e a Compressão de Dados

Na codificação de dados e em particular na codificação de imagens há dois aspectos importantes a ter em conta. Estes aspectos são a redundância e a irrelevância da informação, os quais possibilitam fazer a redução de dados na codificação de fonte.

A redundância, ou mais concretamente a redundância estatística, relaciona-se com a correlação e a previsão da informação encontradas no tempo, espaço e espectro de uma sequência de imagens. Está directamente relacionada com a forma artificial como a cena real é representada e deve-se ao facto de algumas fontes produzirem sinais que são representados com bastante mais *bits* do que os estritamente necessários em termos de percepção humana. A codificação de fonte pode reduzir esta redundância e conduzir a uma nova representação da informação utilizando menos *bits* para a sua representação. Porque somente a informação redundante foi removida, a verdadeira informação não foi perdida.

Por sua vez, a irrelevância, ou mais concretamente a irrelevância do detalhe, é uma forma de redundância subjectiva e consiste na incapacidade de percepção, para lá de uma determinada resolução, dos vários elementos de uma imagem ou sequência de imagens pelo sistema visual humano. Ou seja, duas imagens podem ser indistinguíveis pelo olho humano, apesar de serem fisicamente diferentes, porque as diferenças são consideradas irrelevantes.

Qualquer esquema utilizado para eficientemente representar, transmitir ou gravar informação está baseado na chamada codificação de fonte. Exemplos de codificação de fonte incluem o código Morse da telegrafia, a limitação da largura de banda na voz sobre telefone e a projecção sequencial de imagens estáticas para criar a ilusão de movimento num filme. A compressão de dados é então um subcapítulo da

codificação de fonte.

Em sequências de vídeo muita informação está correlacionada. Normalmente, quando se faz um varrimento ao longo das linhas de uma imagem não há mudanças abruptas de *pixel* para *pixel*, podendo pois inferir-se que estes *pixels* estarão espacialmente correlacionados. Pelo contrário, quando se analisa uma página de texto observam-se transições rápidas entre os *pixels* a branco e a preto que formam as letras. Deste modo, a exploração da correlação espacial numa página de texto não conduz normalmente a uma efectiva codificação da fonte. No caso do vídeo, e não considerando as mudanças de cena, não há mudanças abruptas de uma imagem para outra dentro de uma sequência, pelo que se pode considerar a sequência de vídeo como temporalmente correlacionada.

Na codificação de fonte podemos considerar codificação sem e com perda. Na codificação sem perda, o sinal original pode ser reconstruído sem erros. A informação pode ser então comprimida sem degradação. Isto acontece porque, na codificação sem perda, somente a informação redundante é removida. Todavia, muitas vezes não é necessário proceder-se a uma reconstituição perfeita do sinal podendo-se, então, fazer a codificação de fonte com perda.

Os sinais de audio e de vídeo são exemplos paradigmáticos desta situação, pois muita da informação presente nestes tipos de sinais é irrelevante para os *sensores humanos* da audição e da visão. Sendo irrelevante, esta informação pode ser removida pelo processo de codificação. Este tipo de codificação com perda introduz distorção no sinal a qual, sendo controlada dentro de limites adequados, não é humanamente perceptível.

Os esquemas de compressão exploram, normalmente, as limitações naturais do olho ou do ouvido humano para ver ou escutar determinadas frequências. Sons sintetizados podem conter frequências fora do intervalo de audição humana, enquanto que gráficos gerados computacionalmente podem ter transições bruscas ou cores altamente saturadas, as quais são indistinguíveis pelo olho humano.

Dado que o destinatário final de um sinal de vídeo é o sistema visual humano,

a compreensão das suas características principais é necessária para o desenho de um sistema de compressão eficiente. Pelas razões expostas, a única informação que deverá ser transmitida com fiabilidade é a que seja perceptível pelo observador. A transmissão de informação, para além da estritamente necessária, só implica desperdício de recursos em termos de banda de transmissão, sem acrescentar nenhum benefício para o destinatário.

Assim, aquando do desenvolvimento de um sistema de compressão de imagens, as características do sistema visual humano devem ser tidas em conta. É pois importante considerar-se a dependência da visibilidade do ruído de reconstrução relativamente:

- à intensidade do fundo: decresce com o crescimento da intensidade do fundo;
- ao conteúdo espectral;
- à vizinhança dos contornos: é muito menos visível junto aos contornos;
- ao movimento: é menos visível nas regiões com movimento.

Uma possível aplicação da compressão de imagens é na transmissão digital de imagens de televisão com boa qualidade, através de um canal televisivo terrestre com uma largura de banda limitada. Infelizmente, factores de compressão muito altos não poderão ser obtidos sem uma redução na qualidade das imagens, manifestada sob a forma de erros de codificação. De modo simplificado, o factor de compressão requerido é dado por:

$$\frac{\textit{Taxa de Geração de Dados}}{\textit{Capacidade do Canal de Dados}} = \textit{Factor de Compressão}. \quad (2.2.1)$$

Apesar dos intensos esforços que têm sido feitos, para algumas imagens em sequências de televisão, ainda não se conseguiu atingir factores de compressão significativos sem alguma perda de qualidade. Se o factor de compressão for substancialmente aumentado, todas as técnicas conhecidas provocam degradação das imagens.

Infelizmente, esta degradação não se limita a uma aceitável perda de informação irrelevante, que a visão humana possa reconstruir.

2.3 Medição da Qualidade

Não existe uma medida universal, através da qual se possa quantificar, num simples parâmetro, a qualidade de qualquer técnica de compressão. Entre outras medidas [1, 2, 3], uma medida possível é dada pelo erro quadrático médio entre a imagem reconstruída e a original. No entanto, a sua aplicação não é pacífica nem aceite universalmente, dado que, não destrinça com fidelidade os erros de reconstrução das imagens, como faz o sistema visual humano.

Um dos objectivos mais importantes a ter em conta no desenvolvimento de qualquer metodologia de compressão de imagens é a minimização da percepção da distorção por parte do sistema visual humano. Há muitas questões subjectivas na análise da qualidade das imagens reconstruídas e que não podem ser quantificadas por um método objectivo de medição da distorção.

Há erros de reconstrução que, do ponto de vista da percepção visual, são classificados como erros grosseiros e que não são contabilizados pelas medidas baseadas no erro quadrático médio. O inverso também acontece, isto é, há erros contabilizados pelos métodos baseados no erro quadrático médio e que são imperceptíveis ao olho humano.

Por exemplo, na figura 2.1, um observador humano consideraria que a imagem do meio seria a mais semelhante com a imagem da esquerda, a original. No entanto, um sistema que fizesse uma análise comparativa, *pixel a pixel*, entre a imagem de referência, à esquerda, e as outras duas, classificaria a da direita como a mais correcta, pois detectaria uma maior sobreposição de *pixels* entre estas duas. Poderá considerar-se que a análise efectuada pela visão humana seja feita com base numa comparação global entre as imagens, o que neste caso é verdade apesar de, quer as linhas horizontais quer as verticais, estarem afastadas um *pixel* relativamente à

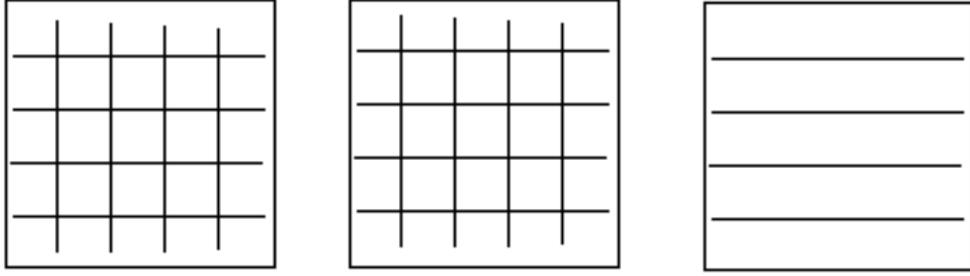


Figura 2.1: Imagens de erro

posição original. Este afastamento é que provocaria a má classificação aquando da comparação estrita entre *pixels* em imagens similares.

Apesar das limitações dos métodos baseados no erro quadrático médio, MSE (*Mean Square Error*), entre os níveis de cinzento das imagens reconstruídas e os das respectivas imagens originais, é vulgar o recurso a estes métodos para se obter uma medida objectiva da qualidade. Assim, bem ou mal, o MSE é vulgarmente considerado como critério de qualidade dos métodos de compressão.

Sendo N^2 o número de *pixels* da imagem, x e \hat{x} *pixels*, respectivamente, da imagem original e da reconstruída e σ^2 a variância da imagem original, o MSE é dado por:

$$MSE = \frac{1}{N^2} \sum_{imagem} (x - \hat{x})^2 \quad (2.3.1)$$

e algumas das medidas de distorção referidas são [4]:

$$\frac{\sigma^2}{MSE} \quad (2.3.2)$$

$$\frac{\sum_{imagem} x^2}{MSE} \quad (2.3.3)$$

$$\frac{x_{max}^2}{MSE} \quad (2.3.4)$$

onde a primeira medida (2.3.2) relaciona o MSE com a variância da imagem original; a medida (2.3.3) relaciona o MSE com a energia média na imagem; e a medida (2.3.4)

relaciona o MSE com a energia máxima na imagem, x_{max}^2 . Esta terceira medida é a mais utilizada e mede a relação sinal-ruído de pico, PSNR (*Peak Signal-Noise Ratio*).

2.4 Compressão de Imagens

Dada a relação existente entre a compressão de imagens estáticas e a compressão de sequências de imagens, serão referidos, nesta secção, alguns dos principais métodos para compressão de imagens, sem ou com perda de informação. O primeiro grupo de técnicas não introduz distorção, pois faz-se a redução do número de *bits* de codificação da imagem pela redução da redundância. No entanto, as taxas de compressão passíveis de serem obtidas com este tipo de compressão são relativamente baixas. Pelo contrário, com as técnicas do segundo grupo, os chamados compressores com perda, consegue obter-se taxas de compressão bastante elevadas, se bem que à custa da distorção das imagens.

2.4.1 Compressão de Imagens Sem Perda

Essencialmente, a compressão de imagens sem perda baseia-se na redução de redundância, não envolvendo assim nenhuma perda de informação. A compressão sem perda é geralmente usada para dados tais como texto, dados gerados por computador e em alguns tipos especiais de imagem e vídeo, onde não é desejável haver perda de informação como é o caso, por exemplo, de imagens médicas.

As principais técnicas utilizadas na compressão sem perdas [4, 5], são:

Codificação de Huffman - Os códigos gerados usando esta técnica são designados por códigos de Huffman. Estes códigos são códigos com prefixo e são óptimos para um dado modelo, isto é, para uma dada distribuição de probabilidades.

Codificação de Huffman Adaptativa- A codificação de Huffman requer o conhecimento *a priori* das probabilidades dos símbolos da fonte. Se este conhecimento não estiver disponível, a codificação de Huffman adaptativa torna-se um

procedimento em dois passos: estimação da estatística da fonte e codificação.

Codificação Aritmética- A codificação aritmética é outro método alternativo para se gerar códigos de comprimento variável. A codificação aritmética é especialmente indicada para a codificação de fontes com alfabetos pequenos, tais como, fontes binárias e alfabetos com distribuição de probabilidades assimétrica.

Na codificação aritmética, um único identificador é gerado para a sequência a codificar. Este identificador corresponde a uma fracção binária, a qual se torna o código binário para a sequência. Em termos práticos a geração do identificador e do código binário são o mesmo processo. Numa primeira fase, um único identificador é gerado para a sequência de símbolos [5].

Este identificador é então um código binário único. Um só código aritmético pode ser gerado para uma sequência de tamanho m , sem haver necessidade de se gerar palavras de código para todas as sequências de tamanho m .

Uma maneira de se representar uma sequência de símbolos é representá-la por um número pertencente ao intervalo unitário $[0, 1)$. Devido ao facto da quantidade de números dentro deste intervalo unitário ser infinita, torna-se possível associar um único identificador para cada sequência de símbolos que surja.

Nestas técnicas assumimos que as entradas eram independentes e identicamente distribuídas. Caso isto não se verifique, estas técnicas não serão as mais adequadas.

Técnicas de Dicionário - Em muitas aplicações, a saída da fonte consiste na repetição de padrões. Uma aproximação correcta para codificar estas fontes é construir uma lista, ou dicionário, da frequência de ocorrência de tais padrões. Quando estes padrões aparecerem na saída da fonte, serão codificados com referência a este dicionário. Se algum padrão não aparecer no dicionário, então poderá ser codificado usando algum outro método e para esse efeito, divide-se a entrada em duas classes: uma dos padrões mais frequentes e outra dos padrões menos frequentes.

Construída esta lista dos padrões que mais frequentemente ocorrem codificam-se esses padrões transmitindo-se o seu índice na lista.

Para que esta técnica seja efectiva, a classe dos padrões mais frequentes e por conseguinte o tamanho do dicionário, deverá ser muito mais pequeno do que o número de todos os possíveis padrões.

Deste modo, estas técnicas estão bem adaptadas a fontes que gerem um número relativamente pequeno de padrões e que os repitam frequentemente, tais como, fontes de texto e de instruções de computador [4, 5, 6].

2.4.2 Compressão de Imagens Com Perda

Quantização Escalar- Nesta técnica, cada saída da fonte pode ser representada usando um valor de entre um pequeno conjunto de valores ou *codewords*. O número de possíveis diferentes valores de saída das fontes é geralmente muito maior do que o número de *codewords* disponíveis para representá-los. Este processo de representar um grande, possivelmente infinito, conjunto de valores através de um pequeno conjunto de valores é chamado de quantização [5].

Para exemplificar a aplicação deste método, considere-se uma fonte que gera números entre -10.0 e 10.0 e represente-se cada saída da fonte com o valor inteiro mais próximo dele. Existe um número infinito de valores entre -10.0 e 10.0 , que vão ser representados por um conjunto que contém unicamente 21 valores $\{-10, -9, \dots, 0, \dots, 9, 10\}$. Deste modo, dado um destes valores de reconstrução, nunca se poderá recuperar exactamente a saída da fonte, pois há neste esquema uma infinidade de valores reais que poderiam corresponder a esse valor de reconstrução. Assim, neste esquema de compressão há perda de informação.

O conjunto de entradas e saídas do quantizador pode ser constituído por escalares ou por vectores. Se forem escalares, são designados por quantizadores escalares; se forem vectores serão designados por quantizadores vectoriais.

Os quantizadores escalares poderão ser uniformes. Nestes quantizadores, todos os intervalos de quantização terão o mesmo comprimento. Como os quantizadores

uniformes podem estar desajustados às estatísticas dos sinais a quantizar, nomeadamente, quando a variância do sinal de entrada é diferente da variância assumida, ou quando o tipo de distribuição do sinal é diferente do tipo de distribuição assumido, é vulgar utilizar-se outros tipos de quantizadores.

Assim, de modo a diminuir a distorção média que este desajuste nos intervalos de quantização possa introduzir, podem-se utilizar quantizadores que aproximem melhor as entradas a regiões de alta probabilidade, com um custo de uma pior aproximação às regiões de mais baixa probabilidade. Isto pode ser feito, utilizando-se intervalos de quantização menores para as regiões que tenham maior probabilidade. Um quantizador com intervalos não uniformes é designado por quantizador não-uniforme. O comprimento dos vários sub-intervalos de quantização pode ser pré-estabelecido, como parâmetro do algoritmo de quantização, ou pode ser determinado em função dos dados, isto é, depois destes serem analisados numa primeira fase.

Neste tipo de quantizadores não-uniformes, existem duas maneiras principais para se adaptarem os parâmetros de quantização às estatísticas do sinal: quantização adaptativa *forward* ou quantização adaptativa *backward*.

Na abordagem adaptativa *forward* a saída da fonte é dividida em blocos de dados. Cada bloco é analisado antes de quantizado e os parâmetros de quantização são adaptados às suas particularidades. Estes parâmetros de quantização serão transmitidos como informação lateral complementar. Na quantização adaptativa *backward* a adaptação é feita baseada na saída do quantizador. Como esta informação está disponível, tanto do lado do transmissor como do lado do receptor, não é necessário enviar informação lateral.

Quantização Vectorial - Em lugar de se considerarem valores escalares isolados, estes podem ser agrupados em blocos e deste modo obtêm-se algoritmos de compressão mais eficientes, os chamados quantizadores vectoriais. Neste tipo de quantizadores, um bloco de dados a comprimir é comparado com os vectores do quantizador, sendo transmitido o índice do vector mais semelhante ao bloco de dados em questão. Relativamente aos quantizadores escalares, o ganho reside no facto

de que, aqui, um índice refere-se a um conjunto de dados [5, 7, 8].

Para se construírem estes quantizadores, geram-se conjuntos representativos das sequências dos dados e de seguida grupos de saídas da fonte são codificados relativamente a estes conjuntos. Para o caso de imagens, considerem-se blocos de \mathcal{L} *pixels* de uma imagem e trata-se cada *pixel* como um componente de um vector de dimensão \mathcal{L} . Este vector de saídas da fonte forma a entrada para o quantizador vectorial. Tanto no codificador como no descodificador vectorial, existe um conjunto de vectores \mathcal{L} – *dimensional*, designado por *codebook* do quantizador vectorial. Estes vectores do *codebook*, designados por *codevectors*, são seleccionados para serem representativos dos vectores gerados pela fonte. Cada *codevector* é associado a um índice binário. No codificador, o vector de entrada é comparado com cada *codevector* de modo a encontrar-se o *codevector* mais próximo do vector de entrada. Os elementos deste *codevector* são os valores quantizados das saídas da fonte. De modo a informar o descodificador sobre qual foi o *codevector* determinado como sendo o mais próximo ao vector de entrada, transmite-se ou armazena-se o índice binário do *codevector*. Como o descodificador tem o mesmo *codebook*, dado o índice binário, pode obter-se o *codevector* e efectuar a reconstrução.

A quantidade de compressão pode ser descrita em termos de taxa de compressão, a qual pode ser medida em *bits* por amostra [5]. Suponhamos um *codebook* de dimensão K e em que os vectores de entrada são de dimensão \mathcal{L} . De modo a informar o descodificador sobre o *codevector* seleccionado, é necessário usar-se $\lceil \log_2 K \rceil$ *bits*, isto é, o número de *bits* por vector é $\lceil \log_2 K \rceil$ *bits*. Por exemplo, se o *codebook* contiver 256 *codevectors*, são necessários 8 *bits* para especificar qual dos 256 *codevectors* foi seleccionado pelo codificador. Como cada *codevector* contém os valores de reconstrução para \mathcal{L} amostras da fonte, o número de *bits* por amostra será $\lceil \log_2 K \rceil / \mathcal{L}$. Deste modo, a taxa para um quantizador vectorial \mathcal{L} – *dimensional*, usando-se um *codebook* de dimensão K , será $\lceil \log_2 K \rceil / \mathcal{L}$ [5].

Como medida de distorção, pode-se usar o erro quadrático médio. Num *codebook* \mathcal{C} contendo K *codevectors* $\{Y_i\}$, o *codevector* Y_j mais próximo do vector de entrada

X será:

$$\|X - Y_j\|^2 \leq \|X - Y_i\|^2, \forall Y_i \in \mathcal{C},$$

com $X = (x_1, x_2, \dots, x_{\mathcal{L}})$ e

$$\|X\|^2 = \sum_{i=1}^{\mathcal{L}} x_i^2.$$

Os valores de saída do quantizador são frequentemente designados por níveis. Assim, quando nos referimos a um quantizador com K valores de saída ou *codevectors*, estamos a referir-nos a um quantizador de $K - \text{níveis}$.

Se a saída da fonte for correlacionada, os valores dos vectores de saída da fonte tendem a agruparem-se. Seleccionando os *codevectors* mais próximos destes agrupamentos, tem-se uma representação mais precisa da saída da fonte. Assim, para a mesma taxa, em *bits* por amostra, a utilização de quantizadores vectoriais resulta numa menor distorção do que a introduzida por um quantizador escalar.

Na quantização vectorial, as fronteiras de decisão entre os níveis de reconstrução não podem ser descritas, tão facilmente, como no caso da quantização escalar. No entanto, se conhecermos a medida de distorção, o simples conhecimento dos *codevectors* dá-nos informação suficiente para realizar o processo de quantização. Para além de definirmos a regra de quantização em termos de fronteiras de decisão, podemos definir a regra de quantização da seguinte maneira:

$$\mathcal{Q}(X) = Y_j \iff d(X, Y_j) < d(X, Y_i), \forall i \neq j.$$

Para o caso da entrada X ser equidistante de dois pontos de saída podemos considerar a saída que tiver o menor índice.

As regiões de quantização V_j podem então ser definidas como:

$$V_j = \{X : d(X, Y_j) < d(X, Y_i), \forall i \neq j\}.$$

Deste modo, o quantizador fica completamente definido pelos *codevectors* e por uma medida de distorção. O conjunto dos *codevectors* é chamado *codebook* do quantizador e o processo de determinação destes *codevectors* é designado por *treino do codebook*.

Um dos maiores problemas relacionados com a utilização de quantizadores vectoriais prende-se com a obtenção do *codebook*, pois o desempenho de um quantizador vectorial está directamente relacionado com o *codebook* utilizado. Existem vários algoritmos para a geração de *codebooks* [9, 10]. Iremos referir mais detalhadamente o algoritmo LBG [10], por ser o mais utilizado devido à sua relativa simplicidade e eficiência. Este algoritmo é baseado no algoritmo de Lloyd [11] o qual é usado em aplicações de reconhecimento de padrões e consiste num procedimento de agrupamento similar ao do algoritmo *k - means*. No esquema de compressão proposto, utiliza-se uma variação deste algoritmo LBG para a geração dos *codebooks*, dado que aí vai ser necessário gerar *codebooks* que permitam a compressão de vários blocos de imagens em simultâneo.

O algoritmo *k - means* funciona da seguinte maneira [5]:

Dado um grande conjunto de vectores de saída de uma fonte, conhecido como conjunto de treino e um conjunto inicial de k padrões representativos, associar cada elemento do conjunto de treino ao padrão representativo mais próximo. Depois de um elemento ser associado, o padrão representativo é actualizado calculando-se o centróide do conjunto de vectores de treino associado a ele. Quando o processo de associação estiver completo, teremos k grupos de vectores agrupados cada um à volta dos *codevectors*.

Lloyd [11], utilizou um procedimento similar para gerar um quantizador escalar optimizado para a função de densidade de probabilidade. A diferença deste algoritmo em relação ao algoritmo *k - means* reside no facto de, em lugar de usar um conjunto de treino, assumir-se que a distribuição é conhecida. O algoritmo de Lloyd consiste nos passos seguintes [5]:

1. Começar com um conjunto inicial de valores de reconstrução $\{y_i^{(0)}\}_{i=1}^M$. Definir $k = 0$, $D^{(0)} = 0$ e seleccionar um limiar ϵ .

2. Procurar as fronteiras de decisão:

$$b_j^{(k)} = \frac{y_{j+1}^{(k)} + y_j^{(k)}}{2}, j = 1, 2, \dots, M - 1.$$

3. Calcular a distorção

$$D^{(k)} = \sum_{i=1}^M \int_{b_{i-1}^{(k)}}^{b_i^{(k)}} (x - y_i)^2 f_X(x) dx.$$

4. Se $D^{(k)} - D^{(k-1)} < \epsilon$, parar, se não continuar.

5. Fazer $k = k + 1$. Calcular os novos valores de reconstrução:

$$y_j^{(k)} = \frac{\int_{b_{j-1}^{(k-1)}}^{b_j^{(k-1)}} x f_x(x) dx}{\int_{b_{j-1}^{(k-1)}}^{b_j^{(k-1)}} f_x(x) dx}.$$

Voltar ao passo 2.

Este algoritmo de Lloyd foi generalizado para o caso em que as entradas não são escalares, obtendo-se o algoritmo LBG. Este algoritmo, para os casos em que a distribuição é conhecida, tem um desempenho muito próximo do algoritmo de Lloyd. Os principais passos do algoritmo LBG são os seguintes [5]:

1. Começar com um conjunto inicial de valores de reconstrução $\{Y_i^{(0)}\}_{i=1}^M$. Definir $k = 0$, $D^{(0)} = 0$ e seleccionar um limiar ϵ .
2. Procurar as regiões de quantização:

$$V_i^{(k)} = \{X : d(X, Y_i) < d(X, Y_j), \forall j \neq i\}, j = 1, 2, \dots, M - 1.$$

3. Calcular a distorção

$$D^{(k)} = \sum_{i=1}^M \int_{V_i^{(k)}} \|X - Y_i^{(k)}\|^2 f_X(X) dx.$$

4. Se $\frac{D^{(k)} - D^{(k-1)}}{D^{(k)}} < \epsilon$, parar, se não continuar.

5. Fazer $k = k + 1$. Determinar os novos valores de reconstrução $\{Y_i^{(k)}\}_{i=1}^M$, os quais são os centróides de $\{V_i^{(k-1)}\}$. Voltar ao passo 2.

Este algoritmo não é muito prático, nomeadamente, por causa dos integrais necessários ao cálculo da distorção. Geralmente, estes integrais são extremamente difíceis de calcular tornando este algoritmo de interesse mais académico. Com um maior interesse prático é a adaptação deste algoritmo para o caso em que se tem um conjunto de treino. Neste caso, o algoritmo assemelha-se muito ao algoritmo $k - means$ e os seus principais passos são [5]:

1. Começar com um conjunto inicial de valores de reconstrução $\{Y_i^{(0)}\}_{i=1}^M$ e um conjunto de vectores de treino $\{X_n\}_{n=1}^N$. Definir $k = 0$, $D^{(0)} = 0$ e seleccionar um limiar ϵ .
2. As regiões de quantização $\{V_i^{(k)}\}_{i=1}^M$ serão dadas por:

$$V_i^{(k)} = \{X_n : d(X_n, Y_i) < d(X_n, Y_j), \forall j \neq i\}, i = 1, 2, \dots, M.$$

3. Calcular a distorção média $D^{(k)}$ entre o conjunto de treino e os valores representativos de reconstrução.
4. Se $\frac{D^{(k)} - D^{(k-1)}}{D^{(k)}} < \epsilon$, parar, se não continuar.
5. Fazer $k = k + 1$. Determinar os novos valores de reconstrução $\{Y_i^{(k)}\}_{i=1}^M$, os quais serão os valores médios dos elementos de cada região $V_i^{(k-1)}$. Voltar ao passo 2.

O algoritmo LBG garante que a distorção de uma iteração para a outra não aumenta. No entanto, não é garantido que o processo conduza à solução óptima global, sendo a solução para o qual o algoritmo converge fortemente dependente das condições iniciais. Dada esta forte dependência das condições iniciais, a selecção do *codebook* inicial é extremamente importante.

Em [10], é descrita uma técnica designada por *splitting technique* para a inicialização do algoritmo. Nesta técnica, começa-se por considerar um quantizador

vectorial com um só *codevector*; por outras palavras, um *codebook* de tamanho um ou seja um quantizador de um só nível. Com um *codebook* de um só elemento, a região de quantização é todo o espaço de entrada e o *codevector* de saída é o valor médio do conjunto inteiro de treino. A partir deste *codevector*, o *codebook* inicial para um quantizador de dois níveis pode ser obtido por inclusão do *codevector* do quantizador de um nível e de um segundo *codevector*, obtido do primeiro pela adição de um vector de perturbação ε . Pode-se então usar o algoritmo LBG para se obter o quantizador vectorial de dois níveis. Após o algoritmo convergir, pode-se usar o *codebook* final do quantizador de dois níveis e dois outros vectores, obtidos por adição de ε aos outros dois vectores no *codebook* final do quantizador vectorial de dois níveis, como *codebook* inicial para um quantizador vectorial de quatro níveis. O algoritmo LBG pode ser então utilizado até este quantizador de quatro-níveis convergir. Deste modo, vai-se duplicando o número de níveis até se atingir o número desejado de níveis. Pela inclusão, em cada divisão, do *codebook* final do estágio anterior garante-se pois que cada *codebook*, depois da divisão, será pelo menos tão bom como o *codebook* anterior à divisão.

No passo 2, aquando do cálculo das regiões de quantização, assumiu-se que nenhuma das regiões de quantização estava vazia. Ora isto nem sempre é assim. Quando se associam entradas a *codevectors* iniciais, pode acontecer que não se associe nenhuma entrada a alguns *codevectors*. Isto é problemático porque para se actualizarem os *codevectors* é necessário considerar o valor médio dos vectores de entrada. Uma aproximação vulgar para resolver este problema, que é designado por problema da célula vazia, é remover qualquer *codevector* que não tenha nenhuma entrada associada e substituí-lo por um valor da região de quantização com mais pontos de saída. Isso pode ser feito seleccionando um ponto, aleatoriamente, da região com maior número de vectores de treino, ou que tenha associada a maior distorção.

Existem muitos outros métodos alternativos aos indicados, quer para a geração dos *codebooks* quer para a própria construção dos quantizadores vectoriais [8, 9]. O

algoritmo introduzido em [9] é uma alternativa para a geração do *codebook* inicial, mas é muito mais pesado computacionalmente.

Codificação Diferencial - Em muitas fontes as amostras obtidas $\{x_n\}$ não variam grandemente de uma amostra para a seguinte. Isto significa que tanto o intervalo dinâmico como a variância da sequência de diferenças $\{d_n = x_n - x_{n-1}\}$ são significativamente mais pequenos do que os da sequência de saída. Em vez de se codificarem os valores da sequência, dada a relação entre a variância da entrada do quantizador e o erro de quantização obtido, é habitual fazer-se a codificação das diferenças de uma amostra para a seguinte. As técnicas onde se transmite informação por codificação das diferenças, são chamadas técnicas de codificação diferencial [4, 5].

As fontes de imagens são, normalmente, altamente correlacionadas de amostra para amostra. Pode-se usar este facto para prever cada amostra baseada no seu passado. Assim, basta unicamente codificar e transmitir as diferenças entre a predição e os valores da amostra. Os esquemas de compressão de imagem baseados em codificação diferencial são construídos com base nesta premissa.

O sistema básico de codificação diferencial é conhecido como Modulação Diferencial por Codificação de Pulsos, DPCM (*Differential Pulse Code Modulation*).

Quando se estabelece um quantizador para uma determinada fonte, o tamanho do intervalo de quantização depende da variância da entrada. Se assumirmos que a entrada é uniformemente distribuída, a variância depende do intervalo dinâmico da entrada. Por outro lado, o comprimento do intervalo de quantização determina a quantidade de ruído de quantização introduzido durante o processo de quantização.

O intervalo dinâmico das diferenças entre amostras sucessivas é substancialmente mais pequeno do que o intervalo dinâmico das saídas da fonte. Por isto, caso se utilize este intervalo em vez de se usar os *pixels* originais, pode-se obter melhores resultados na compressão. Consegue-se então uma aceitável reprodução da sequência original a partir dos valores quantizados das diferenças.

Considerando uma sequência $\{x_n\}$, a sequência das diferenças $\{d_n\}$ é gerada pelo cálculo das diferenças $x_n - x_{n-1}$. Esta sequência de diferenças é quantizada

obtendo-se a sequência $\{\hat{d}_n\}$:

$$\hat{d}_n = \mathcal{Q}[d_n] = d_n + q_n,$$

onde q_n é o erro de quantização. No receptor, a sequência reconstruída \hat{x}_n é obtida adicionando-se \hat{d}_n aos anteriores valores reconstruídos \hat{x}_{n-1} :

$$\hat{x}_n = \hat{x}_{n-1} + \hat{d}_n.$$

Tanto o transmissor como o receptor iniciam com o mesmo valor x_0 , isto é, $\hat{x}_0 = x_0$. Deste modo, tem-se:

$$\begin{aligned} d_1 &= x_1 - x_0 \\ \hat{d}_1 &= \mathcal{Q}[d_1] = d_1 + q_1 \\ \hat{x}_1 &= x_0 + \hat{d}_1 = x_0 + d_1 + q_1 = x_1 + q_1 \\ d_2 &= x_2 - x_1 \\ \hat{d}_2 &= \mathcal{Q}[d_2] = d_2 + q_2 \\ \hat{x}_2 &= \hat{x}_1 + \hat{d}_2 = x_1 + q_1 + d_2 + q_2 = x_2 + q_1 + q_2. \end{aligned} \tag{2.4.1}$$

Continuando este processo obtém-se:

$$\hat{x}_n = x_n + \sum_{k=1}^n q_k.$$

Analisando esta equação verifica-se que o erro de quantização acumula-se durante o processo. Teoricamente, se este processo de cálculo do erro de quantização tiver média nula, a influência dos erros é minimizada. Na prática os erros de arredondamento de cálculo fazem com que os valores de reconstrução não estejam isentos de erro.

Observe-se que o codificador e o decodificador operam com diferente informação. O codificador gera a sequência das diferenças baseado nas amostras originais, enquanto que o decodificador utiliza uma versão distorcida do sinal original. Este

problema pode ser resolvido forçando-se quer o codificador quer o decodificador a usar a mesma informação durante as operações de diferenciação e reconstrução. A única informação disponível no receptor sobre a sequência $\{x_n\}$ é a sequência reconstruída \hat{x}_n . Como esta informação está também disponível no transmissor, pode-se usar uma versão modificada das operações de diferenciação de modo a usar os valores de reconstrução, isto é, pode-se considerar:

$$d_n = x_n - \hat{x}_{n-1}.$$

Então assumindo-se $\hat{x}_0 = x_0$, tem-se:

$$\begin{aligned} d_1 &= x_1 - x_0 \\ \hat{d}_1 &= \mathcal{Q}[d_1] = d_1 + q_1 \\ \hat{x}_1 &= x_0 + \hat{d}_1 = x_0 + d_1 + q_1 = x_1 + q_1 \\ d_2 &= x_2 - \hat{x}_1 \\ \hat{d}_2 &= \mathcal{Q}[d_2] = d_2 + q_2 \\ \hat{x}_2 &= \hat{x}_1 + \hat{d}_2 = \hat{x}_1 + d_2 + q_2 = x_2 + q_2, \end{aligned} \tag{2.4.2}$$

ou seja, temos para a n – *ésima* iteração:

$$\hat{x}_n = x_n + q_n.$$

Com esta alteração não haverá acumulação no ruído de quantização pois o ruído de quantização no n – *ésimo* valor reconstruído é o erro de quantização introduzido só pela quantização da n – *ésima* diferença. Por outro lado, o erro de quantização para a sequência das diferenças é substancialmente menor do que o erro de quantização para a sequência original.

Codificação por Sub-Banda- Cada uma das técnicas que foram analisadas anteriormente estava melhor adaptada para tipos de dados que tinham certas características particulares. Os esquemas baseados em quantização vectorial estão melhor

adaptados ao caso em que os blocos das saídas da fonte tenham um alto grau de agrupamento. A codificação diferencial é mais apropriada quando as diferenças de amostra para amostra forem pequenas. Se as saídas da fonte forem fortemente aleatórias, a melhor opção deverá ser a quantização escalar. Deste modo, se a fonte tiver características bem definidas, podemos escolher um destes esquemas de acordo com as suas características.

No entanto, muitas fontes exibem uma combinação de características o que torna mais difícil escolher o esquema de compressão apropriado. Deste modo, os esquemas de codificação por bandas são candidatos para este tipo de fontes, já que se podem combinar diferentes técnicas, de entre as anteriormente mencionadas, de modo a tirar partido da diversidade de características. Para isso, na codificação por sub-banda as saídas das fontes são decompostas nas suas partes constituintes. Cada uma destas partes constituintes do sinal original é codificada, utilizando-se um ou mais dos métodos anteriormente enunciados [4, 5, 12].

Assim, na codificação por bandas, após a entrada ter sido decomposta nos seus constituintes e para melhorar o desempenho da compressão, pode-se utilizar a técnica de codificação melhor adaptada a cada um destes constituintes. Igualmente, cada componente da fonte pode ter diferentes características de percepção. Por exemplo, o erro de quantização, sendo perceptível num dos componentes, pode ser aceitável num componente diferente e, deste modo, um quantizador mais grosseiro pode ser utilizado para codificar o componente que seja perceptivelmente menos importante.

Um método de decomposição de sinais, que nos últimos anos tem tido bastante aceitação, é o que recorre à decomposição em *wavelets*. Na representação por *wavelets* representamos um sinal em termos de funções, as quais estão localizadas quer no tempo quer na frequência [13, 14].

Codificação por Transformada - A codificação por transformada pode ser vista como uma extensão da técnica, atrás referida, de codificação por sub-bandas.

A ideia subjacente aos métodos baseados na codificação por transformada é a

de que se for considerada uma sequência de entrada a qual é transformada numa outra sequência, onde a maioria da informação está contida unicamente nuns poucos elementos, pode-se então codificar e transmitir somente esses elementos.

Usando-se uma transformada reversível, pode-se compactar muita da informação de uma sequência de saída de uma fonte em poucos elementos de uma sequência transformada e, desprezando os elementos da sequência transformada que não contenham muita informação, assim obter-se uma grande taxa de compressão. Esta é a ideia base por detrás da codificação por transformada [4, 5, 15, 16].

Pode-se analisar o processo de transformada em termos das alterações nas estatísticas entre a sequência original e a transformada. Obtém-se uma maior compactação se for usada uma transformada que descorrelacione a sequência de entrada. Assim, a correlação de amostra para amostra da sequência transformada deverá ser zero.

A codificação por transformada consiste em três passos:

- No primeiro, a sequência de dados $\{x_n\}$ é dividida em blocos de tamanho N . Cada bloco é mapeado numa sequência transformada $\{\theta_n\}$ usando-se uma função reversível. Os diferentes elementos de cada bloco da sequência transformada têm geralmente diferentes propriedades estatísticas.
- O segundo passo, consiste na quantização da sequência transformada. A estratégia de quantização depende de três factores principais: a taxa média de *bits* desejada, as estatísticas dos vários elementos da sequência transformada e o efeito da distorção dos coeficientes transformados na sequência reconstruída.
- Após estes dois passos, faz-se a codificação dos valores quantizados.

As transformadas a utilizar deverão ser necessariamente transformadas lineares e também deverão ser transformadas ortonormais. Uma transformada ortonormal tem a propriedade de que a inversa da sua matriz de transformação é a sua transposta, isto é:

$$\mathbf{B} = \mathbf{A}^{-1} = \mathbf{A}^T.$$

Considerando uma transformada linear bi-dimensional Θ que, para um bloco de imagem de tamanho $N \times N$ e considerando o i -ésimo pixel $X_{i,j}$ da imagem, é dada por:

$$\Theta_{k,l} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{i,j} a_{k,i} a_{l,j},$$

e $a_{i,j}$ é o (i,j) i -ésimo elemento da matriz, isto é $a_{i,j} = [\mathbf{A}]_{i,j}$. Então em notação matricial, teremos

$$\Theta = \mathbf{A} \mathbf{X} \mathbf{A}^T.$$

Deste modo, para uma transformada ortonormal, a transformada inversa será dada por:

$$\mathbf{X} = \mathbf{A}^T \Theta \mathbf{A}.$$

As transformadas ortonormais têm a propriedade de preservarem a energia, isto é, a soma dos quadrados da sequência transformada é igual à soma dos quadrados da sequência original. A eficácia de uma transformada depende de quanta compactação da energia é conseguida pela transformada. Uma maneira de medir a quantidade de energia compactada, obtida com uma determinada transformada ortonormal, consiste em obter a razão entre a média aritmética das variâncias dos coeficientes transformados e as suas médias geométricas. Esta razão é frequentemente referida como o ganho da codificação por transformada, GTC (*Gain of the Transform Codification*):

$$GTC = \frac{\frac{1}{N} \sum_{i=0}^{N-1} \sigma_i^2}{\left(\prod_{i=0}^{N-1} \sigma_i^2\right)^{\frac{1}{N}}}, \quad (2.4.3)$$

onde σ_i^2 é a variância do i -ésimo coeficiente θ_i [17].

A transformada que maximiza o ganho definido em (2.4.3) e que permite portanto obter maior compactação, é a transformada de Karhunen-Loeve, KLT (*Karhunen-Loeve Transform*). No entanto, esta transformada é dependente dos dados, o que não a torna prática em muitas circunstâncias. Deste modo, é habitual utilizar-se outras transformadas que, se bem que não tão eficientes, são independentes dos dados e portanto de uso mais geral.

A transformada discreta do co-seno, DCT (*Discret Cosine Transform*), é uma transformada similar à transformada discreta de Fourier, DFT (*Discret Fourier Transform*). A DCT permite uma compactação próxima à obtida com a KLT com a vantagem de não ser dependente dos dados [16, 18, 19, 20]. A matriz de transformação \mathbb{C} da DCT é obtida das funções em co-seno dadas por:

$$[\mathbb{C}]_{i,j} = \begin{cases} \sqrt{\frac{1}{N}} \cos \frac{(2j+1)i\pi}{2N} & , i = 0 & , j = 0, 1, \dots, N-1 \\ \sqrt{\frac{2}{N}} \cos \frac{(2j+1)i\pi}{2N} & , i = 1, 2, \dots, N-1 & , j = 0, 1, \dots, N-1. \end{cases}$$

Outras transformadas podem ser utilizadas tais como a transformada discreta do Seno, DST (*Discret Sine Transform*) ou a Transformada de Walsh-Hadamard, DWHT (*Discret Walsh-Hadamard Transform*). A DST é a transformada complementar da DCT. Quando o coeficiente de correlação entre amostras é grande, a DCT permite obter um desempenho, em termos de compactação, mais próximo do obtido com a KLT, sendo por isso, habitualmente, utilizada na compressão de imagem. Por seu lado, a DST permite obter uma performance próxima da KLT quando este coeficiente é pequeno, o que a torna apropriada para aplicações de compressão de áudio. Devido a esta propriedade, é frequente usar-se esta complementaridade em aplicações de compressão de imagem e de áudio.

A transformada DWHT é bastante simples de implementar e computacionalmente leve, mas a compactação de energia é menor do que a conseguida com a DCT. Por isso, é preferível usar esta última sempre que a potência computacional disponível o permita.

A DCT está integrada em normas internacionais para compressão como é o caso, entre outras, da JPEG, MPEG e H.261.

Compressão Fractal - Uma abordagem bastante diferente para a compressão de imagem é baseada nas propriedades da auto-semelhança, a aproximação por codificação fractal. Nesta técnica, o que se transmite ou armazena não são as amostras da saída da fonte, mas sim o método para sintetizar a saída [21].

2.5 Compressão de Vídeo

Na compressão de vídeo, ou sequências de imagens, utilizam-se basicamente as mesmas técnicas usadas na compressão de imagens isoladas, adaptadas para se tirar partido da elevada correlação temporal existente entre imagens consecutivas.

Distinguem-se dois grupos básicos na adaptação destas técnicas para a compressão de vídeo: a sequência de imagens pode ser encarada como um sinal tridimensional $S(x, y, t)$, dando origem às técnicas designadas por sistemas de compressão $3D$, ou então, pode-se fazer um processamento, separado relativamente ao tempo e ao espaço, originando os chamados sistemas de compressão híbrida.

Muitas das normas internacionais para compressão de vídeo, tais como as normas H.261, H.263, MPEG-1, MPEG-2, têm sido desenvolvidas com base em sistemas de compressão híbridos.

Nestes sistemas de compressão híbridos podem-se considerar duas fases principais e interligadas: a compressão espacial, que basicamente é idêntica à compressão feita para imagens isoladas e a compressão temporal. Na compressão temporal, procura-se extrair informação relativa ao movimento de modo a proceder-se à respectiva compensação. Esta informação relativa ao movimento, pode ser transmitida directamente ao receptor juntamente com a imagem codificada. Alternativamente, essa informação pode ser determinada, recursivamente no emissor e no receptor, com base apenas nos *pixels* transmitidos.

Os movimentos que se podem encontrar nas sequências de imagem podem ser movimentos globais ou movimentos locais. Os movimentos globais são originados pela câmara correspondendo, normalmente, a deslocções horizontais ou verticais paralelas ao plano da câmara ou a movimentos de *zoom*. Estes movimentos globais podem ser modelizados por funções polinomiais. Pelo contrário os movimentos locais, que são devidos à deslocação de objectos dentro do campo da imagem, são, normalmente, mais difíceis de modelizar. Estes movimentos locais podem também ser rígidos, os objectos não alteram a sua forma durante o movimento, ou não-rígidos,

os objectos podem alterar a forma durante o movimento.

Para se fazer a estimação do movimento utilizam-se, normalmente, técnicas de emparelhamento de blocos. Nestas técnicas, determina-se qual o bloco D_k da imagem que mais se aproxima de acordo com uma medida de erro d , considerando movimentos de translacção, a uma região B_i na imagem a codificar. Obtém-se assim vectores de movimento.

2.5.1 Normas para a Compressão de Vídeo

Com o desenvolvimento de sistemas para transmissão de sequências de imagens, tornou-se necessário o estabelecimento de normas de compressão. A ITU-T, *International Telecommunication Union - Telecommunication*, e a ISO, *International Standard Organization*, foram as principais entidades responsáveis pelo estabelecimento dessas normas. Assim estabeleceram-se, entre outras, as normas ITU-T H.261 e H.263, MPEG-1, MPEG-2, MPEG-4, para a compressão/descompressão de vídeo. Cada uma destas técnicas têm características específicas diferentes e passaremos a descrevê-las sumariamente [18, 22, 23]. Apesar de não serem especificamente normas para a compressão de vídeo, dada a importância que têm neste contexto, referiremos também as normas MPEG-7 e MPEG-21.

Recomendação ITU-T H.261 - A recomendação H.261 é uma norma de compressão híbrida, baseada em DCT, com compensação de movimento. Tem dois modos de funcionamento: o modo intra-imagem, onde a imagem é codificada de forma independente e enviada para o receptor e o modo inter-imagem onde se codifica o erro de predição que se envia para o receptor.

O sistema H.261 destina-se a canais de transmissão com taxas de transmissão múltiplas de 64 kbps, $p \times 64$ kbps, com $1 \leq p \leq 30$. As taxas mais baixas, $p = 1$ ou $p = 2$, são dirigidas para aplicações de vídeo-telefone, enquanto as taxas superiores destinam-se a sistemas de vídeo-conferência ou outras aplicações [18, 19, 20, 24].

Recomendação ITU-T H.263 - Com o objectivo de normalizar a utilização de

vídeo-telefones, utilizando-se a rede telefónica analógica, a ITU estabeleceu a recomendação H.263. Esta recomendação foi baseada na H.261 e está vocacionada para canais de comunicação de baixo débito [23]. Globalmente, as principais diferenças em relação à H.261 prendem-se com melhorias nos algoritmos de compensação de movimento, possibilidade de se poderem usar imagens interpoladas, e a opção de poder utilizar-se codificação aritmética. O algoritmo de compressão da H.263 é um algoritmo de compressão híbrido, também baseado em DCT, com predição temporal. Opcionalmente, pode ser utilizada, no codificador, compensação de movimento.

Recomendação MPEG-1 (ISO/IEC 11172-2) - A metodologia utilizada no desenvolvimento da recomendação H.261 foi também adaptada para a recomendação MPEG-1 (*Motion Picture Expert Group - Phase 1*), a qual se destina à codificação de vídeo para canais com capacidade de 1,5 Mbps e foi, inicialmente, criada para a compressão de sequências audiovisuais e com a finalidade de gravação e reprodução.

Recomendação MPEG-2 (ITU-T H.262 — ISO/IEC 13818-2) - Esta recomendação estabelece normas para compressão de vídeo destinadas a aplicações como, por exemplo, televisão de alta definição, televisão inter-activa, etc., sendo as taxas de transmissão da ordem dos 10 Mbps ou superiores.

Estas três recomendações partilham de características comuns já que o MPEG-1 e o MPEG-2 foram desenvolvidos com base na H.261. Os algoritmos de compressão, em qualquer destas três normas, assentam num sistema híbrido DCT, com compensação de movimento e codificação em modo intra e inter-imagem. Também, em qualquer destes três sistemas, existem imagens Intra (I) e imagens obtidas por predição Inter (P). Nos MPEG existem, ainda, imagens obtidas por interpolação através do recurso a predição bi-direccional (B) [18, 19, 22, 24]. Na figura 2.2, exemplifica-se uma sequência de imagens MPEG.

Estas recomendações MPEG são recomendações genéricas que, por serem independentes de qualquer aplicação em particular, são passíveis de serem aplicadas a vários tipos de aplicações. Estas recomendações genéricas constituem um conjunto de facilidades, que podem ser ou não implementadas numa aplicação específica.

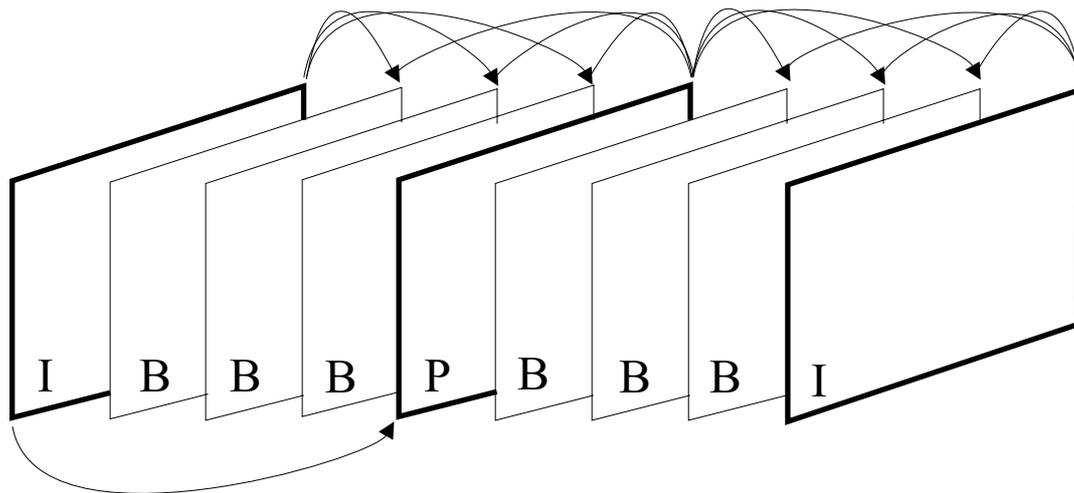


Figura 2.2: Sequência de imagens MPEG

Recomendação MPEG-4 - Esta recomendação, originada pela ISO/SG15-/W611, surge com o objectivo de normalizar aplicações nas áreas da multimedia, do vídeo-telefone, das comunicações móveis audiovisuais, do correio electrónico em vídeo, etc.. É dado particular ênfase ao acesso e manipulação da informação audiovisual. Pretende-se que seja um ambiente flexível, constituído por um conjunto de ferramentas, que suporte várias funcionalidades e esteja aberto à introdução de outras funcionalidades [18, 19, 22].

Assim, com o MPEG-4 pretende-se definir um esquema de codificação genérico para transmissão de vídeo a muito baixas cadências, VLBR (*Very Low Bit Rate*), isto é, abaixo dos 64 Kbps. Posteriormente, houve reformulações dos objectivos iniciais, nomeadamente, ficou previsto o suporte de novas funcionalidades, tais como:

- Acesso e manipulação de dados audiovisuais digitais;
- Manipulação e edição baseada no conteúdo;
- Ferramentas de acesso a informação multimedia, *hiper-links*, indexação, *uploading* e *downloading*;
- Variações de escalas baseadas no conteúdo;
- Codificação de várias fontes de informação;

- Combinação de cenas naturais com cenas artificiais;
- Acesso aleatório à representação codificada a muito baixas taxas de transmissão;
- Robustez a erros de transmissão.

Recomendação MPEG-7 e MPEG-21 - Estas duas normas, que se encontram actualmente em fase de especificação, não são normas para a compressão de vídeo mas sim para a definição de *frameworks* para o multimedia.

Na norma MPEG-7 introduz-se o conceito de *metadados*, isto é, informação digital que descreve o conteúdo de outros dados digitais. Esta informação é também referida como *Content Description*. Com esta norma, pretende-se definir um conjunto completo de ferramentas que permita criar descrições de conteúdo audiovisual e permita o acesso eficiente a informação multimedia - pesquisa, filtragem personalizada, identificação de conteúdos eficiente e ainda a utilização de diferentes atributos dos conteúdos, textura, cor, forma, texto, sons, etc.. Para isso, utilizam-se descritores que acompanham a informação multimedia e que permitem a pesquisa e o acesso a informação multimedia através de atributos múltiplos. Pode-se usar então características visuais tais como cor, forma, som, texto, de conteúdo, etc [18, 22, 25].

Com a norma MPEG-21 procura-se criar uma estrutura completa para a gestão e utilização de bens digitais, incluindo toda a infraestrutura de suporte para transações comerciais e gestão de direitos. Para isso, cria-se um *Multimedia Framework*, que passa por construir uma plataforma para a gestão eficiente de elementos digitais da cadeia de conteúdos audiovisuais, em ambientes heterogéneos. Isto inclui conteúdos vídeo, audio, texto, gráficos, passando por direitos de autor, direitos de utilização, segurança, até aos próprios dispositivos hardware. Assim, facilita-se o acesso aos conteúdos multimedia em ambientes heterogéneos pois suporta uma grande diversidade de formatos - sistemas proprietários, formatos *standards* e assenta na necessidade de normas abertas, em particular, para o *middleware* das *Set Top Boxes* e para os sistemas de desenvolvimento de aplicações interactivas, máxima utilização

através de várias plataformas. Também dará suporte a protocolos e normas da Internet, XML, IP, como papel preponderante para promover a inter-operabilidade [18, 25].

2.5.2 Compressão para Transmissão de Vídeo a Muito Baixa Cadência

O conceito de compressão de vídeo, a muito baixas cadências, refere-se, normalmente, à compressão de sequências de dados que não excedam 64 Kbps, na parte dos dados correspondentes a informação visual [26, 27]. A informação a comprimir corresponde, para este tipo de aplicação, tipicamente a imagens de pequenas dimensões, normalmente, 144×176 *pixels*, QCIF¹. Isto corresponde aproximadamente a 25 *K-pixels* em cada imagem de luminância e um quarto deste valor na crominância. A taxa de amostragem das imagens situa-se, tipicamente, à volta de 5-10 *frames/segundo*. Deste modo, dependendo das aplicações, as taxas de transmissão preconizadas vão desde os 4.8 Kbps aos 64 Kbps [26, 27].

Para além das normas para compressão de vídeo referidas na secção anterior, têm sido desenvolvidas outras técnicas tendo em vista a transmissão de vídeo em canais de muito baixo débito. Sem pretendermos fazer uma descrição exaustiva de todas as técnicas que têm surgido com esta finalidade, referiremos de seguida as que nos parecem mais promissoras, tendo em conta os objectivos para que foram desenvolvidas.

Podem-se agrupar estas técnicas em dois grandes grupos: a compressão de vídeo baseada no sinal e a compressão de vídeo baseada em modelos.

No primeiro destes grupos, de modo a obter-se compressão, procura-se reduzir a redundância no sinal de entrada. Os métodos mais vulgarmente utilizados no domínio temporal empregam estimação/compensação de movimento. Todavia, para

¹CIF (*Common Intermediate Format*), é um tamanho de imagem com 288×352 *pixels* que é uma versão sub-amostrada 2:1 da Recomendação *CCIR Recommendation 601 for Standard Display Format* e QCIF, por sua vez, é uma versão sub-amostrada daquela com 144×176 *pixels* [6].

aplicações de codificação de vídeo a muito baixa cadência, estes métodos baseados na codificação do sinal parecem ter atingido o limite das possibilidades.

Os sistemas de compressão de vídeo baseados no sinal podem, por sua vez, ser classificados em dois tipos: compressão espaço/temporal e compressão espaço-temporal, figura 2.3. O primeiro tipo é normalmente designado por compressão híbrida, enquanto que o segundo tipo é referenciado por compressão *3D*. Para o primeiro tipo, tem sido desenvolvida investigação numa perspectiva de estimação/com-pensação de movimento para multi-*frames* nomeadamente, no âmbito de pesquisa para o MPEG-4 [22].

Para a compressão *3D* foram desenvolvidas técnicas baseadas em DCT, transformadas *3D – DCT* [15], onde a DCT é aplicada simultaneamente no espaço e no tempo; técnicas baseadas em fractais aplicados simultaneamente no tempo e no espaço, *Fractals 3D* [28], ou ainda codificação *3D* com *wavelets* ou *3D Sub-Band* [28]. Estas técnicas têm-se mostrado pouco adequadas a sequências com muito movimento. Para obviar este problema, têm-se desenvolvido técnicas de compensação de movimento as quais, normalmente, resultam no aumento da complexidade do algoritmo de compressão. Isto torna estas técnicas pouco adequadas para processamento em tempo real ou quase real.

Outra possível solução para ultrapassar as limitações da compressão baseada no sinal, é a chamada compressão baseada em modelos. Caso se consiga construir um modelo tri-dimensional, com base em cenas *2D* da sequência de imagens, podem-se analisar e sintetizar as imagens com base neste modelo. Assim, obtém-se uma grande redução na quantidade de informação necessária para se descrever a imagem. Existem duas abordagens principais para a compressão baseada em modelos: compressão da imagem orientada a objectos e compressão baseada em semântica [29].

A ideia básica subjacente à compressão de imagens orientada a objectos [29], consiste em extrair os objectos na sequência de vídeo e transmitir-se a forma, movimento e informação de textura para cada objecto, figura 2.4. O conteúdo de cada

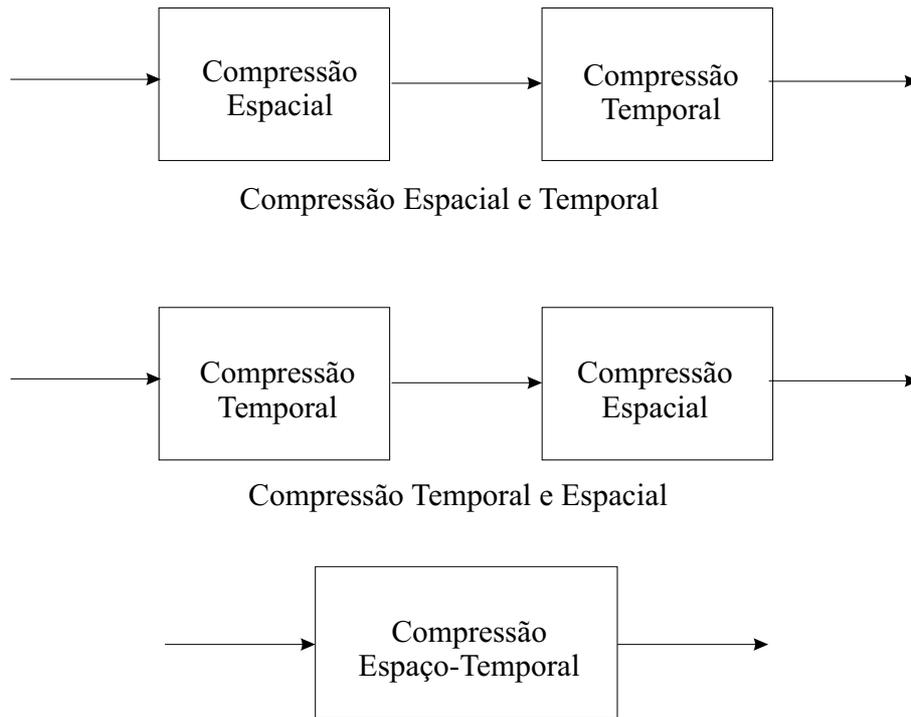


Figura 2.3: Compressão baseada no sinal

imagem é classificado como pertencente ao *background*, objecto de acordo com o modelo, MCO (*Model-Compliance Object*) e objecto não abrangido pelo modelo, MFO (*Model-Failure Object*). Os MCO são codificados com o seu movimento e forma, enquanto que os MFO são codificados com a sua cor e forma. O *background* não necessita de ser codificado. Para aplicações de vídeo-conferência em que, habitualmente, as imagens são de cabeça e tronco humanos esta aproximação, normalmente, funciona bem. A cabeça e o tronco são normalmente classificados como MCO, enquanto que os detalhes na cara, tais como os olhos e os lábios, são classificados como MFO. Esta aproximação pode funcionar com uma grande variedade de padrões na imagem. No entanto, os custos computacionais necessários à análise são normalmente bastante elevados.

No caso da compressão baseada na semântica, a ideia base é a de, no transmissor, se extrair os parâmetros do modelo e, no receptor, sintetizar a imagem. Estas operações são baseadas em modelos 3D explícitos da cena, figura 2.5.

Estes esquemas assentam na existência de um modelo paramétrico da saída da

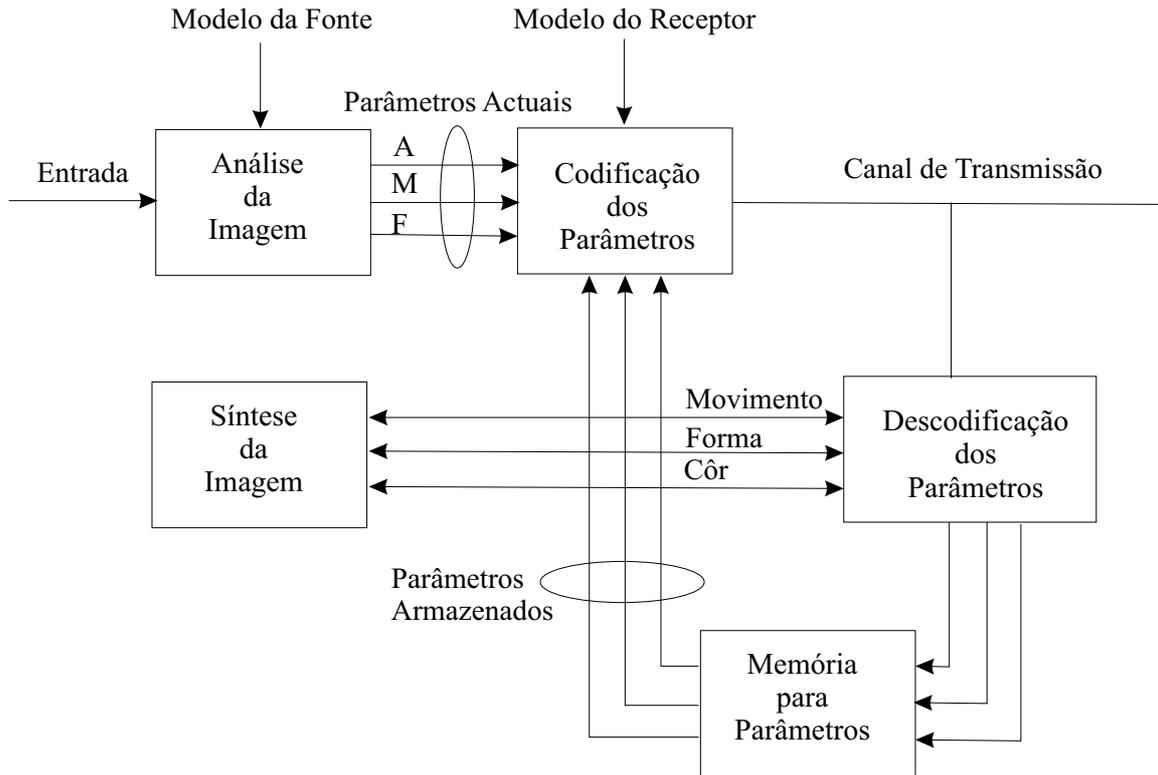


Figura 2.4: Codificação baseada em objectos

fonte. Quando tal modelo existe, o transmissor analisa a saída da fonte e extrai os parâmetros do modelo. Estes parâmetros são transmitidos ao receptor onde, com base nos parâmetros transmitidos, o modelo é usado para sintetizar uma aproximação à saída da fonte. A diferença entre esta abordagem e as técnicas discutidas anteriormente é que, aqui, não se transmite uma representação directa das amostras da saída da fonte, mas sim o modo de proceder para regenerar essas amostras [30].

Esta abordagem não tem mostrado sucesso quando usada para compressão genérica de imagens. No entanto, em aplicações de vídeo-conferência, onde as imagens são basicamente constituídas por imagens constituídas por cabeça e tronco, tem havido algum progresso e têm-se obtido alguns resultados aceitáveis.

Existem trabalhos [29, 30], sobre a aplicação desta abordagem à compressão de vídeo para transmissão a muito baixas cadências. Normalmente, o modelo utilizado é o *modelo em arame* da cabeça, como o mostrado na figura 2.5. Neste caso, são transmitidos unicamente os parâmetros do modelo facial. No receptor são utilizadas técnicas de computação gráfica para reconstruir uma expressão facial aproximada.

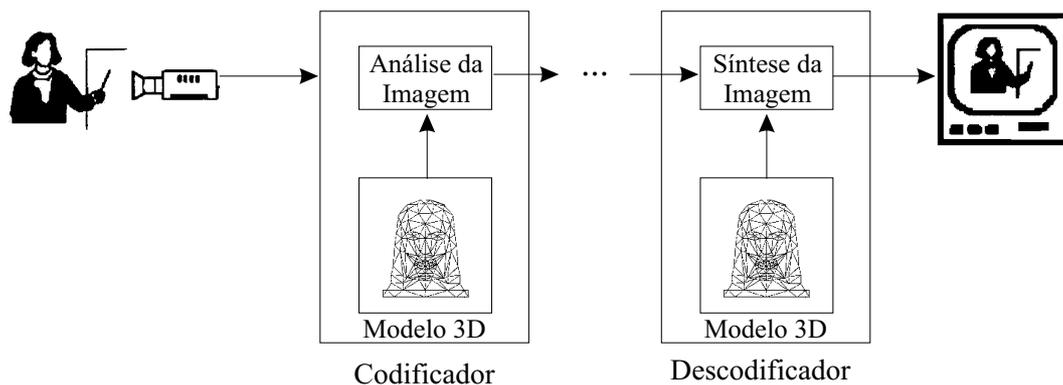


Figura 2.5: Compressão baseada na semântica

Como se transmitem somente muitos poucos parâmetros referentes ao movimento numa grelha de controlo, tais como acção de músculos, este esquema é apropriado para movimentos lentos na face humana. Por outro lado, a maneira de se analisar e extrair os parâmetros numa sequência é outro problema. Para além disto, a compressão baseada na semântica funciona somente para padrões de imagem específicos. Logo, se um padrão inesperado, isto é, que não esteja abrangido pelo *modelo em arame*, aparece na imagem o modelo cresce em complexidade e na consequente taxa de *bits*.

2.6 Sumário

Neste capítulo, fez-se uma abordagem das implicações de alguns resultados da Teoria da Informação e da sua aplicação à compressão de imagens. Analisaram-se estes resultados numa perspectiva da sua influência no desenho e modelização de algoritmos de compressão de imagens, tendo em atenção, por exemplo, factores como entropia e limitações do sistema visual humano.

Também foram analisadas as principais técnicas utilizadas para a compressão de imagens, salientando-se as vantagens e desvantagens de cada uma delas, bem como o tipo de fonte para as quais estarão, à partida, melhor adaptadas. Analisou-se ainda a utilização destas técnicas na compressão de sequências de imagens e as principais

normas internacionais adoptadas para a compressão de vídeo. Por último, abordou-se a compressão de vídeo tendo como objectivo a sua transmissão a taxas muito baixas.

Capítulo 3

Detecção de Movimento em Sequências de Imagens Submarinas

3.1 Introdução

A possibilidade de se visualizar à superfície imagens submarinas obtidas directamente a partir de um submarino *robot*, a operar autonomamente no fundo do mar, é bastante interessante pelas perspectivas que abre à exploração oceanográfica. Isto é interessante quer do ponto de vista científico, quer do ponto de vista económico no que respeita à exploração das reservas submarinas.

Neste caso concreto, interessa transmitir para a superfície sequências de imagens capturadas por um veículo submarino autónomo, cuja missão é explorar zonas com actividade hidrotermal relevante. Este veículo opera autonomamente, isto é, sem qualquer ligação física com a superfície. Deste modo, todas as comunicações do veículo com a superfície terão de ser feitas através do canal acústico submarino, o qual suporta apenas taxas de transmissão muito baixas ($20 - 30Kbps$).

Nas imagens de interesse, o movimento das bolhas é bastante rápido e o contraste muito baixo. Por conseguinte, as técnicas tradicionais de segmentação de imagem não são directamente aplicáveis.

Nos algoritmos propostos, para se segmentarem as imagens, tira-se partido do facto das bolhas gasosas terem um movimento preferencialmente vertical. Assim,

implementou-se um detector que privilegia a detecção de movimento segundo direcções preferenciais. Deste modo, conseguir-se-ão segmentar as imagens em zonas onde não exista movimento, correspondentes ao fundo submarino (*background*), e zonas constituídas pelas regiões onde exista movimento, correspondentes às bolhas gasosas em movimento.

Existem filtros que podem ser orientados preferencialmente para determinadas direcções. Os filtros de Gabor são filtros deste tipo que são usados nesta abordagem para se proceder à segmentação das imagens.

Os filtros de Gabor, quando aplicados a uma sequência de imagens, permitem determinar a velocidade e a direcção de *pixels* da imagem. No entanto em imagens com muita textura, como são as utilizadas, a sua aplicabilidade é difícil, pois podem existir movimentos espúrios que, no caso particular das imagens utilizadas, são, essencialmente, devidos à presença de algas. Estas algas, estando em constante movimento devido às correntes marítimas, provocam ruído na determinação dos movimentos que interessa isolar e que são os movimentos das bolhas gasosas.

Devido a estes factores, desenvolveu-se uma alteração ao modo como tradicionalmente os filtros de Gabor são utilizados na determinação de movimento em sequências de vídeo. Esta alteração, que será pormenorizadamente explicada na secção 3.3.4, permite, de uma forma fácil e precisa, seleccionar nas imagens da sequência somente os blocos que contenham movimento referente a bolhas gasosas. Esta selecção de blocos constituirá uma primeira fase do algoritmo de compressão.

3.2 Transmissão e Reconstrução das Imagens

Devido ao facto dos movimentos nestas imagens serem essencialmente verticais e no sentido de baixo para cima e tendo-se um *background* relativamente estático, faz sentido só transmitir para a superfície os blocos das imagens onde foi detectado movimento.

Uma vez recebidos à superfície os blocos, estes deverão ser processados de modo

a permitirem a reconstrução da sequência de imagens. Para o efeito será necessário transmitir-se, para além dos blocos das imagens onde foi detectado movimento, os blocos correspondentes à parte estática da imagem, ou seja, ao *background*. No entanto, só será necessário transmitir estes blocos, correspondentes ao *background*, ocasionalmente e de acordo com uma política de refrescamento da imagem. Esta política poderá ser implementada de acordo com alguma estratégia pré-definida. Esta estratégia pode passar, por exemplo, por testar no codificador a qualidade das imagens reconstruídas e se esta estiver abaixo de algum limiar pré-definido retransmitir os blocos correspondentes a um novo *background*. Estes blocos constituirão como que uma imagem de base, sobre a qual serão inseridos os blocos correspondentes a cada uma das imagens constituintes da sequência obtendo-se, deste modo, a sequência reconstruída.

Esta imagem de base poderá ser qualquer uma do grupo de imagens a transmitir de cada vez. No entanto, deverá ser escolhida de preferência a imagem equidistante, dentro de cada grupo de imagens, de modo a minorar eventuais alterações que possam existir no *background* entre a primeira e a última imagem do grupo.

Em vez de se utilizar como imagem de base uma qualquer imagem do grupo de imagens a codificar, pode-se optar por utilizar uma outra imagem, que designaremos por *imagem do mundo*, e que consiste numa imagem que será obtida da consideração de todas as imagens do grupo a comprimir. Para isso, consideram-se nesse grupo os blocos que se mantiveram, a menos de um limiar, mais estáveis ao longo da sequência. A imagem base será então esta imagem do mundo e que foi construída com o conjunto destes blocos. Transmite-se esta imagem e faz-se a reconstituição das restantes imagens com base nela.

De acordo com simulações efectuadas, e que são analisadas na secção 5.6, utilizando as imagens submarinas obtidas no cenário de interesse é de esperar que, se o veículo submarino se mantiver relativamente estável durante a filmagem, a taxa de refrescamento da imagem de base será bastante baixa. Assim, o *overhead* intrínseco ao método e decorrente da necessidade de se transmitir os blocos correspondentes às

zonas sem movimento, *background*, dilui-se com o aumento do número de imagens transmitidas.

Outro aspecto a considerar, na transmissão desta imagem de base, é a taxa de compressão desejada para a mesma. Esta imagem pode ser mais ou menos comprimida obtendo-se conseqüentemente uma pior ou melhor reconstituição da mesma. Também, de acordo com as simulações efectuadas e considerando o que atrás foi referido sobre a taxa de refrescamento da imagem de base, a qual é relativamente baixa, optou-se por transmitir esta imagem moderadamente comprimida em modo intra-imagem recorrendo-se para tal à compressão por Quantização Vectorial clássica ou por DCT.

3.3 Detecção dos Blocos de Imagens com Actividade Hidrotermal

No caso das imagens que se pretendem transmitir para a superfície tem-se, como já referido anteriormente, dificuldades acrescidas pelo facto de nestas imagens o contraste ser bastante baixo e a velocidade de deslocamento das bolhas gasosas nas imagens ser bastante alta. Assim, no método a utilizar para detectar o movimento das bolhas e permitir isolar para segmentação essas zonas, tem que se levar em linha de conta estes dois factores.

3.3.1 Percepção do Movimento em Sequências de Imagens

Qualquer observador depende da percepção visual do movimento para perceber o ambiente que o rodeia. A percepção visual do movimento não depende de uma prévia interpretação ou reconhecimento da forma. Desse modo, também não há uma dependência relativamente à informação do movimento, isto é, relativamente a mudanças de intensidade ao longo do tempo dentro do campo visual. Uma superfície sem textura movendo-se muito suavemente, conduz a uma sequência em que muitas

regiões locais não mudam com o tempo. Pelo contrário, numa região muito texturizada, por exemplo imagens com árvores e relva, essa informação sobre o movimento está presente em todo o campo visual.

Algumas das mais importantes funções para a percepção do movimento são [31]:

- 1) Funcionar como um sistema prévio de advertência;
- 2) Permitir ao observador seguir a localização de objectos em movimento e recuperar a sua estrutura tri-dimensional;
- 3) Ajudar o observador a determinar o seu próprio movimento (*egomotion*) relativamente ao meio;
- 4) Ajudar o observador a dividir o campo visual em segmentos perceptíveis, por exemplo, em zonas com movimento *versus* zonas estacionárias ou movimento rígido *versus* movimento não rígido.

É geralmente aceite que a análise visual do movimento se processa em duas etapas [31]. A primeira etapa, corresponde à obtenção de informação sobre o movimento bi-dimensional: direcção do movimento, velocidade e mudanças de posição na sequência de imagens. Na segunda etapa faz-se a interpretação dessa informação e corresponde à segmentação da imagem.

Quando se vê um filme, vemos uma sequência de imagens nas quais os objectos aparecem numa sequência de posições. Embora cada *frame* corresponda a um instante no tempo, o filme dá uma convincente impressão de movimento. De certa forma o sistema visual interpreta a sucessão das imagens conseguindo a percepção de cenas com movimento contínuo. Este fenómeno representa uma forma de movimento aparente. O sistema visual humano, supostamente, compara pontos correspondentes nos sucessivos *frames* e calcula e infere uma dada velocidade. Esta correspondência é calculada com base na distância percorrida durante o intervalo entre *frames* e deste modo o movimento aparente é apercebido [32]. Muita pesquisa tem sido efectuada para se estabelecer esta correspondência permanecendo contudo como um problema fundamental a ser resolvido.

Num sistema típico de comparações o sistema visual deverá fazer comparações em

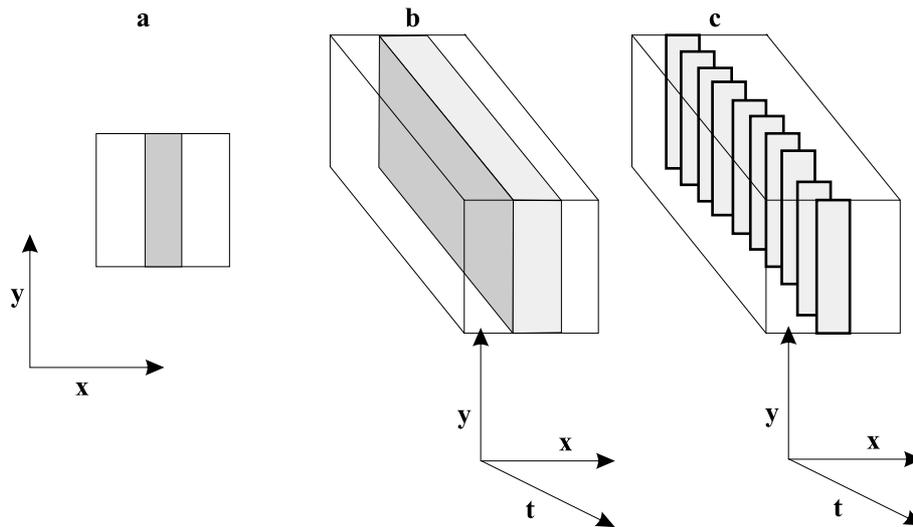


Figura 3.1: Deslocamento tri-dimensional

- a) Barra vertical a deslocar-se
- b) Diagrama tri-dimensional do deslocamento da barra
- c) *Filme* da barra em movimento

grandes regiões da imagem. Para isso, executará comparações em janelas sucessivas de *frame* para *frame* para encontrar uma correspondência ideal na *frame* seguinte. Os modelos de comparação [32], são estabelecidos para fazer previsões sobre os estímulos presentes em sequências de *frames*.

A maioria dos modelos de comparação globais são formulados, unicamente, para lidar com movimentos globais simples. Deste modo, não podem ser facilmente aplicados a situações nas quais vários movimentos são visíveis em diferentes posições.

Os estímulos de movimento podem ser graficamente representados como ocupando um espaço tri-dimensional, no qual x e y são as duas dimensões espaciais e t é a dimensão temporal. Consideremos uma barra vertical deslocando-se continuamente para a direita, como é mostrado na figura 3.1-a) [32]. Na figura 3.1-b) é representado o diagrama tri-dimensional representativo do deslocamento dessa barra no tempo. Se esse movimento contínuo for amostrado em intervalos discretos, o resultado é o representado na figura 3.1-c). Essa amostragem poderá ser representada como um filme da barra em movimento. Se a amostragem feita para a obtenção dos *frames* individuais, representados na figura 3.1-c), for suficientemente frequente no tempo os dois estímulos serão vistos como idênticos.

Pode-se questionar como aplicar as noções clássicas de amostragem e *aliasing* ao caso da amostragem tri-dimensional no espaço e no tempo, considerando as propriedades da filtragem espaço-temporal do sistema visual humano. Uma imagem com movimento contínuo tem um espectro de Fourier tri-dimensional em $f_x - f_y - f_t$ e a versão amostrada tem um diferente espectro. As diferenças entre o espectro do contínuo e o das imagens amostradas são chamadas artefactos de amostragem. São estes artefactos que permitem a um observador distinguir entre o contínuo e o amostrado. Um objectivo principal é o de assegurar que os artefactos, devidos à amostragem, tenham tão baixo contraste que sejam invisíveis ao observador humano. Para isso, não é necessário remover os componentes dos artefactos na totalidade mas simplesmente prevenir que atinjam o limiar de visibilidade.

Quando a amostragem temporal é grosseira o movimento tende a parecer deslocado, se bem que continue visível. Isto é, para provocar a impressão de movimento, o estímulo amostrado deverá ser indistinguível do contínuo. A análise de frequência espaço-temporal ajuda a perceber bem isto, porque os estímulos contínuos ou amostrados partilham uma grande quantidade de energia espaço-temporal, apesar de não a partilharem na totalidade. Pode-se esperar então que os dois estímulos pareçam similares, de modo a que os mecanismos visuais respondam a esta energia partilhada.

Se o movimento for amostrado mais frequentemente no tempo, a aproximação ao movimento contínuo é melhorada. Neste caso, os artefactos terão menos energia, especialmente na gama de frequências visíveis ao olho humano.

Se a amostragem for feita com uma frequência suficiente, isto claramente fará com que os artefactos tenham tão pouca energia, na gama de frequências espaciais e temporais visíveis, que se tornarão claramente invisíveis. Isto acontece porque a estrutura espaço-temporal dos artefactos é esbatida ao nível da invisibilidade da resposta temporal e espacial do olho. A este nível os estímulos contínuos e amostrados tornam-se perfeitamente indistinguíveis.

Para que o movimento pareça similar não é necessário que o estímulo amostrado seja idêntico ao contínuo. O mecanismo de movimento, o qual responde a baixas

frequências espaciais e temporais, dará a mesma resposta aos dois estímulos, apesar de os mecanismos de sensibilidade darem diferentes respostas para altas frequências. Deste modo, o movimento pode ser percebido em sinais amostrados ou contínuos, desde que tenham suficiente energia na orientação espaço-temporal apropriada.

O problema da detecção de movimento é então o da detecção da orientação espaço-temporal. Uma maneira de detectar orientações em localizações espaciais é através do uso de detectores de barras ou de detectores de *edges*. É conhecido que células simples do córtex visual actuam como filtros lineares. Os seus perfis de recepção representam funções pesadas, quer com pesos negativos quer com pesos positivos, as quais têm uma resposta impulsiva espacial similares às de um sistema linear.

Caso se consiga construir um sistema com uma resposta impulsional espaço-temporal análoga a uma dessas células de resposta impulsiva espacial, conseguir-se-à determinar o movimento no tempo. A resposta impulsiva espaço-temporal desse sistema será orientada no espaço e no tempo.

Se as respostas impulsiva espacial e temporal forem combinadas de uma maneira simples, o resultado é um impulso espaço-temporal separável. Se a resposta impulsiva espacial for $H_s(x)$, função de x , e a resposta impulsiva temporal for $H_t(t)$, função de t , então a resposta impulsiva espaço-temporal será $H_{st}(x, t) = H_s(x) * H_t(t)$. Respostas impulsivas espaço-temporais separáveis são fáceis de construir. Se as entradas forem obtidas de um conjunto de posições espacialmente distribuídas, pesadas por uma resposta impulsiva espacial e enviadas então para a saída através de um filtro temporal, a resposta impulsiva espaço-temporal continuará separável. Do mesmo modo, se as saídas de um grande número de receptores forem todas filtradas temporalmente da mesma maneira e as saídas filtradas forem combinadas com uma função espacial pesada, a resposta continuará separável. Separabilidade é frequentemente observada nos estágios iniciais do processamento visual no córtex.

3.3.2 Filtros Direccionais Sensíveis ao Movimento

Nas imagens de interesse o movimento predominante, correspondente às bolhas gasosas, terá uma direcção vertical ou quase vertical e sentido ascendente. Todo o outro tipo de movimento poderá ser então considerado espúrio e sem interesse. Deste modo, é lógico que se pense em isolar nas imagens as zonas onde existam essas colunas de bolhas e somente transmitir esses blocos ao decodificador. Esta fase de selecção, para transmissão, somente dos blocos que contenham movimento resultante da ascensão das bolhas gasosas funciona como uma primeira etapa no sistema global de compressão. Para se proceder a esta separação é, contudo, necessário utilizar-se filtros que permitam fazer a separação entre movimentos com direcções diferentes. Estes filtros são genericamente designados por filtros direccionais.

Uma das maneiras possíveis de se representar o movimento numa imagem, é através do fluxo óptico. O fluxo óptico pode ser definido como sendo um conjunto de vectores bi-dimensionais da velocidade de pequenas regiões do campo visual. Por isso, muitos dos algoritmos desenvolvidos para a determinação de movimento em imagens são baseados em técnicas que recorrem ao uso do fluxo óptico. Conseguem-se assim determinar diferentes zonas da imagem, as quais, tendo diferentes velocidades indicam diferentes zonas de movimento. Alguns desses algoritmos utilizam dois *frames* de uma sequência de imagens para extrair o fluxo óptico, outros procuram características comuns nas imagens usando para isso dois *frames* consecutivos ou calculando alterações na intensidade dos *pixels* entre *frames* sucessivos. Todavia, em imagens muito texturadas estas abordagens podem não se mostrar adequadas. Estas imagens apresentam demasiadas características para serem comparadas com sucesso e o gradiente de direcção da imagem pode variar, aleatoriamente, de ponto para ponto. De facto, grandes erros são obtidos quando a imagem é fortemente texturada, isto é, os erros aparecem precisamente em zonas onde pode existir maior informação sobre o movimento.

As imagens do fundo submarino, cuja compressão e transmissão para a superfície

é necessário efectuar, são fortemente texturadas. Para além disto, como são imagens de zonas onde a actividade hidrotermal é bastante elevada, este facto cria condições para a existência de micro-climas ambientais que favorecem o aparecimento de uma elevada quantidade de algas marinhas. Estando em constante movimento devido às correntes marítimas estas algas provocam disparidades bastante grandes na análise do movimento.

Por outro lado, em qualquer algoritmo de segmentação é necessário encontrar um contraponto entre a complexidade do modelo e a análise de estabilidade. Modelos insuficientes resultam, inevitavelmente, em excessiva segmentação. Modelos mais elaborados introduzirão mais complexidade e requererão mais cálculos e restrições para se obter estabilidade. A finalidade da segmentação deverá ser a de explorar coerências espaciais e temporais nos dados, de modo a identificar adequadamente regiões de movimento coerente, utilizando para tal, modelos simples de determinação de movimento. Então, na segmentação baseada no movimento, os *pixels* deverão ser agrupados com base na sua similaridade de movimento.

Existem dois outros aspectos que também devem ser considerados no desenvolvimento de algoritmos para a determinação de movimento. Por um lado, a capacidade de lidar com o chamado problema da abertura e, por outro, o facto da saída do filtro não depender somente da velocidade do estímulo mas particularmente da sua frequência espacial, frequência temporal e do contraste.

A consideração de todos estes factores, levou à procura de métodos alternativos para a determinação do movimento e isolamento, das zonas correspondentes às bolhas gasosas, nestas imagens submarinas. Os bons resultados preliminares, obtidos com a utilização de filtros de Gabor para a determinação de movimento nestas imagens, incentivou à sua adaptação a este tipo específico de imagens e para as necessidades particulares de compressão. Outra forte motivação para a sua utilização foi o facto, de sendo filtros direccionais, poder-se optimizar o seu desempenho para saídas segundo determinadas direcções. Deste modo e *a priori*, a utilização de filtros de Gabor modificados mostrava-se adequada para a detecção e isolamento das

colunas ascendentes de bolhas. Pode-se então fazer uma segmentação das imagens em zonas sem movimento e zonas com movimento correspondente às bolhas gasosas.

Assim, foi desenvolvido um algoritmo baseado em filtros espaço-temporais de Gabor que, levando em consideração todos estes factos, permite fazer a determinação do movimento nas sequências de imagens submarinas onde se regista actividade hidrotermal. Este algoritmo utiliza um número não muito grande de *frames* amostrados muito próximo no tempo e consegue determinar e isolar bastante bem as zonas com movimento devido às bolhas gasosas.

3.3.3 Detecção de Movimento com Filtros de Gabor

Como foi analisado na secção anterior, algumas propriedades do movimento nas imagens são mais evidentes no domínio de Fourier [31, 32]. No caso do movimento numa dimensão, o espectro de potência de um sinal ocupa uma linha no domínio da frequência espaço-temporal. De modo análogo, o espectro de potência da translacção de uma textura bi-dimensional ocupa um plano inclinado no domínio da frequência. A frequência espacial do movimento de uma onda sinusoidal é expressa em ciclos por unidade de distância, por exemplo, ciclos por *pixel*, e a sua frequência temporal é expressa em ciclos por unidade de tempo, em geral ciclos por *frame*. A velocidade, definida como a distância no tempo de *pixels* por *frame*, é igual à frequência temporal dividida pela frequência espacial:

$$v = W_t/W_x.$$

Quando um sinal é amostrado igualmente no tempo com componentes de frequência maiores do que a frequência de Nyquist (1/2 ciclo por *frame*) torna-se sub-amostrado ou aliasado. Como consequência uma onda sinusoidal, cujo padrão é deslocado mais do que metade do seu período de *frame* para *frame*, pode aparentar mover-se na direcção oposta. Deste modo, o intervalo de possíveis velocidades para o movimento de uma onda sinusoidal está limitado pela sua frequência espacial.

Consideremos agora um sinal uni-dimensional movendo-se com uma dada velocidade e que tenha várias componentes de frequência espacial. Cada um desses componentes W_x tem uma frequência temporal $W_{t_1} = W_x v$, enquanto cada componente de frequência espacial $2W_x$ tem duas vezes a frequência temporal $W_{t_2} = 2W_x v$ [31]. De facto, a frequência temporal deste sinal é uma função da sua frequência espacial e é uma linha que passa pela origem e cujo declive é v , a velocidade de movimentação do sinal.

De modo análogo uma textura bi-dimensional, com movimento de translação no plano da imagem, ocupa um plano no domínio da frequência espaço-temporal:

$$W_t = uW_x + vW_y,$$

onde $V = (u, v)$ é a velocidade da textura.

Se o movimento de uma pequena região de uma imagem poder ser aproximado por uma translação no plano da imagem, a velocidade dessa região pode ser calculada no domínio de Fourier por determinação do plano no qual toda a energia reside. Considerando estas abordagens, para se extrair o fluxo óptico pode-se considerar pequenas janelas espaço-temporais sobre a sequência de imagens e ajustá-las a cada plano do seu espectro de potência. Como os filtros de Gabor são filtros espaço-temporais sensíveis ao movimento podem ser usados para estimar a velocidade, pois fazem a amostragem de um modo eficiente desse espectro de potência.

Um filtro de Gabor, em fase de seno, é composto por um seno multiplicado por uma janela gaussiana, equação (3.3.1) [31]:

$$g(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{\frac{-t^2}{2\sigma^2}\right\} \sin(2\pi Wt). \quad (3.3.1)$$

O espectro de potência de uma onda sinusoidal é um par de impulsos localizados em W e $-W$ no domínio da frequência. O espectro de potência de uma gaussiana é também uma gaussiana, isto é, é um filtro passa-baixo. Dado que a multiplicação no domínio do espaço (ou do tempo) é equivalente à convolução no domínio da frequência, o espectro de potência de um filtro de Gabor é a soma do

par de Gaussianas centrado em W e $-W$ no domínio da frequência, isto é, é um filtro passa-banda.

A extensão desta representação uni-dimensional dos filtros de Gabor a duas ou três dimensões é possível desde que se imponham restrições de separabilidade. Seja,

$$\mathbf{u} = [xyt]^T \quad \text{e} \quad \omega_0 = [\omega_{x_0}\omega_{y_0}\omega_{t_0}]^T, \quad (3.3.2)$$

respectivamente, o vector espaço-temporal das coordenadas e a frequência central do filtro 3D. Deste modo, o Filtro de Gabor 3D pode ser descrito pela seguinte representação quadrática:

$$\begin{aligned} g_{\sin_i}(\mathbf{u}) &= (2\pi)^{-3/2}(\det\Lambda)^{-1/2} \exp\left(-\frac{1}{2}\mathbf{u}^T \Lambda^{-1} \mathbf{u}\right) \sin(2\pi\omega_0^T \mathbf{u}) \\ g_{\cos_i}(\mathbf{u}) &= (2\pi)^{-3/2}(\det\Lambda)^{-1/2} \exp\left(-\frac{1}{2}\mathbf{u}^T \Lambda^{-1} \mathbf{u}\right) \cos(2\pi\omega_0^T \mathbf{u}), \end{aligned} \quad (3.3.3)$$

onde $\Lambda = \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_t^2)$ é uma matriz diagonal e $(\det\bullet)$ refere-se ao operador determinante. A função de transferência do filtro de Gabor é a soma de um par de gaussianas centradas em ω_0 e $-\omega_0$ no domínio da frequência. Por outro lado, $(\omega_{x_0}, \omega_{y_0}, \omega_{t_0})$ é a frequência central, isto é, é a frequência espacial e temporal para as quais este filtro dá a sua saída máxima e $(\sigma_x, \sigma_y, \sigma_t)$ é o espalhamento da janela gaussiana espaço-temporal.

É relativamente simples sintonizar o filtro para diferentes frequências e orientações, embora mantendo a mesma largura de banda. Para se alterar a sintonização da frequência varia-se, independentemente, $\omega_{x_0}, \omega_{y_0}$ e ω_{t_0} . Estreitando-se a janela gaussiana no domínio espaço-temporal alarga-se a janela passa-banda no domínio da frequência espaço-temporal e vice versa.

A soma do quadrado da saída do filtro com fase em seno com o quadrado da saída do filtro com fase em co-seno, isto é filtros com fase par e com fase ímpar mas com idêntica orientação e largura de banda, dá-nos a medida da energia de Gabor a qual é então invariante à fase do sinal. A frequência de resposta deste filtro de energia de Gabor é a soma do par de gaussianas 3D dada por [31]:

$$\begin{aligned}
G(\omega_x, \omega_y, \omega_t) = & 1/4 \times \exp\{-4\pi^2[\sigma_x^2(w_x - w_{x_0})^2 + \sigma_y^2(w_y - w_{y_0})^2 + \sigma_t^2(w_t - w_{t_0})^2]\} \\
& + 1/4 \times \exp\{-4\pi^2[\sigma_x^2(w_x + w_{x_0})^2 + \sigma_y^2(w_y + w_{y_0})^2 + \sigma_t^2(w_t + w_{t_0})^2]\}.
\end{aligned}
\tag{3.3.4}$$

Esta equação mostra que um filtro de energia de movimento, com a frequência central $(W_{x_0}, W_{y_0}, W_{t_0})$, dá uma saída de $G(W_x, W_y, W_t)$ para o movimento de um seno com as frequências espacial e temporal (W_x, W_y, W_t) . O filtro dará a maior saída para um estímulo que tenha a maior parte da energia próxima da frequência central do filtro e dará a menor saída para um estímulo que tenha pouca ou nenhuma energia próxima da frequência central do filtro.

Com base nestas funções, pode-se construir um modelo baseado num banco de filtros de Gabor, todos eles sintonizados para a mesma banda de frequência espacial W_0 , mas para diferentes orientações espaciais e frequências temporais, isto é, $W_0 = \sqrt{W_{x_0}^2 + W_{y_0}^2}$ é constante para todos os filtros dum mesmo grupo. Deste modo, podem-se construir vários bancos de filtros, cada um deles sintonizados para uma diferente banda de frequência espaço-temporal.

3.3.4 Aplicação de Filtros de Gabor na Detecção do Movimento nas Imagens Submarinas

Os passos atrás descritos, constituem uma base comum a alguns algoritmos que utilizam filtros de Gabor para a determinação de movimento em sequências de imagens [31, 33, 34]. De um modo geral, tais algoritmos, em passos seguintes, procurarão extrair padrões de fluxo baseados numa estimação de mínimos quadrados da velocidade [31] ou no cálculo de coeficientes da transformada 2-D de Gabor para imagens diferença [33]. Em [34], explora-se o facto de cada um destes padrões de fluxo corresponder a uma zona de imagem com diferente velocidade pelo que se consegue dividir a imagem em *layers*.

Para atingir o objectivo de determinar a velocidade em todas as zonas das imagens, nestes algoritmos são desenvolvidas técnicas bastante complexas as quais nem sempre conduzem a bons resultados, pois tais resultados podem depender de condições iniciais difíceis de obter. Por exemplo em [31], é utilizado o método da descida de gradiente de Gauss-Newton com os problemas inerentes à sua inicialização.

Com base nestas dificuldades e no facto de não interessar para efeitos de compressão das imagens determinar com exactidão as velocidades nas diferentes zonas da imagem, pois o que interessa, recorde-se, é tão somente isolar os blocos das imagens com movimento ascendente originado pelas bolhas gasosas, optou-se por desenvolver uma técnica alternativa. Nesta técnica, tira-se partido das potencialidades dos filtros de Gabor de detectarem movimentos em direcções preferenciais em sequências de imagens. Isto permitiu, de um modo relativamente simples, segmentar as imagens em termos de zonas com movimento devido às bolhas gasosas e zonas sem esse movimento. Por outro lado, a aplicação directa dos métodos referidos no início às imagens utilizadas seria de reduzida utilidade, pois nestas imagens há muitos movimentos espúrios devidos aos movimentos das algas. Assim, os algoritmos referidos poderiam perder-se na determinação das zonas com movimento pois foram idealizados para imagens onde existem somente algumas regiões bem demarcadas tendo cada uma delas uma velocidade diferente, como por exemplo em [34].

Tendo em conta estas particularidades, foi desenvolvido um algoritmo baseado na medida de energia de Gabor indicada na equação (3.3.5), o qual será descrito de seguida.

Considere-se uma sequência de imagens, figura 3.2, dividida em S blocos tridimensionais B_S cada um com dimensão $(k \times k) \times t$, onde $(k \times k)$ é o tamanho de uma grelha espacial com $k \times k$ *pixels* e t é o número de *frames* considerados no tempo. A cada um desses S blocos tridimensionais aplique-se um filtro de Gabor o qual, recorde-se, é constituído por um par de gaussianas, uma em seno e a outra em co-seno.

Seja $\{g_i\}_{i=1}^D$, o banco de filtros de Gabor, onde cada i indexa a direcção espacial

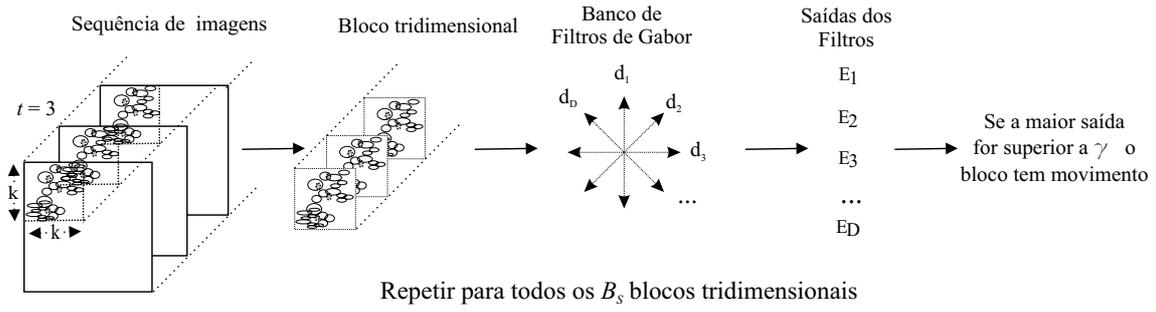


Figura 3.2: Divisão das imagens em blocos tridimensionais e aplicação dos filtros

d_i e seja E_i , a energia da saída de $g_i, i = 1, \dots, D$, para uma dada entrada de bloco tridimensional.

A medida da energia de Gabor para o total de pares de filtros D usados, onde $g_i, i = 1 \dots D$ é o i -ésimo filtro utilizado, é dada por:

$$E_i = \int \int \int |CG \sin(i)|^2 dx dy dt + \int \int \int |CG \cos(i)|^2 dx dy dt, \quad (3.3.5)$$

em que $CG \sin(i)$ e $CG \cos(i)$ é o resultado da convolução de cada bloco tridimensional B_S da sequência com o i -ésimo filtro de Gabor em seno e em co-seno respectivamente, isto é:

$$CG \sin(i) = conv(B_k, g_{\sin(i)}(x, y, t))$$

$$CG \cos(i) = conv(B_k, g_{\cos(i)}(x, y, t)).$$

Se considerarmos o que atrás se disse sobre os filtros de Gabor é fácil concluir-se que o máximo de todas estas medidas de energia E_i , para um dado bloco, corresponderá à energia máxima nesse bloco. O significado físico deste máximo é que o filtro (i), para o qual este valor máximo foi atingido, corresponde à direcção do movimento predominante nesse bloco. Recorde-se que o movimento nas imagens com registo de actividade hidrotermal, é no sentido ascendente e numa direcção quase vertical. Na construção dos bancos de filtros de Gabor deve-se então privilegiar filtros que dêem saídas máximas para direcções verticais ou quase verticais. Assim, as saídas

máximas de todos os filtros, no conjunto total da imagem, serão obtidas para as zonas onde existam colunas verticais de bolhas.

Obviamente que, mesmo para zonas da imagem onde não existam colunas verticais de bolhas, obtêm-se valores máximos de saída nos bancos de filtros correspondentes aos movimentos mais fortes nessas zonas da imagem. Por exemplo, no caso das imagens utilizadas, como existem também movimentos de algas marinhas obtêm-se saídas que corresponderão ao movimento dessas mesmas algas.

Não tendo contudo estes movimentos uma direcção correspondente à que os filtros estejam a privilegiar terão uma saída máxima relativamente inferior à obtida para movimentos correspondentes às direcções privilegiadas pelos filtros. Torna-se então necessário dispor de um mecanismo que nos permita fazer a separação entre estes movimentos. Para fazer a separação entre estes movimentos bastará então considerar um limiar que permita classificar estes movimentos entre movimentos de interesse e espúrios, isto é, movimentos correspondentes às bolhas e outros.

Utiliza-se então um banco de filtros de Gabor para analisar cada bloco tridimensional em que se dividiu a sequência de imagens. Todos estes filtros estarão sintonizados para a mesma banda de frequência espacial mas terão distintas orientações espaciais. Deste modo, podemos amostrar a orientação espacial do movimento em cada bloco tridimensional das imagens e então, determinar o movimento na direcção dominante, figura 3.2.

Associamos à direcção dominante do movimento, nesse bloco tridimensional, a energia:

$$E_m = \arg \max_i \{E_i\}_{i=1}^D, \quad (3.3.6)$$

onde em $m = 1, \dots, S$ se consideram todos os blocos tridimensionais da sequência de vídeo em análise. Dado o conhecimento, *a priori*, que temos sobre as características principais das imagens, isto é, o movimento dominante nas bolhas ser ascendente, podemos isolar os blocos tridimensionais onde a probabilidade desta ocorrência for

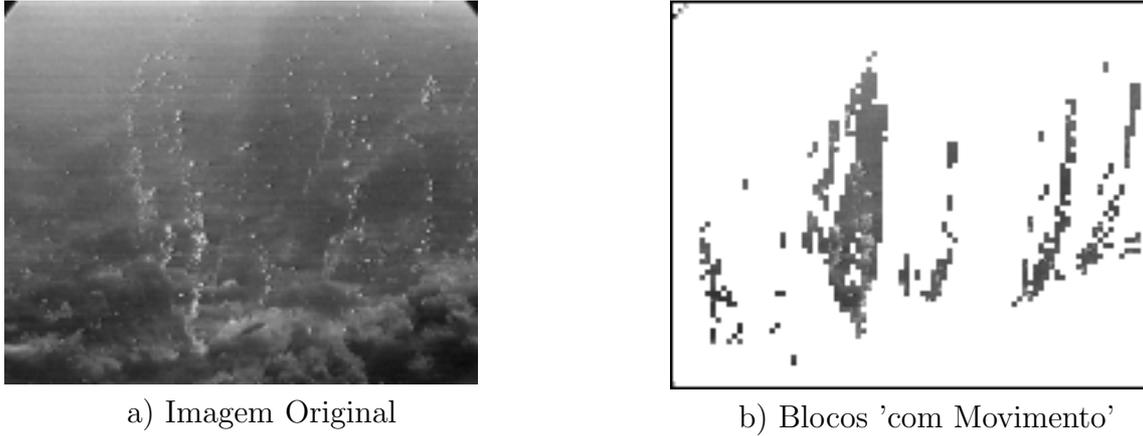


Figura 3.3: Classificação dos Blocos - Sequência Açores 1

máxima. Definindo as hipóteses,

$$\begin{aligned}
 H_0 & : \text{não existe movimento de bolhas} \\
 H_1 & : \text{existe movimento de bolhas}
 \end{aligned}
 \tag{3.3.7}$$

podemos efectuar o seguinte teste:

$$E_m \underset{H_1}{\overset{H_0}{\leq}} \gamma, m = 1, \dots, S,
 \tag{3.3.8}$$

onde γ é um limiar adequadamente escolhido.

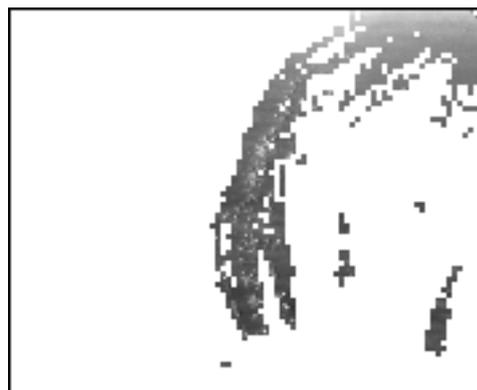
Podemos considerar para limiar de separação o valor correspondente à média entre todos os valores de energia de Gabor, obtidos para cada um dos S blocos tridimensionais B_S em que se dividiu a imagem, isto é:

$$\gamma = \text{média}(\{E_m\}_{m=1}^S).
 \tag{3.3.9}$$

Esta técnica de classificação entre blocos 'com movimento' e blocos 'sem movimento' é bastante simples, fácil de implementar e computacionalmente leve. No entanto permite obter bons resultados como se pode constatar pelas figuras 3.3, 3.4 e 3.5. Nestas figuras podem observar-se as imagens originais e os blocos classificados como 'com movimento'.

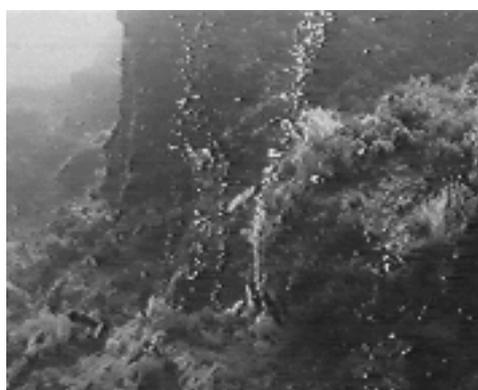


a) Imagem original



b) Blocos 'com Movimento'

Figura 3.4: Classificação dos Blocos - Sequência Açores 2



a) Imagem original



b) Blocos 'com Movimento'

Figura 3.5: Classificação dos Blocos - Sequência Açores 3

3.4 Sumário

Neste capítulo fez-se uma análise de técnicas que permitem fazer a segmentação das imagens em zonas com interesse ou sem interesse.

Considerando que qualquer algoritmo que seja desenvolvido para a segmentação de imagem deve ter em conta as particularidades do sistema visual humano, no que respeita à sua capacidade para destringar artefactos nas imagens, abordaram-se questões relacionadas com a sensibilidade do olho e maneiras de se ultrapassar essas dificuldades. Fez-se também a análise de técnicas que, utilizando filtros direccionais, permitem a selecção segundo direcções preferenciais de zonas com movimento.

Com base nestas considerações, propôs-se um algoritmo que permite fazer a segmentação das imagens em zonas que contenham blocos onde se registre movimento

de bolhas gasosas, isto é, zonas 'com movimento' e zonas 'sem movimento'. Somente os blocos classificados como 'com movimento' serão transmitidos após serem comprimidos. Este algoritmo de segmentação baseado em filtros espaço-temporais direccionais de Gabor permitiu identificar e isolar bastante bem os blocos de imagem correspondentes a zonas com movimento.

Os blocos, constituintes da zona de *background*, isto é, os blocos classificados como 'sem movimento', só serão enviados esporadicamente e integrados numa imagem de referência, sobre a qual se dispõem os blocos 'com movimento' das outras imagens, de modo a reconstituírem-se essas outras imagens.

No capítulo seguinte, descreve-se a compressão dos blocos seleccionados para transmissão, utilizando-se técnicas de Quantização Vectorial, modificadas para comprimir mais do que um bloco em simultâneo.

Capítulo 4

Compressão de Vídeo Submarino

4.1 Introdução

A fase de segmentação baseada em filtros de Gabor corresponde a uma pré-compressão bastante significativa, com a conseqüente grande redução da quantidade de informação presente. No entanto, considerando a largura de banda disponível no canal acústico submarino, esta pré-compressão não é ainda suficiente para permitir a transmissão de vídeo por este canal.

Deste modo, relativamente aos blocos seleccionados para transmissão, é necessário proceder também à sua compressão. Os algoritmos desenvolvidos para esta segunda fase de compressão foram baseados na Quantização Vectorial. Esta escolha da quantização vectorial resultou do facto de, relativamente às técnicas referidas no capítulo 2 para a compressão de imagem, esta técnica ser a que pareceu, de acordo com ensaios prévios efectuados, passível de permitir a obtenção de melhores resultados, quer a nível de taxa de compressão, quer a nível de qualidade visual das imagens. Mas, a aplicação directa das técnicas de compressão de imagens isoladas, baseadas em Quantização Vectorial, à compressão de sequências de imagens, se bem que permita obter taxas de compressão elevadas, não se mostra também e ainda suficiente para obter resultados que permitam a transmissão de vídeo através do canal acústico.

Por isso, foram desenvolvidas diferentes estratégias baseadas na Quantização

Vectorial, as quais, em conjugação com o algoritmo de selecção e segmentação baseado em filtros de Gabor referido no capítulo anterior, permitiram aumentar a taxa de compressão, sem no entanto provocar uma grande perda de qualidade visual das imagens.

As estratégias desenvolvidas para esta fase da compressão, e que permitem codificar mais do que um bloco de imagem em simultâneo, são descritas nas secções seguintes.

4.2 Compressão de Sequências de Imagens com Quantização Vectorial Multibloco

Na compressão de imagem baseada em Quantização Vectorial, e tal como foi descrito no capítulo 2, as imagens são particionadas em blocos e cada um desses blocos é codificado, determinando-se no *codebook* o *codevector* mais próximo do vector constituído pelos *pixels* do bloco de imagem que se quer transmitir. Encontrado esse *codevector*, basta transmitir ao decodificador o índice correspondente e o decodificador, efectuando as operações inversas das do codificador, reconstrói o bloco de imagem.

Se bem que para imagens estáticas estas técnicas permitam obter taxas de compressão razoáveis, as quais são directamente proporcionais ao tamanho do bloco a codificar e ao conseqüente tamanho do vector de *codebook*, a sua aplicação a sequências de imagens não permite tirar partido, directamente, de alguma eventual redundância que possa existir em blocos correspondentes nas várias imagens da sequência. De facto, os vectores do *codebook* correspondem a blocos bi-dimensionais da imagem. Seria desejável que os vectores do *codebook* correspondessem a blocos de imagem tri-dimensionais em que a terceira dimensão fosse logicamente o tempo. Isto possibilitaria que um índice num *codebook* correspondesse, não somente a um bloco de uma imagem, mas sim a um bloco tri-dimensional na sequência de imagens.

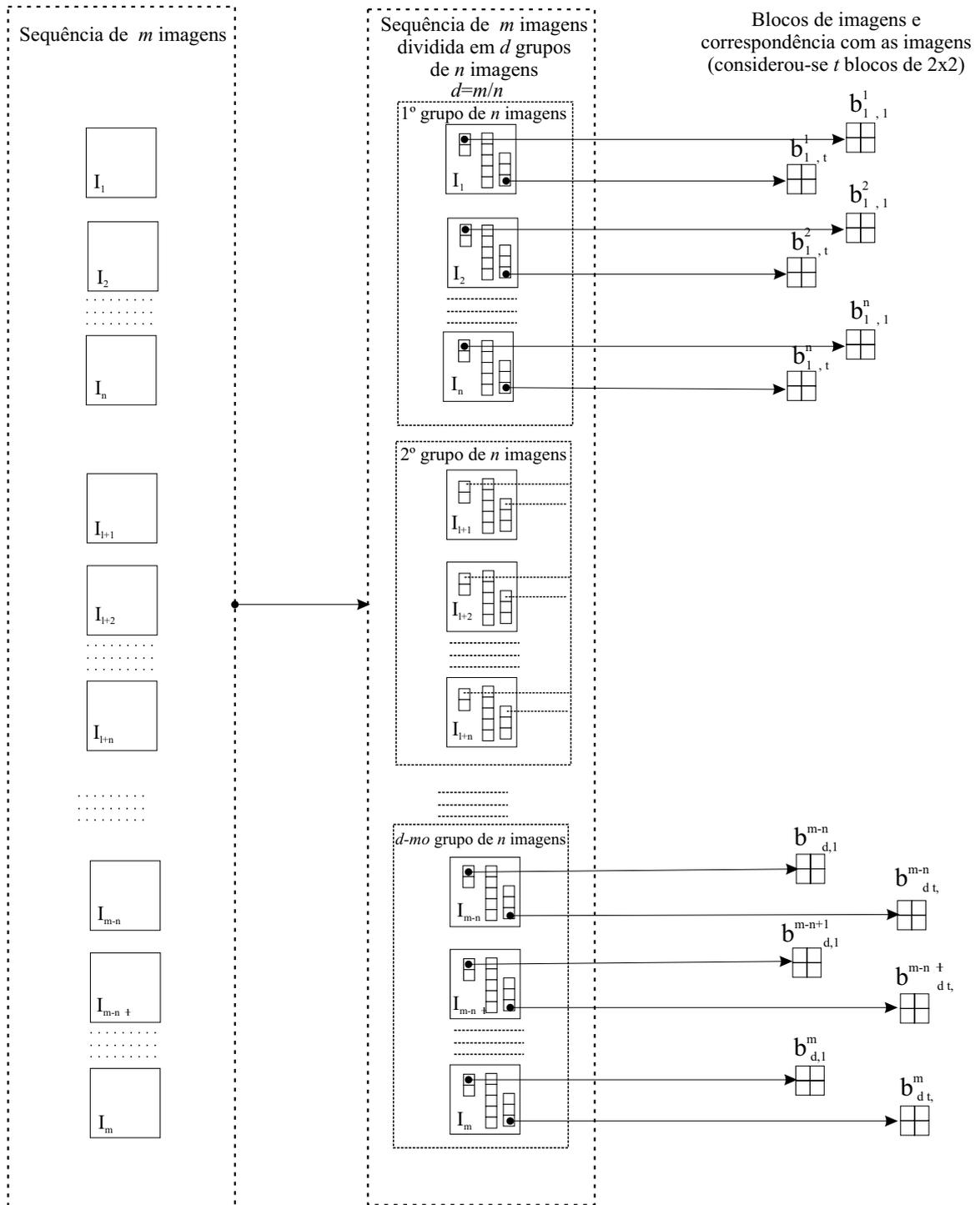
Desse modo, se cada um dos índices do *codebook* se referir a mais do que uma imagem, a taxa de compressão inerente a esta modificação será maior e proporcional ao número de imagens em cada bloco tri-dimensional.

Na literatura que se consultou, relativa à aplicação de quantização vectorial à compressão de sequências de imagens, encontraram-se apenas 2 referências à possibilidade de codificação de mais do que um bloco [7, 35]. Nestas referências, os autores não são muito explícitos sobre a maneira de o fazer nem detalham uma possível implementação. Em [35], apenas é indicada uma medida de distorção aplicável à determinação de índices num *codebook* para mais do que uma imagem. Esta medida não é mais do que a extensão da medida de distorção apresentada no capítulo 2 para a Quantização Vectorial de imagens estáticas.

Para uma melhor percepção da exposição que se irá efectuar, esquematiza-se na figura 4.1 a divisão efectuada nas imagens. De modo a poder aplicar-se filtros de Gabor, a sequência de m imagens deve ser dividida em grupos de n imagens, isto é, no número de imagens que se quer comprimir em simultâneo.

Alternativamente, pode-se dividir esta sequência em grupos com um número de imagens diferente do número de imagens a comprimir em simultâneo na 2ª etapa de compressão. No entanto, neste caso, é preciso que esse número seja um múltiplo de n . Isto é necessário porque, embora estas duas etapas, a de segmentação e a de compressão, sejam independentes, não deixam de estar intimamente relacionadas, fornecendo a primeira à segunda os blocos que irão ser aí comprimidos. Daí, poder ser diferente a divisão da sequência de imagens para a 1ª e para a 2ª etapa. A necessidade de, no caso de serem diferentes, terem de ser um múltiplo de n é para não ficarem imagens por comprimir. Para não dificultar a exposição do algoritmo, será considerado que se fez a divisão da sequência de m imagens em grupos de n imagens, quer para a segmentação, quer para a compressão.

Podem ser identificadas duas etapas principais no modo como é efectuada a compressão/descompressão deste tipo de vídeo submarino:



$b_{l,t}^n$ = t -ésimo bloco de imagem do grupo l referente à n -ésima imagem desse grupo

Figura 4.1: Estrutura da divisão da sequência

1. Segmentação da sequência de imagens em blocos 'com movimento' devido às bolhas gasosas e blocos 'sem movimento' devido às bolhas gasosas, *background*.
2. Compressão dos blocos classificados como 'com movimento'. Três alternativas podem ser usadas nesta etapa, tendo a primeira delas 2 variantes:
 - (a) Compressão baseada em Quantização Vectorial com *n-codebooks*:
 - i. com *n-codebooks* 2.5D, Compressão VQ2.5DN;
 - ii. com *n-codebooks* 3D, Compressão VQ3DN;
 - (b) Compressão baseada em Quantização Vectorial com *codebook* de níveis de cinzento e *codebook* de índices, *i-codebooks*, Compressão VQIND;
 - (c) Compressão baseada em Quantização Vectorial com codificação dos índices através do algoritmo de dicionário adaptativo de Lempel-Ziv, Compressão VQZIP.

As duas etapas principais da fase de compressão do vídeo submarino são representadas graficamente nas figuras 4.2 e 4.3. Para não sobrecarregar a imagem, representa-se na figura 4.3 cada uma das operações feitas na compressão simultânea de n grupos de imagens por um só bloco. Cada um destes blocos representa um dos métodos alternativos que se podem usar nesta etapa, cada um com especificidades próprias.

Uma descrição gráfica mais detalhada de cada um dos compressores será mostrada nas figuras 4.6, 4.10, 4.19 e 4.26. A representação gráfica detalhada do bloco referente ao decodificador correspondente será mostrado nas figuras 4.7, 4.11, 4.20 e 4.27.

Nas secções seguintes detalham-se cada uma destas técnicas de compressão propostas.

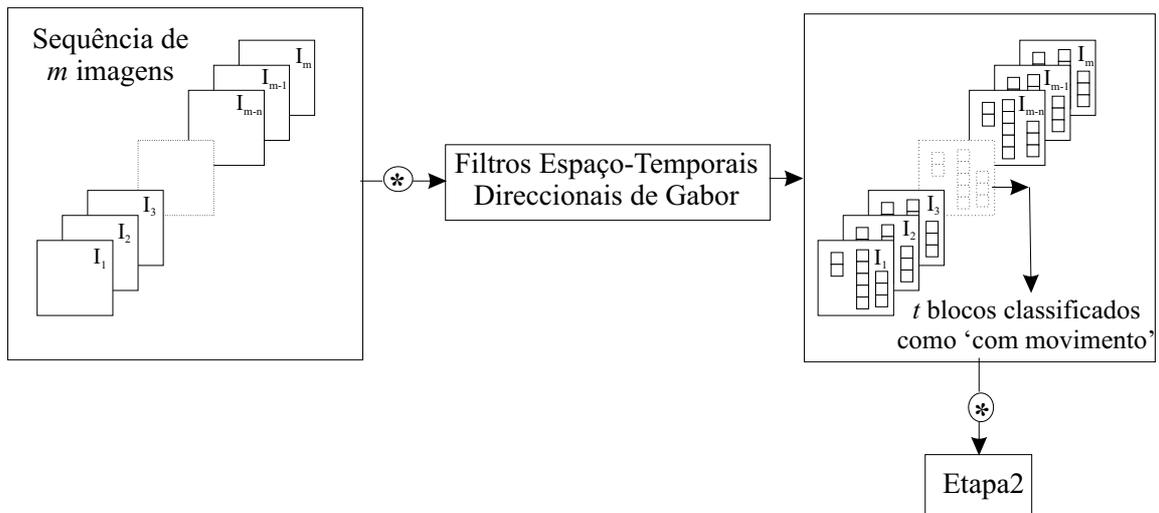


Figura 4.2: Etapa 1 do algoritmo de compressão

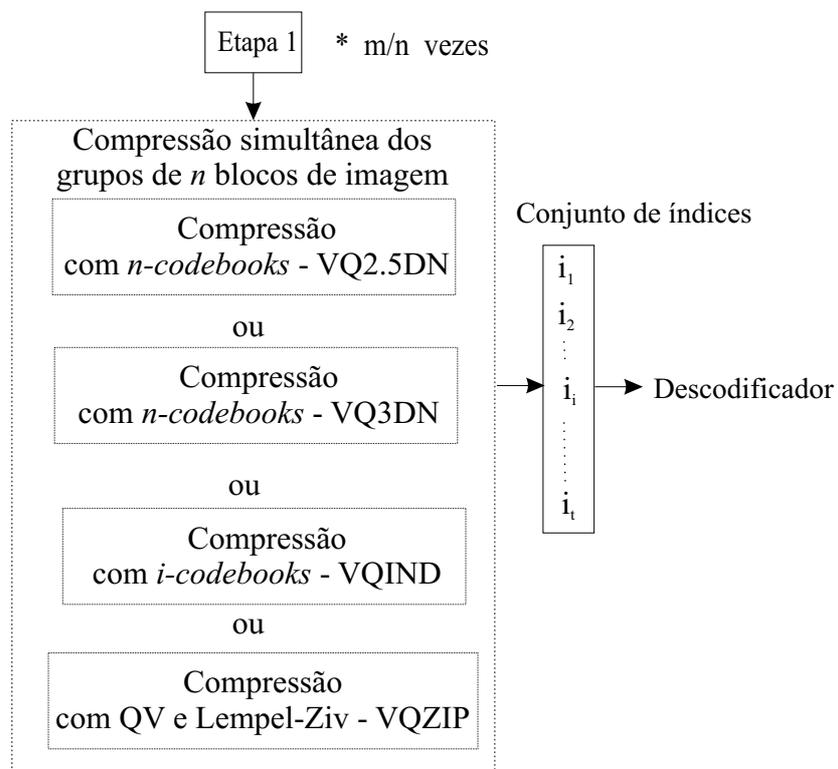


Figura 4.3: Etapa 2 do algoritmo de compressão

4.3 Compressão de Imagens com Quantização Vectorial com n -codebooks

Nesta codificação utilizam-se n -codebooks, isto é, *codebooks* com mais do que um plano. Por plano, entenda-se cada conjunto de p níveis ou *codevectors* diferentes, cada um com dimensão \mathcal{L} , referidos na secção 2.4.2. Assim, podemos ter n -codebooks com n planos diferentes os quais nos permitirão comprimir em simultâneo n blocos de imagens, dado este número de blocos ser igual ao número de planos.

Para cada uns dos n blocos de imagem de tamanho $(k \times k)$ a comprimir, determina-se a entrada no *codebook* correspondente ao *codevector* de dimensão $(k \times k)$ que melhor descreva esse conjunto de $n(k \times k)$ blocos de imagens em simultâneo. Este conjunto de $n(k \times k)$ blocos de imagens pode ser então representado por um único índice. Deste modo, quando comparada com as técnicas de Quantização Vectorial clássicas, a técnica que propomos, utilizando n -codebooks, permite aumentar a taxa de compressão por um factor de n , pois naquelas cada índice do *codebook* representa um bloco de imagens e aqui representa n blocos de imagem.

Para se codificarem n blocos de imagem de cada vez com um só índice, é necessário ser possível fazer a aproximação dos $n(k \times k)$ blocos de imagem às $n \times p$ entradas do *codebook* e para isso o n -codebook terá de ter $n \times p$ entradas, cada uma de dimensão $(k \times k)$. Esta aproximação pode ser feita utilizando-se n *codebooks* onde cada *codevector* tenha dimensão $(k \times k)$, os quais simulam os n planos. Alternativamente, pode-se utilizar um *codebook* em que cada *codevector* tenha agora dimensão $n(k \times k)$. Podemos pois considerar 2 variações, as quais estão relacionadas com o *codebook* a utilizar:

- O *codebook* pode ser gerado de tal modo que cada um dos n planos seja gerado separadamente dos outros $(n - 1)$ planos. Fazem-se então n operações de geração de planos onde, em cada uma delas, se gera um plano diferente. Para

isto, utilizam-se conjuntos iniciais de valores diferentes cada um deles correspondente a uma imagem de treino. Cada um dos n planos é constituído por *codevectors* de dimensão $k \times k$. A esta variação deu-se o nome de compressão com Quantização Vectorial VQ2.5DN;

- Alternativamente, a geração dos n planos do *codebook* pode ser feita de uma só vez, isto é, geram-se todos os n planos simultaneamente. Para isto, em vez de se considerarem n diferentes conjuntos iniciais de valores, cada um deles referente a uma imagem de treino, considera-se um super conjunto de treino inicial constituído por todas as n imagens de treino. Então, os *codebooks* resultantes terão um só plano e cada *codevector* terá dimensão $n(k \times k)$. Designou-se esta variação por compressão com Quantização Vectorial VQ3DN.

4.3.1 Algoritmo para Compressão VQ2.5DN

Nesta versão, utilizam-se então n *codebooks*, isto é, utilizam-se em simultâneo n *codebooks* diferentes cada um deles gerado em separado.

O n *codebooks* são previamente gerados e posteriormente utilizados para a compressão dos blocos de imagem classificados como 'com movimento'. A não ser que venha a verificar-se ser necessário proceder a um refrescamento dos *codebooks*, estes só são gerados uma única vez no início do processo de compressão.

Para codificar os $n(k \times k)$ blocos de imagens, seleccionam-se os índices I nos n -*codebooks* que melhor se aproximem dos valores de níveis de cinzento correspondentes a cada um dos pixels em cada um dos blocos de imagem.

Cada índice I , o qual codificará o $n(k \times k)$ bloco de imagem, é determinado considerando-se:

$$I \longleftrightarrow \arg \min_j \{e_j^2\}_{j=1}^p. \quad (4.3.1)$$

Nesta expressão considera-se e_j^2 como o erro quadrático médio definido por:

$$e_j^2 = \sum_{i=1}^n \|c_{ij} - img_i\|_F^2, \quad (4.3.2)$$

onde F denota a norma de Frobenius, $img_i, i = 1, \dots, n$, representa cada um dos $n(k \times k)$ blocos de imagens a comprimir e $c_{ij}, i = 1, \dots, n; j = 1, \dots, p$, é uma entrada do *nésimo-codebook*, sendo p a dimensão do *codebook*.

Cada um destes índices I representará a mesma linha em todos os n -*codebooks*.

Exemplifiquemos este método para o caso da compressão de 3 blocos de imagem em simultâneo, isto é, $n = 3$. Para uma melhor percepção das operações considere-se a figura 4.4.

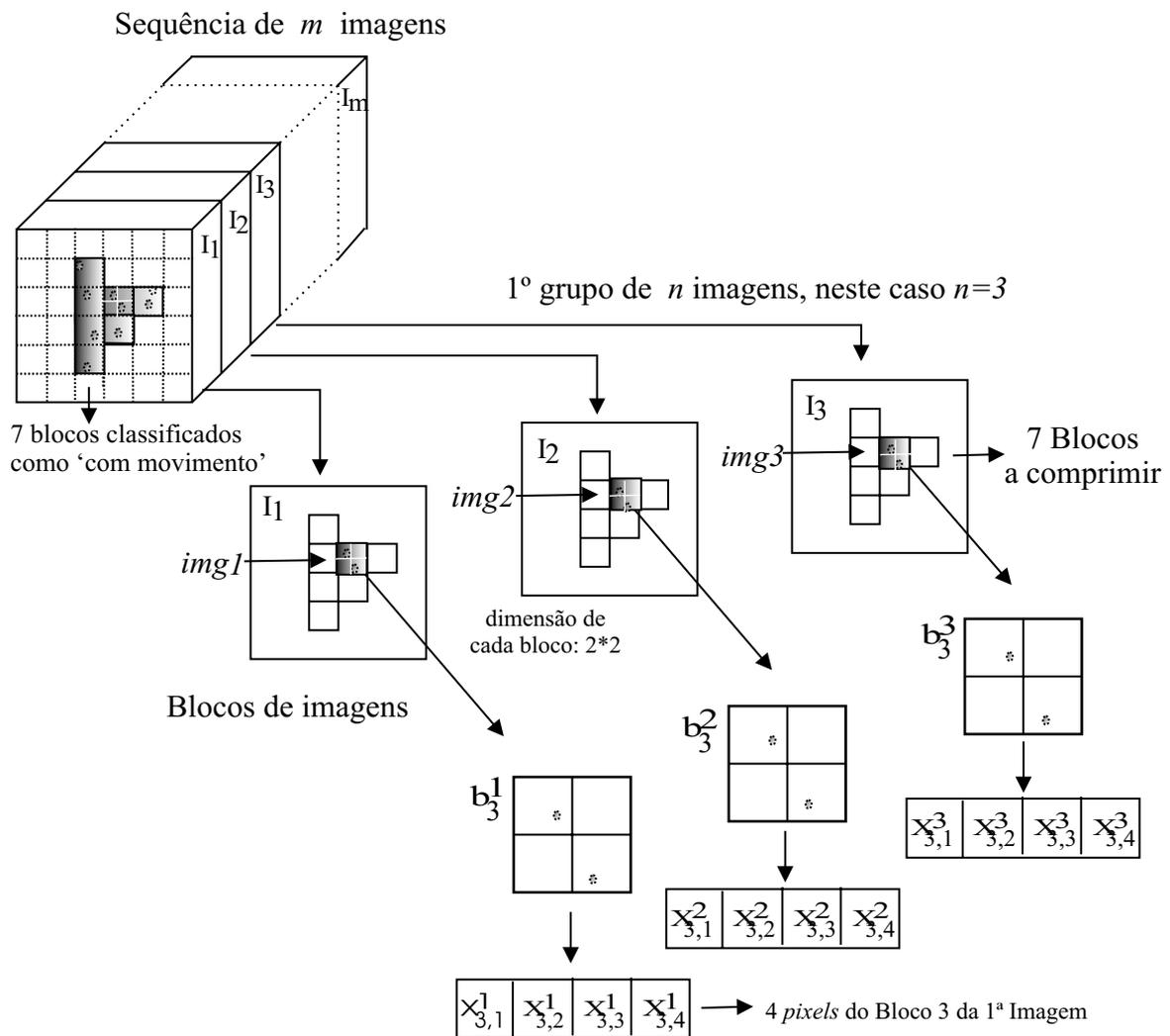


Figura 4.4: Disposição dos blocos b_t^n e dos *pixels* $X_{t,r}^n$ na sequência.
Variante VQ2.5DN

Nesta figura consideramos uma sequência de m imagens, onde foram classificados como 'com movimento' $t, t = 1, \dots, l$ blocos. Estes blocos vão ser comprimidos em grupos de n blocos. Assim, b_t^n refere-se ao t -ésimo bloco classificado como 'com

movimento' na n -ésima imagem.

No exemplo mostrado na figura, considerou-se que foram classificados como 'com movimento' 7 blocos em cada uma das 3 imagens que se estão a comprimir em simultâneo ou seja, temos $n = 3$ e $l = 7$, pois estamos a comprimir em simultâneo 3 imagens e em cada uma delas foram classificados 7 blocos como tendo movimento. Considerou-se que a dimensão espacial k de cada um desses blocos era de 2×2 *pixels*, pelo que, no exemplo da figura, $r = 1, \dots, 4$. Assim, o escalar $X_{3,2}^1$ refere-se ao valor, na escala de cinzentos, do *pixel* que está no 3º bloco a comprimir da 1ª imagem, bloco b_3^1 e dentro deste na 2ª posição.

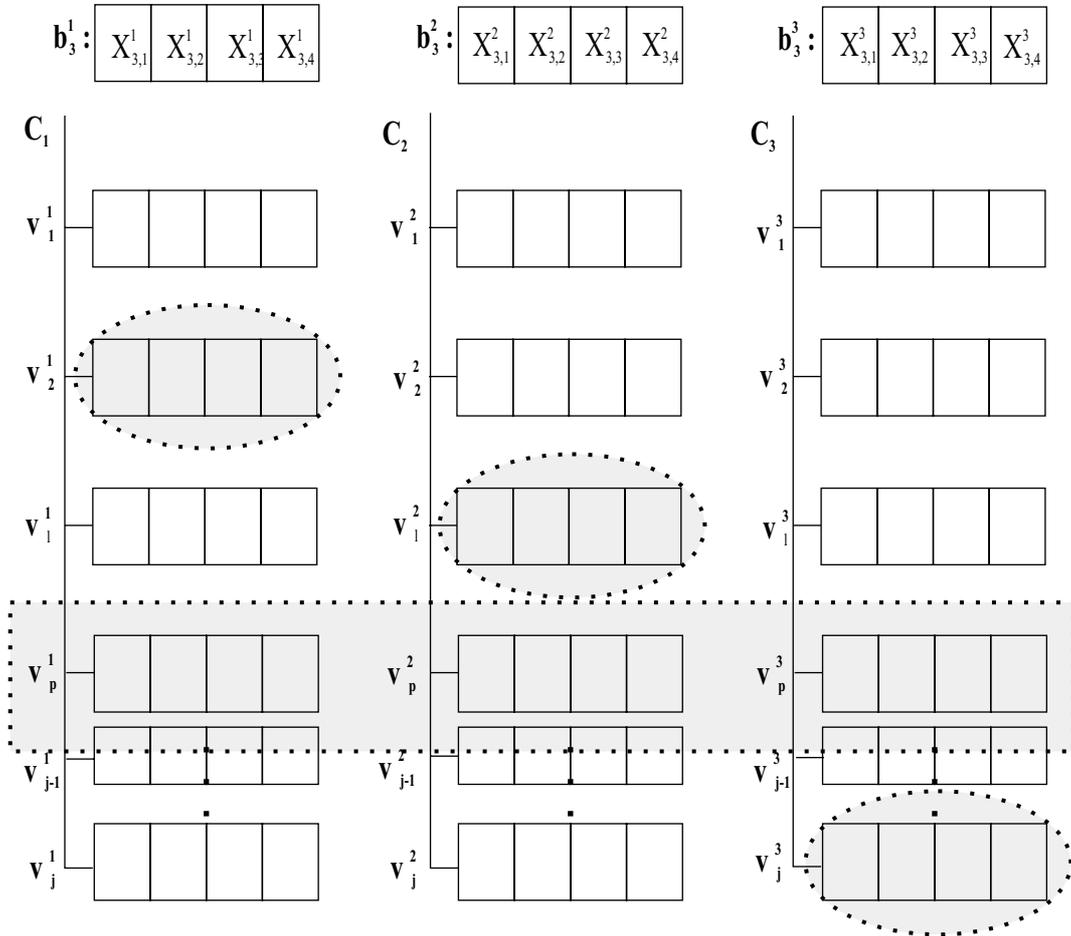


Figura 4.5: Determinação dos índices para os 3 blocos - Variante VQ2.5DN

Estes $X_{t,r}^n$, $r = 1, \dots, (k \times k)$ vão ser comprimidos utilizando-se VQ2.5DN. Para exemplificar esta compressão consideremos o esquema representado na figura 4.5. Neste esquema, sejam C_1 , C_2 e C_3 os n *codebooks*, neste caso $n = 3$, cada um

deles com *codevectors* de dimensão $k \times k$, neste caso $k = 2$. Estes *codebooks* irão ser utilizados para codificar em simultâneo os 3 blocos de imagens, cada um deles também de dimensão 2×2 .

Seja v_j^n , respectivamente, o n -ésimo *codevector* de cada um desses 3 *codevectors*, $n = 1, 2, 3$ e caso se considere *codebooks* com 256 níveis de cinzento, teremos $j = 1, 2, \dots, 256$. Assim, considerando $n = 1$ e $j = 3$, v_3^1 corresponderá ao terceiro *codevector* do primeiro *codebook*, isto é $v_3^1 = \{\hat{X}_{3,1}^1, \hat{X}_{3,2}^1, \dots, \hat{X}_{3,K}^1\}$, onde $K = k \times k$ corresponde à dimensão de cada *codevector* e conseqüentemente, à dimensão espacial de cada bloco de imagem a comprimir, em que, no exemplo mostrado na figura, são considerados blocos de dimensão 2×2 ou seja $K = 2 \times 2 = 4$.

Estes escalares $\hat{X}_{t,r}^n, r = 1, \dots, (k \times k)$ são os valores que serão utilizados na comparação com cada um dos *pixels* $X_{t,r}^n$, presentes em cada bloco b_t^n da imagem n a comprimir, figura 4.4. São escolhidos os que minimizarem uma medida de distância relativamente aos *pixels* desses blocos.

Considere-se b_t^1, b_t^2 e b_t^3 , o t -ésimo bloco a codificar, pertencentes respectivamente, às imagens img_1, img_2 e img_3 e constituídos pelos níveis de cinzento das imagens em cada um desses blocos. Neste exemplo, $t = 3$, ou seja, estaremos a comprimir o 3º bloco classificado como 'com movimento' em cada um destes 3 blocos de imagens, b_3^1, b_3^2 e b_3^3 , figura 4.5.

O conjunto destes 3 blocos de imagem é então comparado com conjuntos de 3 *codevectors*, cada um deles pertencente a cada um dos *codebooks* C_1, C_2 e C_3 .

A determinação do índice que melhor aproxima em simultâneo estes 3 blocos de imagem b_t^n , relativamente ao conjunto dos 3 *codebooks* C_1, C_2 e C_3 , pode ser feita por minimização da medida de distorção baseada no erro quadrático médio, equação (4.3.1).

Assim, para se determinar por exemplo o índice correspondente ao terceiro bloco nas várias imagens, calcula-se para conjuntos de 3 imagens e *codebooks* com 256 níveis de cinzentos.

$$d(\underbrace{b_3^n, v_j^n}_{\substack{n=1,2,3 \\ j=1,2,\dots,256}}) = \sum_{l=1}^K (X_{3,l}^1 - \hat{X}_{j,l}^1)^2 + (X_{3,l}^2 - \hat{X}_{j,l}^2)^2 + (X_{3,l}^3 - \hat{X}_{j,l}^3)^2, \quad (4.3.3)$$

em que b_3^n é o terceiro bloco de imagens a codificar, relativamente à n -ésima imagem, $n = 1, 2, 3$ e v_j^n é o respectivo j -ésimo *codevector* pertencente ao n -ésimo *codebook*, $j = 1, 2, \dots, 256$, para *codebooks* com 256 níveis de cinzento e $n = 1, 2, 3$ planos diferentes. Por sua vez, K é a dimensão de cada bloco em cada uma das imagens, o que corresponde a ser a dimensão de cada *codevector* em cada *codebook*.

Para o exemplo mostrado nas figuras 4.4 e 4.5, considera-se que os blocos de imagem têm dimensão 2×2 , o que implica que cada um dos *codevectors* dos *codebooks* C_1 , C_2 e C_3 , sejam constituídos por 4 elementos, isto é, $K = 4$ na equação (4.3.3).

Na figura 4.5, a zona rectangular a sombreado corresponderá à escolha que, em simultâneo, melhor represente os 3 blocos de imagem a comprimir. O índice seleccionado, $I = l$, corresponde ao *codevector* v_p^l com n cada um dos 3 planos do *codebook*. Esse índice será então transmitido ao decodificador o qual, através de uma simples leitura dos *codevectors* correspondentes ao índice recebido, reconstruirá os 3 blocos das imagens.

Deste modo, verifica-se que um só índice permitirá fazer a reconstituição de 3 blocos de imagens. Relativamente à Quantização Vectorial clássica de imagens, onde cada índice se refere a um só bloco da imagem e consequentemente permite a reconstrução de um só bloco de imagem, aqui consegue-se um aumento de 3 vezes na taxa de compressão pois têm-se 3 vezes menos índices para transmitir.

No exemplo apresentado, considera-se a hipótese de se comprimirem em simultâneo 3 blocos de imagens. É óbvio que este valor pode ser outro, inferior ou superior, reflectindo-se essa escolha, proporcionalmente, na taxa de compressão.

Na figura 4.5, os *codevectors* que, em cada um dos planos do *codebook*, estão assinalados dentro de uma oval sombreada, corresponderiam a eventuais índices óptimos para cada um dos blocos de imagem a comprimir, caso a determinação

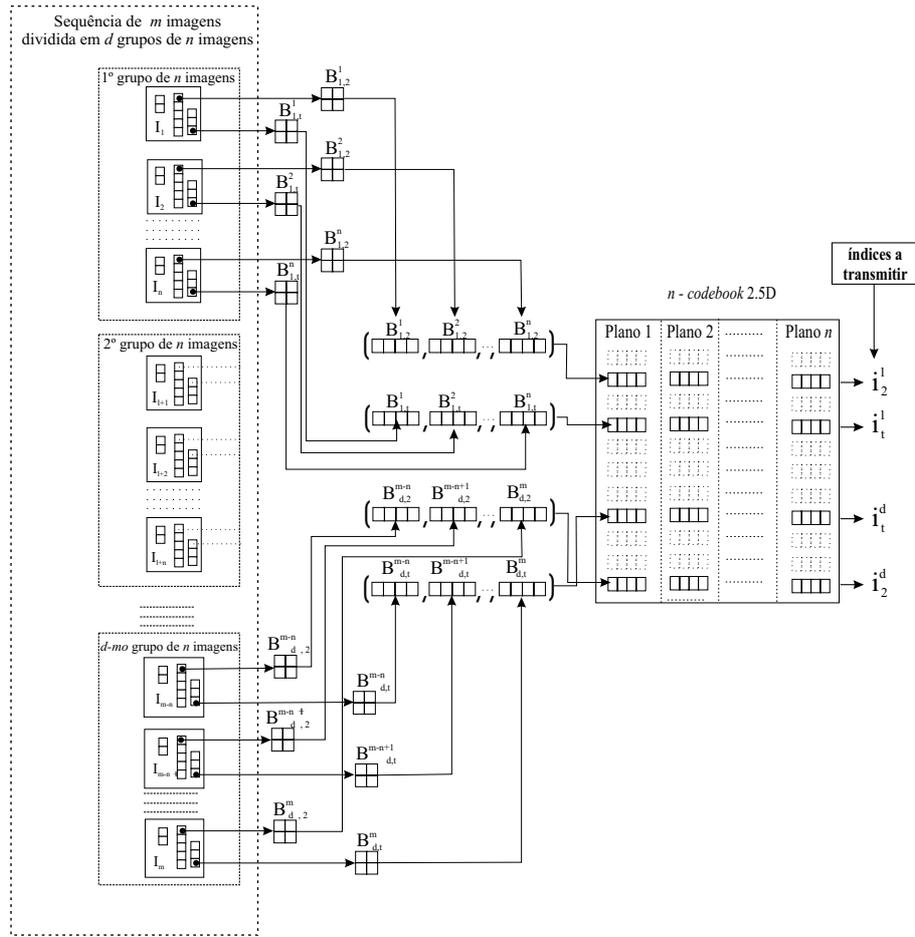


Figura 4.6: Compressor multi-bloco baseado em n -codebook 2.5D

dos índices para cada um dos blocos fosse feita isoladamente e não considerando simultaneamente todos os n .

4.3.2 Resumo do Algoritmo VQ2.5DN

Em resumo, este algoritmo, figuras 4.6 e 4.7, terá então os seguintes passos:

- Fase Prévia \iff Geração do n -codebook 2.5D;
- Fase de Compressão:
 1. Dividir a sequência de m imagens em d grupos de n imagens, $d = m/n$;
 2. Determinar em cada imagem, mas considerando todas as n imagens do grupo, os t blocos com movimento usando filtros de Gabor \iff Etapa 1;

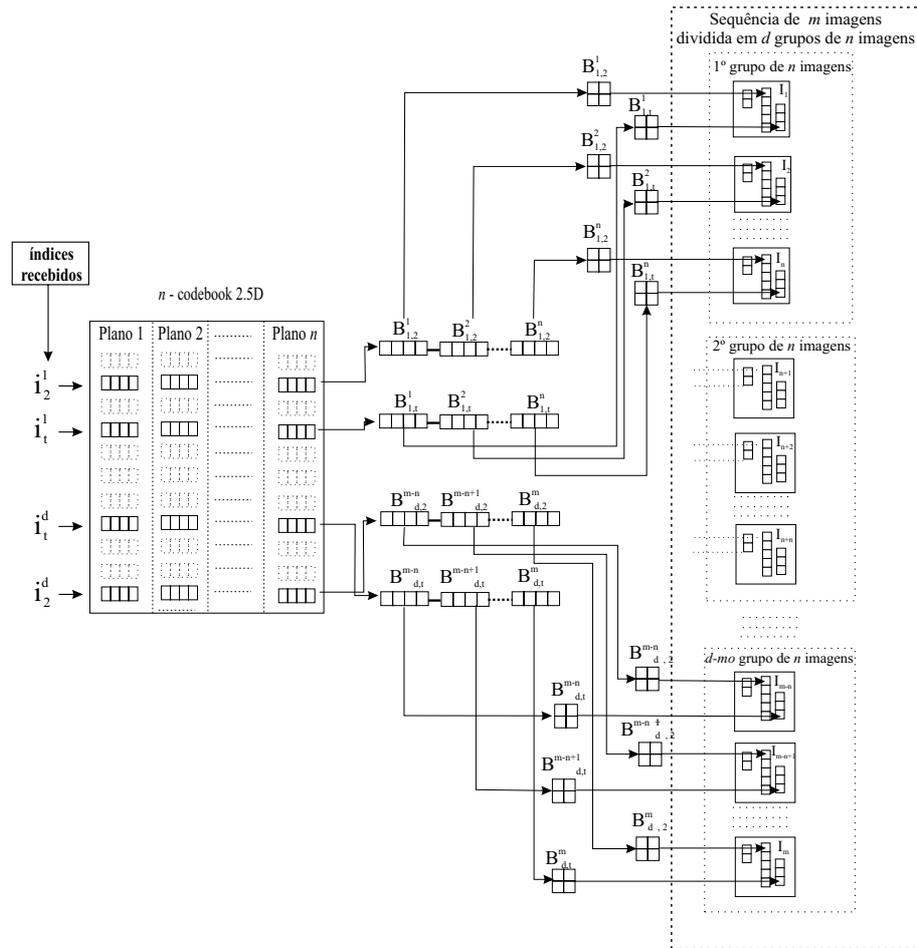


Figura 4.7: Descompressor multi-bloco baseado em n -codebook 2.5D

3. Considerar n blocos de imagem onde cada bloco pertence a uma imagem do grupo de n imagens;
4. No *codebook* com n planos, n -codebook 2.5D, seleccionar o índice que corresponda à entrada que, em simultâneo, minimize os n blocos de imagem;
5. Repetir os passos 3. e 4. até que todos os índices no n -codebook 2.5D referentes aos t blocos do grupo de n imagens, tenham sido determinados;
6. Repetir os passos anteriores até que tenham sido considerados todos os d grupos de n imagens em que se dividiu a sequência de m imagens;
7. Transmitir para o decodificador os índices atrás determinados.

- Fase de Descompressão:

1. Para cada índice recebido determinar, no n -*codebook* 2.5D, cada um dos n blocos correspondentes a cada uma das imagens do grupo de n imagens;
2. Reconstruir o bloco correspondente na imagem respectiva;
3. Repetir os passos 1. e 2. para todas as entradas respeitantes a cada um dos grupos de n imagens;
4. Repetir os passos anteriores para todos os d grupos de n imagens em que se dividiu a sequência de m imagens.

A fase de geração do n -*codebook* 2.5D é indicada como fase prévia, pois não é necessário gerar-se sempre este n -*codebook*. Dependendo de condições várias, este *codebook* pode ser utilizado mais do que uma vez, só sendo necessário ser novamente gerado se houver diferenças significativas nas imagens a comprimir. Efectuou-se um estudo sobre a frequência com que é necessário renovar os *codebooks*, sendo esse estudo referido na secção 5.6.

4.3.3 Algoritmo para Compressão VQ3DN

Como referido, dependendo do modo como o n -*codebook* é gerado pode-se ter uma compressão 2.5DN ou 3DN.

A diferença entre estas duas variantes reside essencialmente no facto de na compressão 3DN o *codebook* ser gerado considerando-se simultaneamente as n imagens de treino. Recorde-se que na compressão 2.5DN cada uma dessas imagens foi considerada isoladamente, treinando-se não um *codebook* com *codevectors* aumentados de n mas sim n *codebooks* disjuntos obtidos a partir de cada uma das n imagens de treino.

Deste modo, com o n -*codebook* assim gerado, também se podem comprimir em simultâneo n imagens.

Para se perceber o modo como é feita a determinação dos índices, aquando da fase de compressão, considere-se o esquema mostrado na figura 4.8 e considere-se também a figura 4.9. Os esquemas mostrados nestas figuras são similares aos

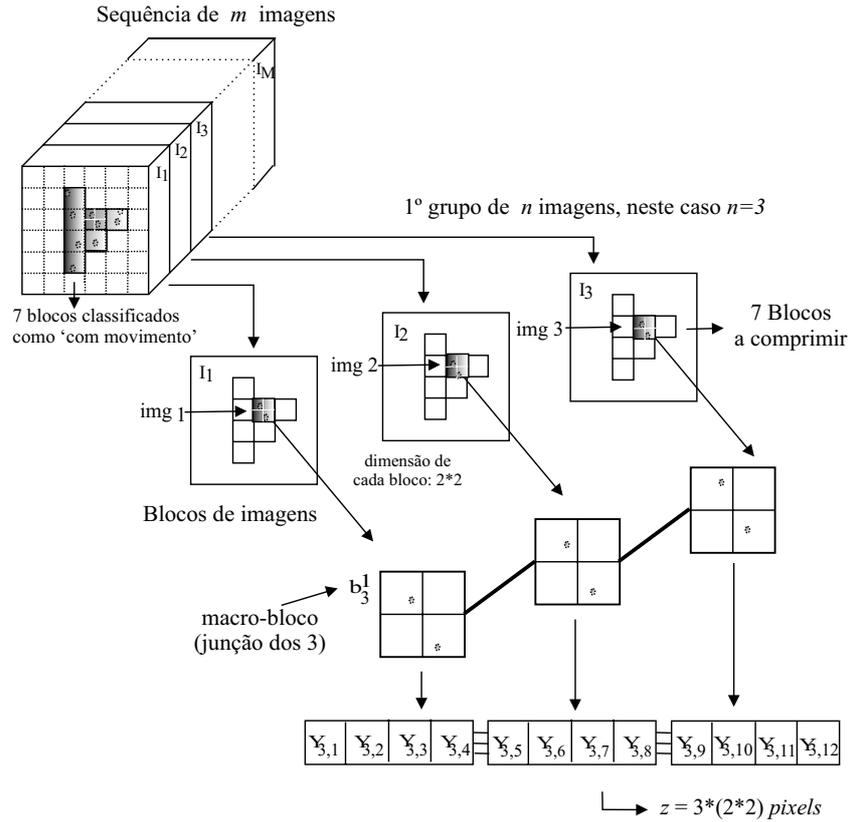


Figura 4.8: Disposição dos blocos b_t^g e dos *pixels* $Y_{t,z}$, $z = n(k \times k)$ na sequência. Variante VQ3DN

mostrados nas figuras 4.4 e 4.5, diferenciando-se pelo facto de agora cada *codevector* ter uma dimensão $z = n(k \times k)$, com $(k \times k)$ a dimensão espacial de cada bloco de imagem a comprimir. De igual modo, o vector de dados correspondente aos n blocos de imagem a comprimir em simultâneo terá dimensão $z = n(k \times k)$, onde $(k \times k)$ é a dimensão espacial de cada bloco de imagem, pois corresponderá a um *macro-bloco* resultante da junção dos n blocos individuais. Como não é preciso fazer a diferenciação entre os *codebooks*, os *codevectors* só serão referenciados por v_j , com j o número de entradas no *codebook*, tipicamente $j = 1, 2, \dots, 256$.

Relativamente aos dados dos blocos de imagem a comprimir, também serão considerados blocos b_t^g , onde $t = 1, \dots, l$ é o número de blocos classificados na fase de segmentação como sendo blocos 'com movimento' e g é o grupo de n imagens que se está a comprimir nesse instante, $g = 1, \dots, d$.

Conceptualmente, as operações a efectuar para a determinação dos índices são

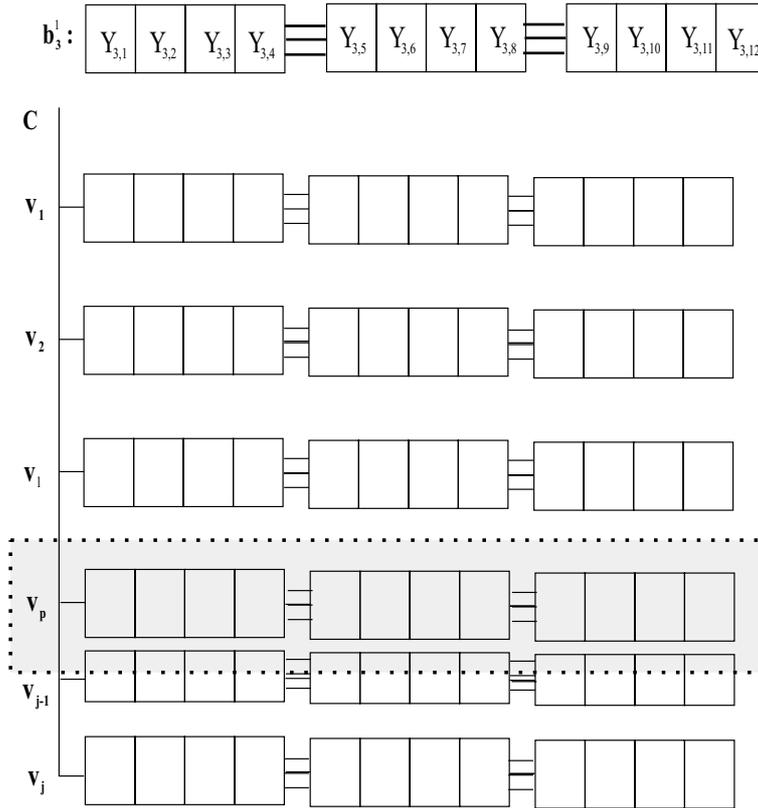


Figura 4.9: Determinação dos índices para os 3 blocos - Variante 3DN

similares às feitas para a compressão VQ2.5DN e consistem em pesquisar dentro do *codebook*, qual o *codevector* que contém os z escalares mais próximos aos z valores de cinzento do conjunto de n blocos a comprimir. Esta determinação é também feita por minimização de uma medida de distorção, baseada no erro quadrático médio, equações (4.3.1) e (4.3.2).

Para cada grupo g de n imagens, e considerando os blocos b_t^g como sendo os blocos classificados como tendo movimento, o índice do *codevector* a transmitir obtém-se por minimização de:

$$d(\underbrace{b_t^g}_{j=1,2,\dots,256}, v_j) = \sum_{r=1}^z (Y_{t,r} - \hat{Y}_{j,r})^2, \quad (4.3.4)$$

sendo a dimensão dos *codevectors* v_j igual a $z = n(k \times k)$. Os $\hat{Y}_{j,r}$ referem-se a escalares na posição r , dentro de cada *codevector* j .

Na figura 4.9, a zona a sombreado corresponderá ao índice do *codebook* que

eventualmente traduz a melhor escolha para este bloco de dados a comprimir. Este índice, assim determinado, é então transmitido ao decodificador, onde será utilizado para a reconstrução das imagens.

Com esta variante também se consegue, obviamente, referenciar com um só índice mais do que um bloco de imagens. Assim, as considerações relativas aos ganhos obtidos na taxa de compressão e referenciados na secção anterior também são aqui válidas.

Dado o facto de nesta variante de compressão, com quantização vectorial e n -codebooks, se ter considerado, na geração dos *codebooks*, dados de treino tridimensionais, que permitem treinar um *codebook* onde cada *codevector* reflecte um bloco tridimensional, designamos esta variante por VQ3DN.

4.3.4 Resumo do Algoritmo VQ3DN

Este algoritmo, que graficamente se representa nas figuras 4.10 e 4.11, é bastante similar ao anterior e tem os seguintes passos:

- Fase Prévia \iff Geração do n -codebook 3D;
- Fase de Compressão:
 1. Dividir a sequência de m imagens em d grupos de n imagens, $d = m/n$;
 2. Determinar em cada imagem, mas considerando todas as n imagens do grupo, os t blocos 'com movimento' usando filtros de Gabor \iff Etapa 1;
 3. Agrupar os n blocos, resultantes da segmentação de n imagens diferentes, num *macro-bloco* de dimensão $z = n(k \times k)$, sendo n o número de imagens simultâneas e $k \times k$ a dimensão espacial de cada bloco;
 4. No n -codebook, seleccionar o índice que corresponda à entrada que minimize o *macro-bloco*;
 5. Repetir o passo anterior, até que todos os índices do n -codebook referentes aos t *macro-blocos* da sequência de imagens tenham sido determinados;

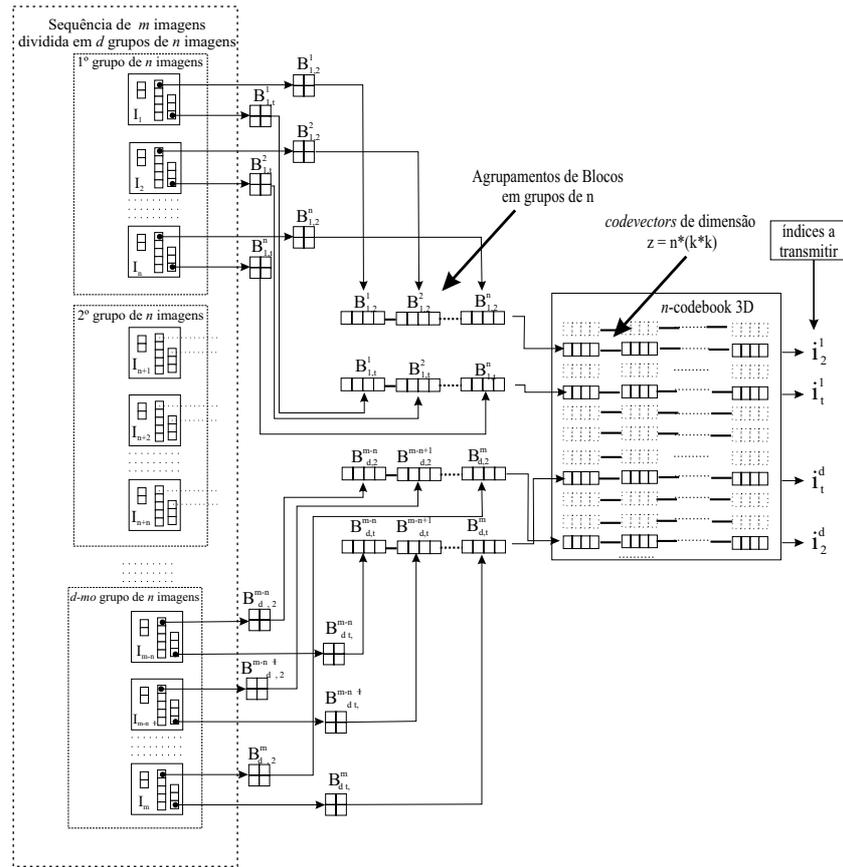


Figura 4.10: Compressor multi-bloco baseado em n -codebook 3D

6. Repetir os passos 3., 4. e 5. até que tenham sido considerados todos os d grupos de n imagens em que se dividiu a sequência de m imagens;
7. Transmitir para o decodificador os índices atrás determinados.

- Fase de Descompressão:

1. Para cada índice recebido, determinar no n -codebook cada um dos *codevectors* correspondentes a cada *macro-bloco* de imagens do grupo de n imagens;
2. Reconstruir os blocos correspondentes na imagem respectiva;
3. Repetir os passos 1. e 2. para todas as entradas respeitantes a cada um dos *macro-bloco* de n imagens;
4. Repetir os passos anteriores para todos os d grupos de n imagens em que se dividiu a sequência de m imagens.

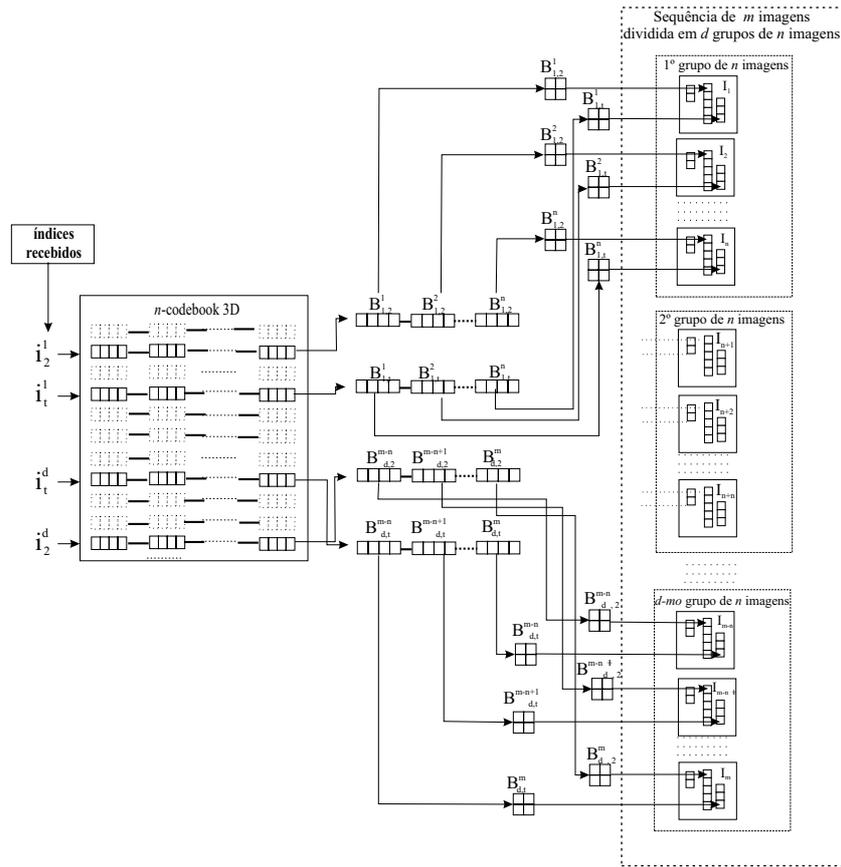


Figura 4.11: Descompressor multi-bloco baseado em n -codebook 3D

A fase de geração do n -codebook é indicada como fase prévia, pelas mesmas razões apontadas para o caso da VQ2.5DN. Também para este caso foi feito um estudo, que é referido na secção 5.6, sobre a estabilidade dos *codebooks*.

Na figura 4.10, é representado o bloco compressor deste algoritmo e na figura 4.11 faz-se a representação do respectivo bloco descompressor. Estes blocos são obviamente semelhantes aos mostrados nas figuras 4.6 e 4.7 para o caso VQ2.5DN.

4.3.5 Quantização Vectorial com n -codebooks versus Qualidade das Imagens Reconstruídas

O facto de se estar a fazer uma minimização de uma medida de distorção, baseada no erro quadrático médio, aplicada em simultâneo a n vectores diferentes, leva ao aumento da distorção total da imagem reconstruída. Este facto reflecte-se num abaixamento tanto da qualidade visual das imagens reconstruídas como da relação



a) Imagem original



b) Reconstrução com VQ3DN

Figura 4.12: Reconstrução da Sequência Açores 1 - Com VQ3DN



a) Imagem original



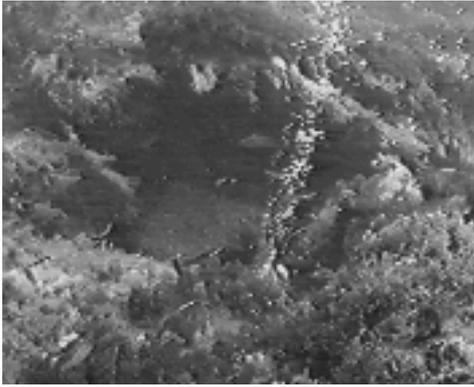
b) Reconstrução com VQ3DN

Figura 4.13: Reconstrução da Sequência Açores 3 - Com VQ3DN

sinale-ruído destas imagens, quando comparados com os valores que se podem obter na Quantização Vectorial clássica.

No entanto, para as imagens que foram usadas como teste e cujas reconstruções são mostradas nas figuras 4.12, 4.13 e 4.14, a qualidade visual apresenta-se perfeitamente aceitável. Este facto também se pode constatar da análise das tabelas referentes aos valores da relação sinal-ruído e apresentados na análise de resultados na secção 5.4.

Assim, apesar da distorção nas imagens reconstruídas ter aumentado permanece contudo dentro de limites aceitáveis. Os ganhos obtidos na taxa de compressão são bastante significativos e directamente proporcionais a n , sendo n o número de imagens a comprimir em simultâneo.



a) Imagem original



b) Reconstrução com VQ3DN

Figura 4.14: Reconstrução da Sequência Açores 5 - Com VQ3DN

Atendendo ao facto dos ganhos na taxa de compressão serem directamente proporcionais ao número de imagens que são comprimidas em simultâneo, poder-se-ia tentar aumentar significativamente este valor n , o número de imagens a comprimir em simultâneo, de modo a aumentar ainda mais a taxa de compressão. No entanto, o aumento do número de imagens a comprimir em simultâneo, se bem que aumente a taxa de compressão, faz aumentar também proporcionalmente a distorção total nas imagens reconstruídas, pelo que o número de imagens a comprimir em simultâneo não pode ser muito grande. Testes realizados permitem concluir que, comprimindo 3, 5, 10 ou 15 imagens em simultâneo, os resultados obtidos, quer de um ponto de vista subjectivo relacionado com a qualidade visual das imagens, quer de um ponto de vista baseado na medida do PSNR são bastante aceitáveis.

Para se ultrapassar esta dependência, aumento do número de imagens a comprimir em simultâneo *versus* aumento da distorção total nas imagens reconstruídas, foram desenvolvidas duas outras estratégias, descritas nas secções seguintes, para tentar aumentar o número de imagens a comprimir em simultâneo sem aumentar proporcionalmente a distorção nas imagens reconstruídas.

4.4 Compressão de Imagens com Quantização Vectorial com *i-codebooks*

Com esta abordagem, que assenta numa filosofia diferente da anterior, procurou-se também comprimir mais do que uma imagem em simultâneo.

Na abordagem anterior utilizaram-se *codebooks* de níveis de cinzento que, por conterem mais do que um plano, permitem a compressão em simultâneo de um número de blocos de imagens equivalente ao número de planos do *codebook*.

Aqui, utilizam-se dois tipos de *codebooks* diferentes: um *codebook* de níveis de cinzento e um *codebook* de índices. O primeiro servirá para, numa fase prévia, gerar o segundo. Depois, na fase de selecção dos índices, determina-se relativamente ao *codebook* de cinzentos um conjunto de n índices, n -pletos de índices do *codebook* de cinzentos, cada um deles referente a um dos n blocos de imagem, e depois referencia-se esse conjunto de índices relativamente ao *codebook* de índices. É a entrada correspondente a este último *codebook* que é transmitida ao decodificador e que é utilizada para a reconstrução das imagens.

Nesta abordagem, parte-se do principio de que as imagens que se têm de comprimir constituem um conjunto relativamente fechado de imagens. Por conjunto fechado de imagens quer-se significar não só que numa sequência de imagens a probabilidade de ocorrência de uma imagem com um elevado grau de diferenciação, relativamente às outras, é muito reduzida mas também que, pelo contrário, todas as imagens têm um grau de semelhança muito grande. As sequências de imagens submarinas, que serão utilizadas, foram obtidas com a câmara parada ou quase parada e para um fundo relativamente estático, onde a inovação se traduz no aparecimento de bolhas gasosas, pelo que se podem considerar como conjunto fechado de imagens.

É de esperar que, para um conjunto deste tipo, haja uma grande constância no valor dos índices do *codebook* de cinzentos, seleccionados para blocos que ocupem a mesma posição ao longo de várias imagens dentro da sequência. Isto é, se para a primeira imagem da sequência, o índice, no *codebook* de cinzentos, que melhor

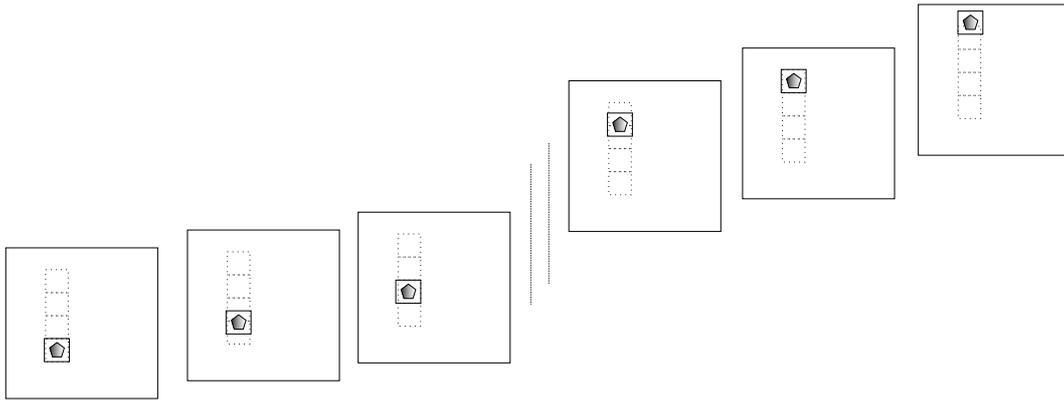


Figura 4.15: Movimento rectilíneo de um padrão

descreva o vector referente ao bloco b_k , for o índice $i = p$, será de esperar que o índice referente ao bloco que ocupe a mesma posição na imagem seguinte tenha sensivelmente o mesmo valor que o anterior, $i = p \pm \Delta p$. O valor Δp representa o ajuste que terá de ser feito ao valor inicial do índice para adaptá-lo ao novo bloco. Deste modo, é previsível que sejam referenciados, a menos de um Δ , os mesmos índices do *codebook* para os mesmos blocos dentro de uma sequência de imagens.

Por outro lado, se nas imagens a comprimir os movimentos tiverem direcções aproximadamente rectilíneas, como é o caso das colunas de bolhas nas imagens submarinas que se querem comprimir e enviar para a superfície, é também de esperar que os índices no *i-codebook*, para cada conjunto de imagens a comprimir em simultâneo, sejam quase iguais aos índices no *i-codebook* obtidos para estes blocos quando se deslocarem no sentido ascendente, como se mostra na figura 4.15, ou seja, pode haver uma repetição dos padrões ao longo dessa direcção ascendente. Deste modo, havendo esta repetição, o número de *n-pletos* do *i-codebook* necessários para codificar os blocos de imagens não terá de ser necessariamente muito grande.

4.4.1 Obtenção dos *codebooks*

Partindo destes pressupostos, pode-se pensar em determinar a partir de um conjunto alargado de imagens de treino, o conjunto de índices no *codebook* de cinzentos referenciados para cada $b_l^n, l = 1, \dots, m$, sendo m o número total de blocos em

que se dividiu cada imagem de treino e n o número da imagem a comprimir em simultâneo, figura 4.16. Cada um dos m blocos de imagem terá dimensão de $k \times k$ *pixels*. Assim, obtêm-se todos os índices referentes a M conjuntos de $m \times n$ blocos relativamente ao *codebook* de níveis de cinzento. Estes valores de índices são então usados para determinar um outro *codebook*, treinado a partir destes valores, que reflecta as combinações de n valores de índices com maior probabilidade de ocorrência. Designamos este segundo *codebook* por *i-codebook*. A dimensão do *i-codebook* será g , devendo g ser muito maior do que j , a dimensão do *codebook* de cinzentos, de modo a poder obter-se a maior variabilidade possível de combinações de índices do *codebook* de cinzentos.

O *codebook* de cinzentos é gerado a partir de imagens de treino usando o algoritmo LBG [10], enquanto que o *i-codebook* é obtido como se passa a descrever.

Sejam c_j , as entradas do *codebook* de cinzentos e sejam $j = 1, \dots, J$ os índices respectivos. Considerem-se agora T conjuntos de $n(k \times k)$ blocos de imagens arbitrárias. Para cada bloco de imagem $img_i, i = 1, \dots, n$ em cada conjunto, calculamos,

$$e_{ij}^2 = \|c_j - img_i\|_F^2, i = 1, \dots, n; j = 1, \dots, J. \quad (4.4.1)$$

O índice que representa a $n(k \times k)$ imagem arbitrária no *codebook* de cinzentos é obtido resolvendo o problema de minimização,

$$p_i = \arg \min_j \{e_{ij}^2\}_{j=1}^J, i = 1, \dots, n. \quad (4.4.2)$$

Repetindo este procedimento para todos os T conjuntos, obtemos o conjunto

$$\{[p_1, \dots, p_n]_{l=1}^T\}, \quad (4.4.3)$$

que será o conjunto de treino do *i-codebook*. Este é gerado usando-se o algoritmo LBG [10], onde estes T conjuntos de índices funcionam como "imagens".

Na figura 4.16, exemplifica-se, para um grupo de $n = 3$ imagens de uma sequência, a obtenção deste conjunto de índices para os blocos 1 e 25. Estando cada imagem particionada em m blocos, o cardinal deste conjunto de treino será $\#M = m \times n$.

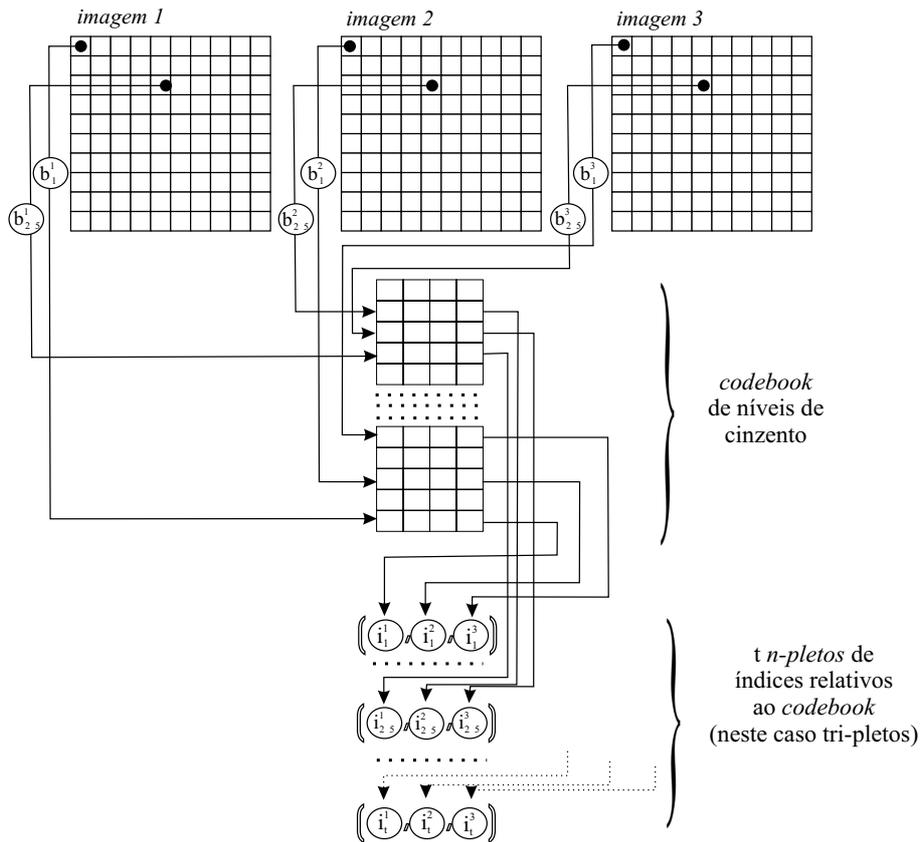


Figura 4.16: Exemplo de obtenção dos índices para a determinação do *i-codebook*

Neste exemplo consideraram-se $n = 3$ imagens e como cada imagem contém $m = 100$ blocos ter-se-à $\#M = 100 \times 3 = 300$. Em situações reais, o cardinal deste conjunto de índices será bastante maior, pois ter-se-ão muitos mais blocos por imagem.

Concretamente para as imagens que estamos a usar, cuja dimensão é QCIF, 144×176 , se considerarmos blocos $(k \times k) = 2 \times 2$, teríamos 6.336 blocos por imagem. No entanto, este número pode ser aumentado de modo a aumentar também a variabilidade de combinações de índices. Para isso, basta usar imagens de treino de dimensão maior, as quais podem ser obtidas, por exemplo, pela concatenação de outras mais pequenas. Cada um destes *n-pletos* de índices, no exemplo *tri-pletos*, deverão, pelas razões já apontadas, variar pouco relativamente às outras imagens da sequência que não as usadas para a sua determinação. Então, de um modo análogo ao descrito no capítulo 2 para a obtenção dos *codebooks* dos níveis de cinzento de uma imagem, pode-se pensar em obter a partir deste conjunto de índices, um *codebook* constituído por *n-pletos* de números, correspondentes aos índices das n imagens

usadas para a sua determinação.

Os *codebooks* obtidos a partir dos índices serão designados por *i-codebooks* para, por um lado, salientar o facto de serem constituídos por *n-pletos* de índices e, por outro, para distingui-los dos *n-codebooks* já referidos na secção anterior.

Estes *codebooks* de índices, desde que obtidos a partir de um conjunto de imagens representativas e divididas num número *m* suficientemente grande de blocos, e desde que contenham um número *g* relativamente elevado de *codevectors* de índices, *n-pletos* de índices, com $m \gg g$, permitirão obter uma boa referenciação dos níveis de cinzento.

Com esta técnica, o índice que é transmitido ao decodificador é o índice correspondente à entrada no *codebook* de índices. O decodificador, conhecedor do *i-codebook* e do *codebook* de cinzentos, começará por consultar o *i-codebook* de modo a obter o *n-pleto* de valores, cada um dos quais correspondente a uma entrada no *codebook* de cinzentos e correspondendo também a cada um dos *n* blocos de imagens codificados em simultâneo. Depois, com base nestes valores, consulta o *codebook* de cinzentos onde obtém os valores de cinzento apropriados para a reconstrução de cada *pixel* da imagem.

Como referimos, o *i-codebook* permite codificar grupos de *n* índices, sendo *n* o número de $(k \times k)$ blocos de imagens a comprimir em simultâneo. O *codebook* de níveis de cinzento terá *J* entradas, cada uma de dimensão $(k \times k)$ e o *i-codebook* terá *g* entradas, cada uma de dimensão *n*.

4.4.2 Algoritmo para Compressão VQIND

Obtidos os *codebooks*, quer o de cinzentos quer o de índices, faz-se primeiro a determinação para cada um dos blocos classificados como 'com movimento' dos seus índices no *codebook* de cinzentos obtendo-se um grupo de *n* índices. Depois determina-se no *i-codebook* o índice do *codevector* que melhor descreva este grupo de índices e transmite-se de seguida essa informação ao decodificador.

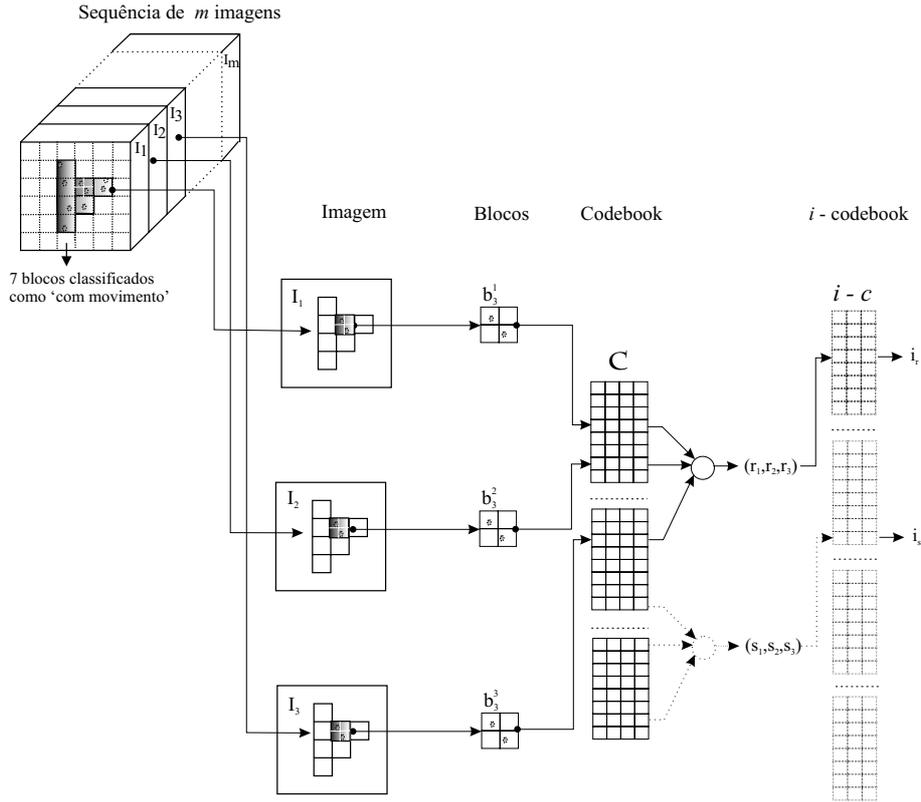


Figura 4.17: Codificador com i -codebook

Descreve-se a seguir esta determinação dos índices do i -codebook, referente a cada conjunto de n blocos de imagens classificado como 'com movimento' na segmentação.

Seja $N_i, i = 1, \dots, n$ o índice do codebook de cinzentos referente ao n -ésimo bloco de imagem. Este índice é obtido do codebook de cinzentos a partir de:

$$N_i = \arg \min_j \{e_{ij}^2\}_{j=1}^J, i = 1, \dots, n. \quad (4.4.4)$$

Para isso calcula-se $e_{i,j}^2$ considerando-se para cada bloco de imagem img_i e para os c_j codevectors do codebook de cinzentos:

$$e_{ij}^2 = \|c_j - img_i\|_F^2, i = 1, \dots, n; j = 1, \dots, J. \quad (4.4.5)$$

Deste modo obtém-se um conjunto de entradas no codebook de cinzentos respeitantes aos n blocos de imagem. De seguida, indicam-se as operações necessárias para se determinar a entrada no i -codebook correspondente a este conjunto de índices do codebook de cinzentos.

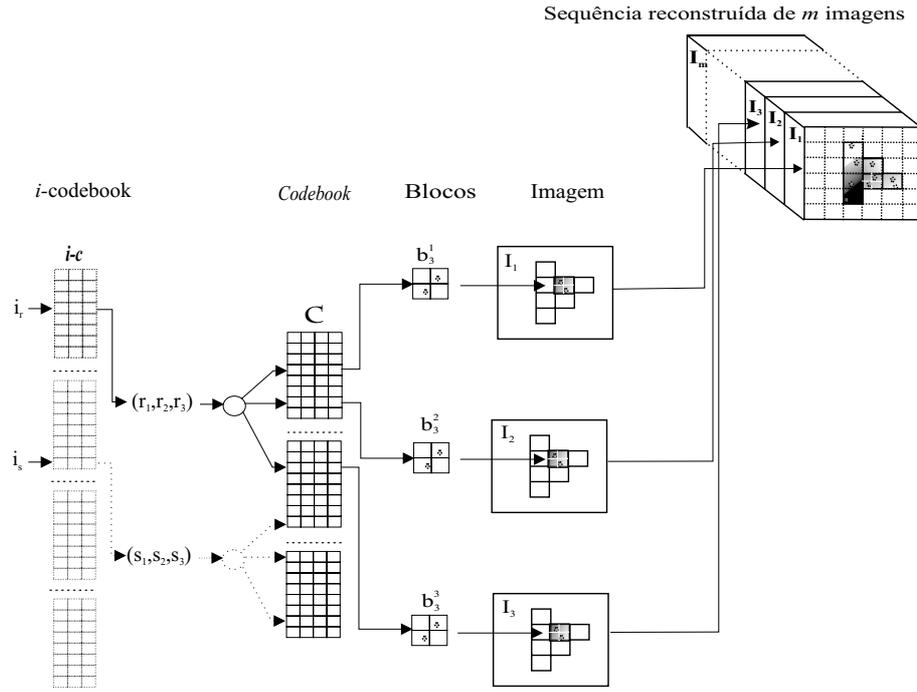


Figura 4.18: Descodificador com *i-codebook*

Tendo-se obtido este grupo de N_i índices do *codebook* de cinzentos para os n blocos de imagem e designando-o por N_{ind} , determina-se de seguida, no *i-codebook*, a entrada que melhor se aproxime de N_{ind} .

Seja I cada um desses índices do *i-codebook*. I obtém-se a partir de

$$I = \arg \min_g \{e_g^2\}_{g=1}^G, \quad (4.4.6)$$

onde

$$e_g^2 = \|c_g - N_{ind}\|_F^2, g = 1, \dots, G, \quad (4.4.7)$$

e $c_g, g = 1, \dots, G$ são os *codevectors* do *i-codebook*.

Na figura 4.17 descreve-se, graficamente, esta determinação do índice no *i-codebook* de 3 blocos diferentes a partir do *codebook* de cinzentos e, na figura 4.18, a respectiva descodificação do lado do receptor.

No exemplo mostrado na figura 4.17, simulamos a codificação do terceiro bloco de cada uma das três imagens I_1, I_2 e I_3 , respectivamente os blocos b_3^1, b_3^2 e b_3^3 .

Feita a selecção no *codebook* de níveis de cinzentos, com base na minimização de

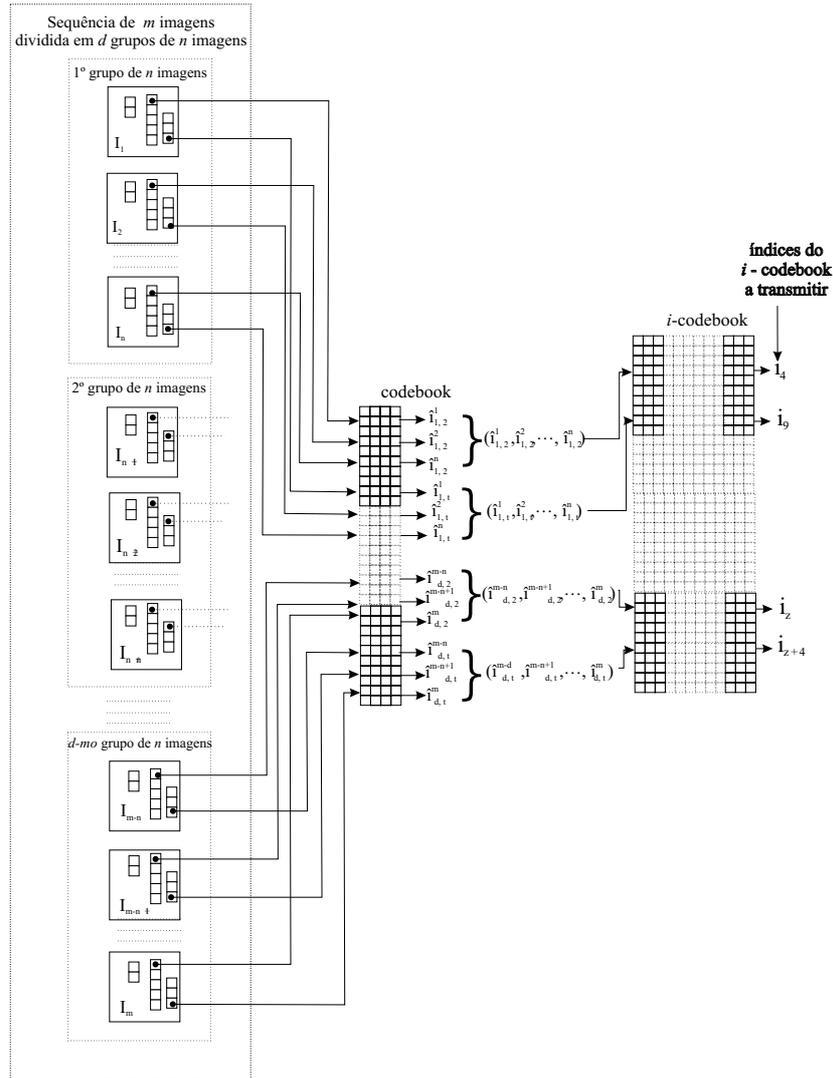


Figura 4.19: Compressor multi-bloco baseado em VQIND

uma medida de distorção, similar à indicada na equação (4.4.4), obteve-se para a primeira imagem o índice r_1 , para a segunda o índice r_2 e para a terceira o índice r_3 .

Depois, no i -codebook obtém-se, com base na minimização da medida de distorção indicada na equação (4.4.6), o índice deste codebook que melhor se aproxima do *tripleto* de índices (r_1, r_2, r_3) , tendo-se obtido o valor de índice i_r . Transmitiu-se este índice para o decodificador e reconstruíram-se as imagens executando as operações inversas correspondentes, figura 4.18.

A codificação de todos os outros b_t^n blocos de imagem, onde n é o número de imagens a comprimir em simultâneo e t o t -ésimo bloco classificado como 'com

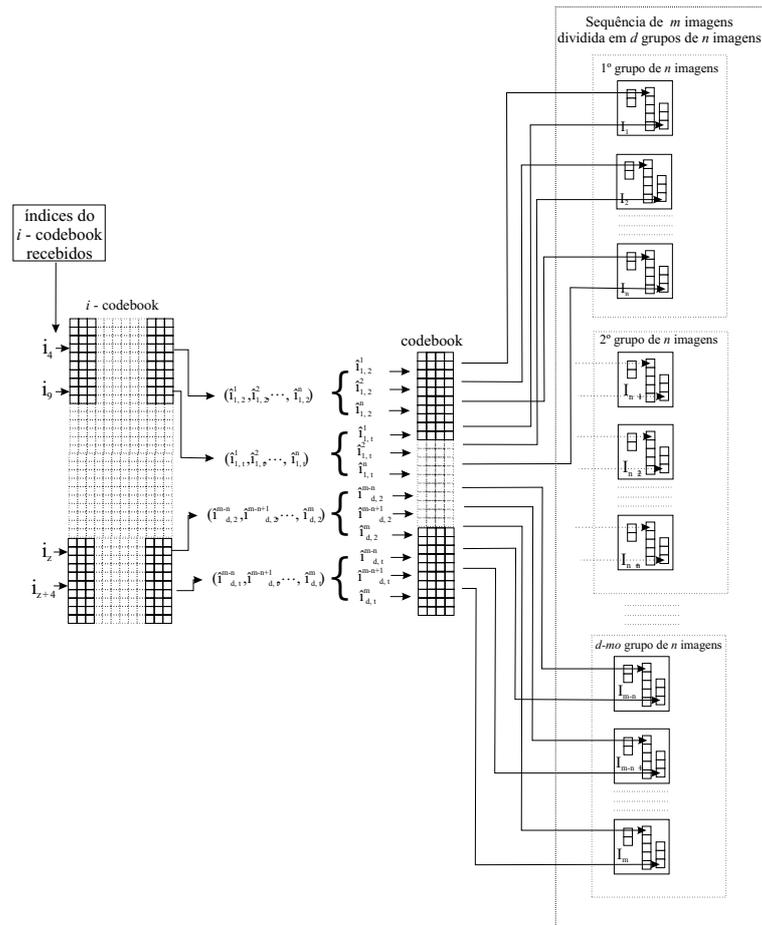


Figura 4.20: Descompressor multi-bloco baseado em VQIND

movimento' na fase de segmentação, seria feita de um modo similar.

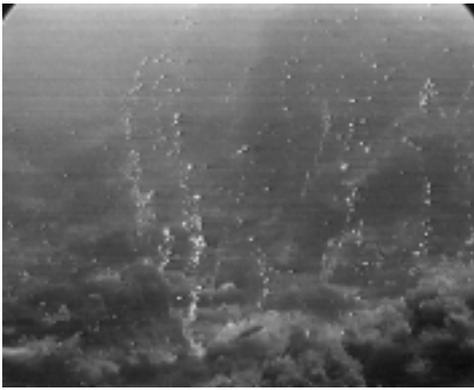
4.4.3 Resumo do Algoritmo VQIND

Na figura 4.19 mostram-se, graficamente, os passos deste algoritmo para o bloco compressor e na figura 4.20 o respectivo bloco descompressor.

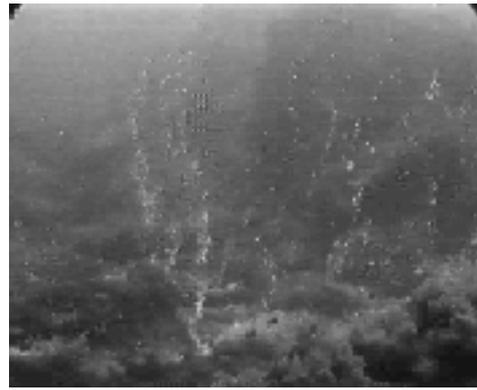
O algoritmo para a compressão baseada neste método resume-se a:

- Fase Prévia
 1. \iff Geração do *codebook* de cinzentos;
 2. \iff Geração do *i-codebook*;
- Fase de Compressão:

1. Dividir a sequência de m imagens em d grupos de n imagens, $d = m/n$;
 2. Determinar em cada imagem, mas considerando todas as n imagens do grupo, os t blocos 'com movimento' usando filtros de Gabor \Leftrightarrow Etapa 1;
 3. Para cada grupo de n blocos, determinar o n -pleto de índices no *codebook* de níveis de cinzento;
 4. Determinar no *codebook* de índices, *i-codebook*, as entradas que melhor se aproximem desse n -pleto de índices atrás determinados;
 5. Repetir os passos 3. e 4. até se determinarem todos os n -pletos de índices no *codebook* de cinzentos e o respectivo índice no *i-codebook*, referentes a todos os t blocos classificados como 'com movimento' nesse grupo de imagens;
 6. Repetir os passos anteriores até que tenham sido considerados todos os d grupos de n imagens em que se dividiu a sequência de m imagens;
 7. Transmitir para o decodificador os índices do *i-codebook* atrás determinados.
- Fase de Descompressão:
 1. Determinar no *i-codebook* as correspondências em termos de índices do *codebook* de cinzentos, respeitantes a cada uma das entradas do *i-codebook* recebidas do codificador e respeitantes a cada bloco pertencente a cada um dos n -pletos de imagens em que se dividiu a sequência;
 2. Para cada uma das correspondências encontradas em 1. reconstruir o bloco da imagem respectiva, por leitura no *codebook* de cinzentos do *codevector* correspondente a esse índice;
 3. Repetir os passos 1. e 2. para todas as entradas, respeitantes a todos os blocos de cada grupo de imagens;
 4. Repetir os passos anteriores até que todos os d grupos de n imagens em que se dividiu a sequência de m imagens tenham sido reconstruídos.



a) Imagem original



b) Reconstrução com VQIND

Figura 4.21: Reconstrução da Sequência Açores 1 - Com VQIND



a) Imagem original



b) Reconstrução com VQIND

Figura 4.22: Reconstrução da Sequência Açores 3 - Com VQIND

4.4.4 Quantização Vectorial com *i-codebooks* de Índices *versus* Qualidade das Imagens Reconstruídas

Relativamente ao método anterior, o aparente *overhead* que poderíamos ser levados a pensar existir, pela consideração de dois *codebooks* diferentes, o de níveis de cinzento e o de índices, na realidade não é significativo devido ao facto de aqui o *codebook* de cinzentos ser apenas constituído por um plano.

Por outro lado, suponhamos que após a transmissão de f imagens, $f \gg 1$, a qualidade das imagens reconstruídas é muito baixa. Poderá então ser necessário transmitir-se novos *codebooks* gerados com imagens mais recentes. No entanto, em vez de transmitir um novo *codebook* de cinzentos pode ser suficiente transmitir apenas um novo *i-codebook*, construído com base no *codebook* de cinzentos corrente e num

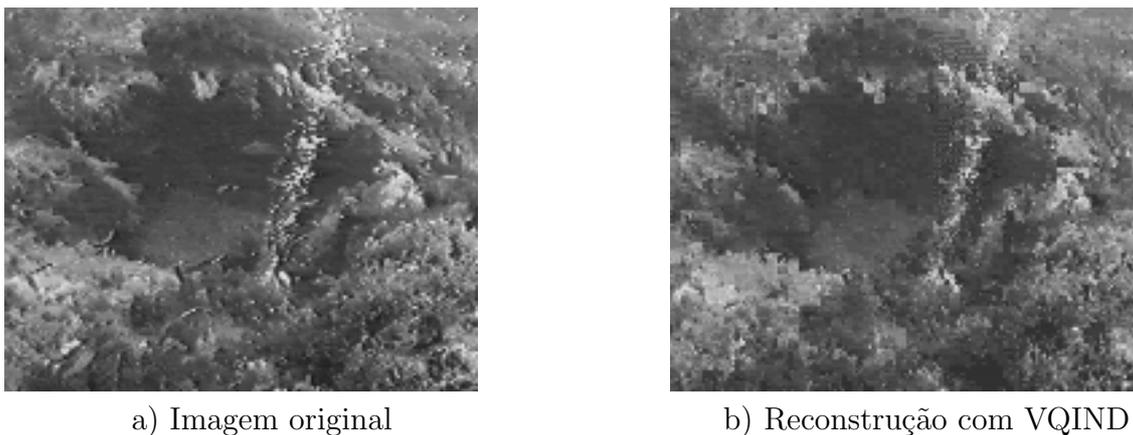


Figura 4.23: Reconstrução da Sequência Açores 5 - Com VQIND

novo conjunto alargado de imagens.

Este método também pode introduzir distorção. De facto, tendo o *i-codebook* um tamanho limitado, não é possível conter todas as combinações (r_1, r_2, r_3) referentes aos índices no *codebook* de cinzentos e portanto pode escolher-se um índice diferente do ideal. No entanto, este erro pode ser minimizado se o *codebook* de cinzentos estiver ordenado de alguma maneira. Em tal caso, se por exemplo o r_1 correcto for $r_1 = 13$ e no *i-codebook* a melhor escolha recair sobre o índice correspondente ao índice do *codebook* de cinzentos $r_1 = 10$, esse erro será menor se o *codebook* de cinzentos estiver ordenado, pois o *codevector* do *codebook* de cinzentos correspondente ao índice 10 conterà escalares com valores mais próximos aos do *codevector* correspondente ao índice 13; caso não existisse essa ordenação os valores de um e de outro destes dois *codevectors* poderiam ser muito díspares. Nas simulações efectuadas, os *codevectors* do *codebook* de cinzentos foram ordenados com base na norma do vector constituinte de cada *codevector*. Desse modo, é de esperar que eventuais inexactidões na determinação dos índices, referentes ao *codebook*, sejam minimizadas.

Com a utilização do *i-codebook* é óbvio que a degradação aumenta com o aumento do número de imagens a comprimir em simultâneo. Nos testes que se efectuaram, conseguiu-se comprimir 3, 5, 10 e 15 imagens em simultâneo, mantendo uma razoável qualidade visual e valores aceitáveis para a relação sinal-ruído.

Como se pode constatar, as imagens das figuras 4.21, 4.22 e 4.23 e os dados

apresentados na secção 5.4 mostram que os resultados obtidos com a utilização desta técnica também são bastante bons, quer relativamente à qualidade visual, quer relativamente à relação sinal-ruído.

4.4.5 Escalabilidade da Compressão

Também é possível, através da utilização de *i-codebooks*, fazer-se uma escalabilidade da compressão, isto é, pode-se variar o número de imagens a comprimir em simultâneo, quer para melhorar a qualidade das imagens reconstruídas, quer para se fazer a adaptação ao débito máximo de transmissão possível num dado instante.

Para isso, o descompressor pode emitir um comando que instrua o compressor para aumentar ou para reduzir o número de imagens a comprimir em simultâneo e desse modo possibilitar um aumento da qualidade das imagens reconstruídas ou aumentar/baixar o débito no canal de transmissão. Note-se que isto pode ser feito, sem alterar quer o *codebook* de níveis de cinzentos quer o *i-codebook*, desde que o número máximo desejado de imagens a comprimir em simultâneo não seja superior ao número máximo de imagens passível de ser comprimido por um determinado *i-codebook*. Assim, se o *i-codebook* foi gerado para comprimir n imagens é possível com esse *i-codebook* comprimir n ou menos imagens de cada vez mas nunca mais do que n imagens.

Por exemplo, se tivermos um *i-codebook* com o qual seja possível comprimir em simultâneo no máximo 15 imagens, esse mesmo *i-codebook* pode ser utilizado para comprimir, por exemplo, 3 imagens em simultâneo. Neste caso, na pesquisa no *i-codebook* basta considerar somente os 3 primeiros valores para a minimização da medida de distância ao vector de dados. A qualidade da imagem pode aumentar, por exemplo, ao passar-se da codificação simultânea de 15 para 3 imagens, porque na minimização da medida de distorção podem-se obter valores mais aproximados.

4.5 Compressão de Imagens com Quantização Vectorial e Lempel-Ziv

4.5.1 Algoritmo para Compressão VQZIP

Para efeitos de comparação com as estratégias desenvolvidas utilizamos Quantização Vectorial multibloco, recorrendo-se a um *codebook* de cinzentos, seguida da aplicação do algoritmo de compressão por dicionário adaptativo de Lempel-Ziv para a compressão dos índices resultantes dessa quantização.

Tal como para a técnica descrita na secção anterior, onde se utilizaram *i-codebooks* em conjugação com *codebooks* de níveis de cinzentos, nesta técnica começa-se por codificar separadamente cada bloco de imagem classificado como 'com movimento'.

Suponhamos que se pretende comprimir em simultâneo n blocos de imagens. Para simplificar a exposição considere-se $n = 3$. Começa-se então por determinar, no *codebook* de cinzentos, qual o índice que representa cada um dos blocos de imagem. Para esta determinação do índice faz-se a minimização de uma medida de distância entre cada um dos *codevectors* e o bloco a codificar, recorrendo-se por exemplo ao método dos mínimos quadrados.

Do grupo das 3 imagens que se quer comprimir em simultâneo, determinam-se esses índices para todos os blocos de imagem. O conjunto de índices assim construído é de seguida comprimido recorrendo-se a um dicionário adaptativo. Os índices comprimidos são transmitidos para o decodificador.

Do lado do decodificador procede-se às operações inversas e reconstroem-se as imagens. Começa-se por descomprimir os índices de modo a reconstituir os índices do *codebook* correspondente a todos os blocos de imagens do conjunto que estamos a comprimir em simultâneo. Com esses índices, consultando o *codebook* de cinzento, reconstroem-se os blocos correspondentes.

Nas figuras 4.24 e 4.25 representam-se, graficamente, as operações necessárias para se fazer, *a posteriori*, a compressão com Lempel-Ziv dos índices obtidos do

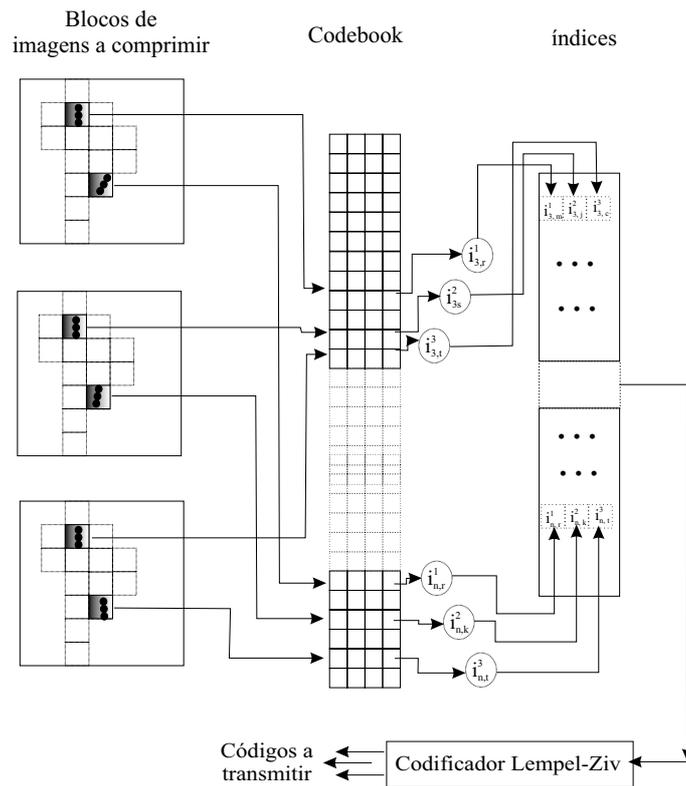


Figura 4.24: Codificador com VQ e Lempel-Ziv

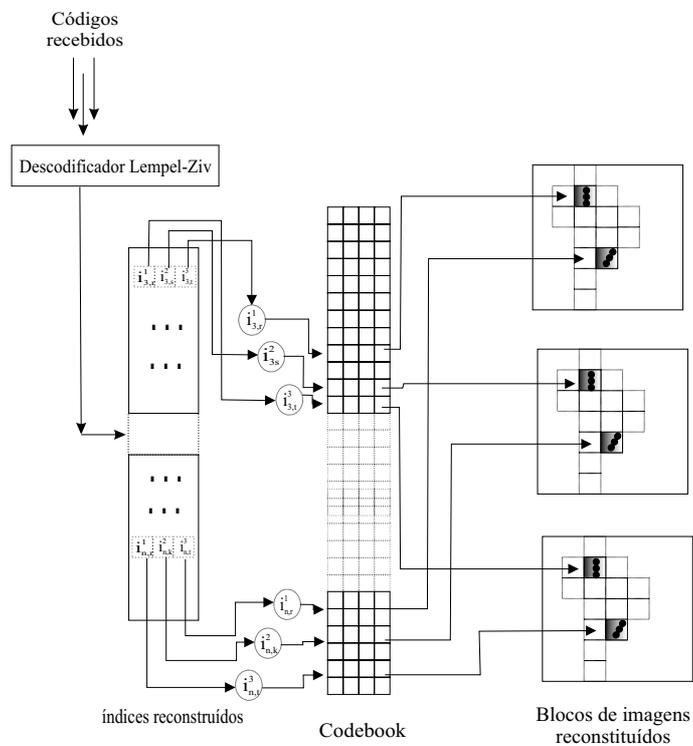


Figura 4.25: Descodificador com VQ e Lempel-Ziv

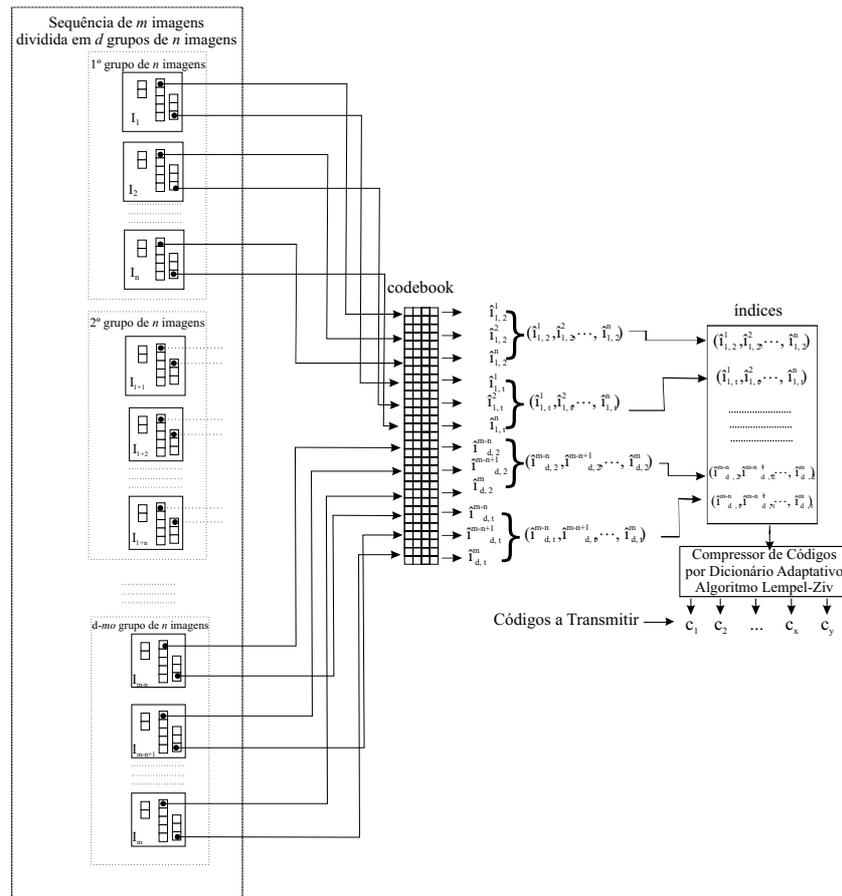


Figura 4.26: Compressor multi-bloco baseado em VQZIP

codebook, respectivamente do lado do codificador e do lado do decodificador. Neste exemplo, usou-se $n = 3$ imagens cujos blocos 'com movimento' se quer codificar em simultâneo.

Tal como na técnica apresentada na secção anterior, a que aqui se considerou também só necessita de um *codebook* de níveis de cinzento com um só plano.

Designou-se esta técnica onde se utiliza compressão com Quantização Vectorial e Lempel-Ziv por VQZIP.

4.5.2 Resumo do Algoritmo VQZIP

Na figura 4.26 representam-se, graficamente, os passos deste algoritmo para o bloco compressor e na figura 4.27 o respectivo bloco decompressor.

Para esta técnica, o algoritmo proposto terá então os seguintes passos principais:

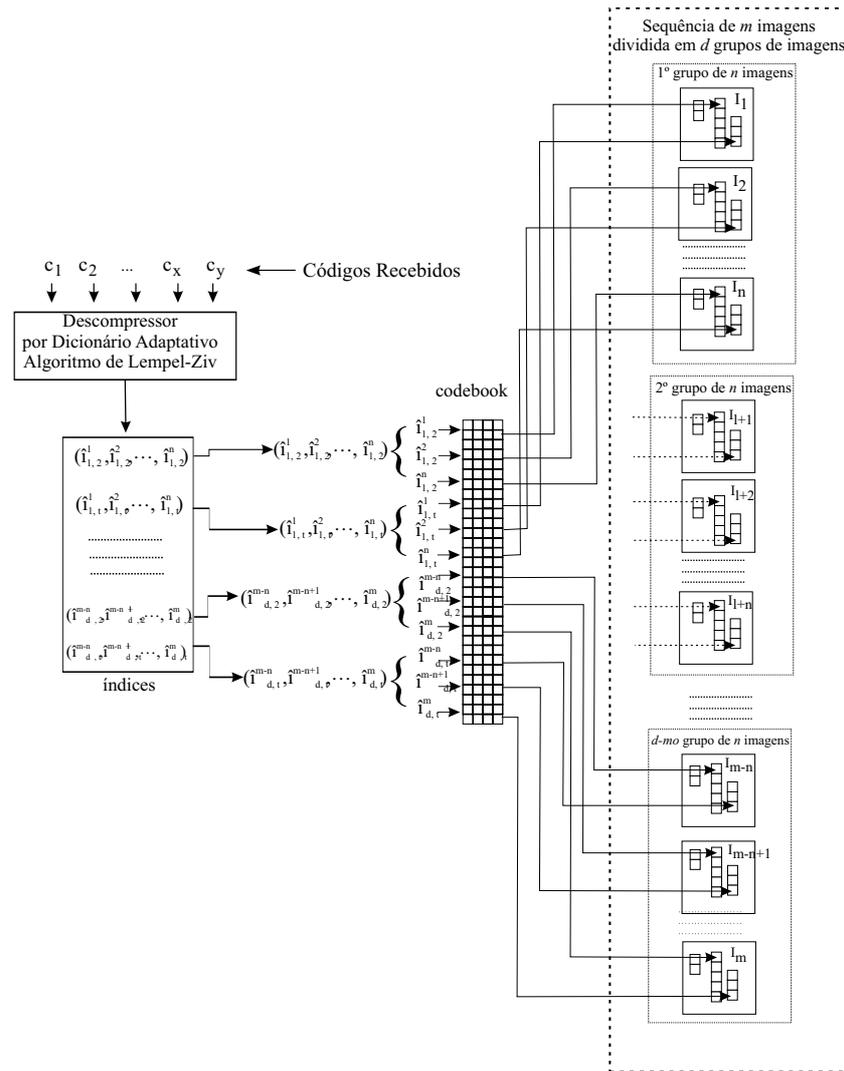


Figura 4.27: Descompressor multi-bloco baseado em VQZIP

- Fase Prévia \iff Geração do *codebook* de cinzentos;
- Fase de Compressão:
 1. Dividir a sequência de m imagens em d grupos de n imagens, $d = m/n$;
 2. Determinar em cada imagem, mas considerando todas as imagens do grupo, os t blocos com movimento usando para isso filtros de Gabor \iff Etapa 1;
 3. Para cada conjunto de n blocos, determinar o conjunto de n índices no *codebook* de níveis de cinzentos;

4. Repetir o passo 3. para todos os t blocos classificados como 'com movimento';
5. Repetir os passos 3. e 4. para todos os d grupos de n imagens da sequência;
6. Dispor todos os grupos de n índices e comprimi-los com o algoritmo de dicionário adaptativo de Lempel-Ziv;
7. Transmitir para o decodificador essa informação comprimida.

- Fase de Descompressão:

1. Aplicar o algoritmo de Lempel-Ziv para descomprimir os índices, de modo a obter as referências das entradas no *codebook* de cinzentos;
2. Para cada um dos índices, reconstruir o bloco respectivo, na imagem correspondente, por consulta do *codebook* e de modo a reconstruir as m imagens da sequência.

4.5.3 Quantização Vectorial com Lempel-Ziv *versus* Qualidade das Imagens Reconstruídas

As imagens assim reconstruídas também têm distorção. No entanto, esta distorção só é introduzida pela quantização vectorial e não pela compressão Lempel-Ziv, a qual não induz perda de informação. Por outro lado, as taxas de compressão máximas são bastante inferiores às obtidas com os métodos anteriores. Assim, a qualidade das imagens reconstruídas é superior à obtida com os outros métodos.

Nas figuras 4.28, 4.29 e 4.30 apresentam-se alguns exemplos de imagens reconstruídas com base nesta técnica.

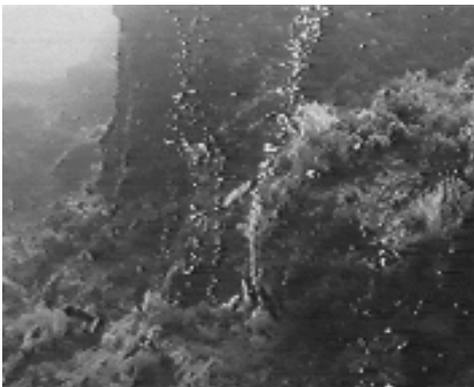


a) Imagem original



b) Reconstrução com VQZIP

Figura 4.28: Reconstrução da Sequência Açores 1 - Com VQZIP

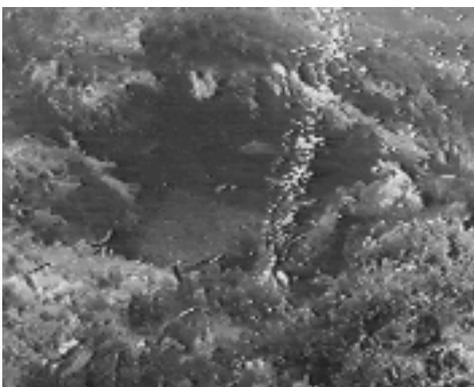


a) Imagem original



b) Reconstrução com VQZIP

Figura 4.29: Reconstrução da Sequência Açores 3 - Com VQZIP



a) Imagem original



b) Reconstrução com VQZIP

Figura 4.30: Reconstrução da Sequência Açores 5 - Com VQZIP

4.6 Sumário

Neste capítulo, explicitaram-se as técnicas propostas para a segunda fase do processo de compressão de vídeo submarino, que correspondem a comprimir os blocos classificados como 'com movimento', determinados na primeira fase e enviar, ao decodificador, a informação necessária à sua reconstrução. Com essa finalidade, foram desenvolvidas várias alternativas. Apresentaram-se ainda resultados que permitem uma análise qualitativa do desempenho das técnicas propostas.

Na primeira técnica geram-se *n-codebooks*, os quais são *codebooks* de níveis de cinzento com n planos e que permitem, exactamente devido a esta característica, comprimir em simultâneo n blocos de imagens, um por cada plano do *n-codebook*. Os n blocos são aproximados, por minimização de uma medida de distância, a *codevectors* do *n-codebook* e o respectivo índice é transmitido ao decodificador. De seguida, recebido este índice, consulta-se o *n-codebook* nessa posição e reconstroem-se em simultâneo os n blocos de imagem. Procede-se de igual modo para todos os blocos da sequência de imagens. Desenvolveram-se duas variantes a esta técnica as quais se diferenciam, essencialmente, pelo modo como os *n-codebooks* são gerados: ou são gerados n *codebooks*, um de cada vez, considerando-se as imagens de treino isoladamente, ou então gera-se um só *codebook* com *codevectors* de dimensão aumentada de n , utilizando-se simultaneamente para isso as n imagens de treino.

Na segunda técnica desenvolvida, para além do *codebook* de níveis de cinzento, que aqui terá um só plano, utilizou-se também um *codebook* de índices de entradas no *codebook* de cinzentos, o *i-codebook*, o qual terá *codevectors* de dimensão n , sendo n o número de blocos de imagens a comprimir em simultâneo. Para se comprimir em simultâneo n blocos de imagens determina-se, para cada bloco, no *codebook* de níveis de cinzento o índice correspondente. Para os n índices assim obtidos determina-se no *i-codebook*, com base na minimização de uma medida de distância entre o vector constituído pelos n índices e cada um dos *codevectors* do *i-codebook*, a entrada que melhor se aproxime desse vector de n índices. Transmite-se o índice referente a esta

entrada ao decodificador e este, efectuando as operações inversas correspondentes, reconstrói de uma só vez os n blocos de imagens. Isto é repetido até todos os blocos da sequência de imagens terem sido transmitidos e reconstruídas as imagens.

Na última técnica desenvolvida, utilizou-se um *codebook* de níveis de cinzento e os índices obtidos nesse *codebook*, para cada um dos blocos da imagem, são agrupados e sobre esses grupos aplica-se compressão baseada no algoritmo de dicionário adaptativo de Lempel-Ziv.

No capítulo seguinte, será feita a análise dos resultados obtidos com a aplicação dos algoritmos propostos, considerando as várias alternativas e a sua aplicação a sequências de imagens onde se regista actividade hidrotermal manifestada pela presença de colunas ascendentes de bolhas gasosas.

Capítulo 5

Simulações e Análise de Resultados

5.1 Introdução

De modo a validar resultados previstos para a utilização específica aqui considerada, desenvolveu-se uma aplicação informática que implementa os algoritmos propostos e através da qual é possível proceder a simulações computacionais. Estas visam, nomeadamente: verificar, se para imagens de interesse, é feita uma correcta e efectiva segmentação entre as zonas que contenham bolhas gasosas e as outras zonas, o *background*; verificar se é ou não obtido um ganho significativo, em termos de taxa de compressão resultante.

Em geral, os resultados obtidos são plenamente satisfatórios na medida em que se verificou uma segmentação correcta das imagens, obtendo-se também altas taxas de compressão. Conclui-se assim, ser possível obter uma massiva compressão das sequências de vídeo, o que torna possível a sua transmissão através do canal acústico submarino.

Apesar das altas taxas de compressão atingidas, a qualidade das sequências reconstruídas é bastante boa, quer quando se faz uma avaliação subjectiva, atendendo ao aspecto visual das imagens, quer quando se faz uma avaliação mais objectiva.

Para se ter uma ideia do comportamento dos algoritmos propostos quando aplicados a outros tipos de imagens, que não imagens submarinas, usaram-se nomeadamente sequências padrão e sequências de tráfego.

Os resultados podem ser analisados segundo várias perspectivas como, por exemplo: o tipo de sequência de imagens, o número de imagens a comprimir em simultâneo e a alternativa de quantização vectorial utilizada. Os resultados desta análise são apresentados em tabelas comparativas focando-se, por exemplo, a qualidade visual *versus* taxa de compressão; relação sinal-ruído *versus* taxa de compressão, etc..

Também se apresenta um estudo sobre a estabilidade dos *codebooks* e das imagens de referência, analisando-se qual a frequência de refrescamento necessária por forma a manter a qualidade das imagens reconstruídas.

Por último, refira-se que, para não mascarar de algum modo os resultados obtidos, não se fez qualquer pré ou pós-processamento das imagens originais ou das reconstruídas. Assim todas as imagens que se apresentam nas secções subsequentes são as saídas directas dos algoritmos de compressão/descompressão propostos. É óbvio que, numa aplicação final, poder-se-á proceder a pós-processamento adequado para amenizar os artefactos que possam surgir nas imagens reconstruídas, especialmente quando as originais tenham sido comprimidas a taxas muito altas.

5.2 Aspectos da Implementação

Depois de estabelecido um enquadramento teórico adequado para a compressão massiva do tipo de sequências de imagens de interesse, quer no respeitante à segmentação, quer à compressão propriamente dita, realizaram-se algumas experiências computacionais.

Assim, sintetizaram-se algumas sequências de imagens que consistiam em objectos geométricos bem definidos deslocando-se dentro do campo das imagens, por exemplo, com deslocações feitas segundo direcções verticais ou quase verticais, no

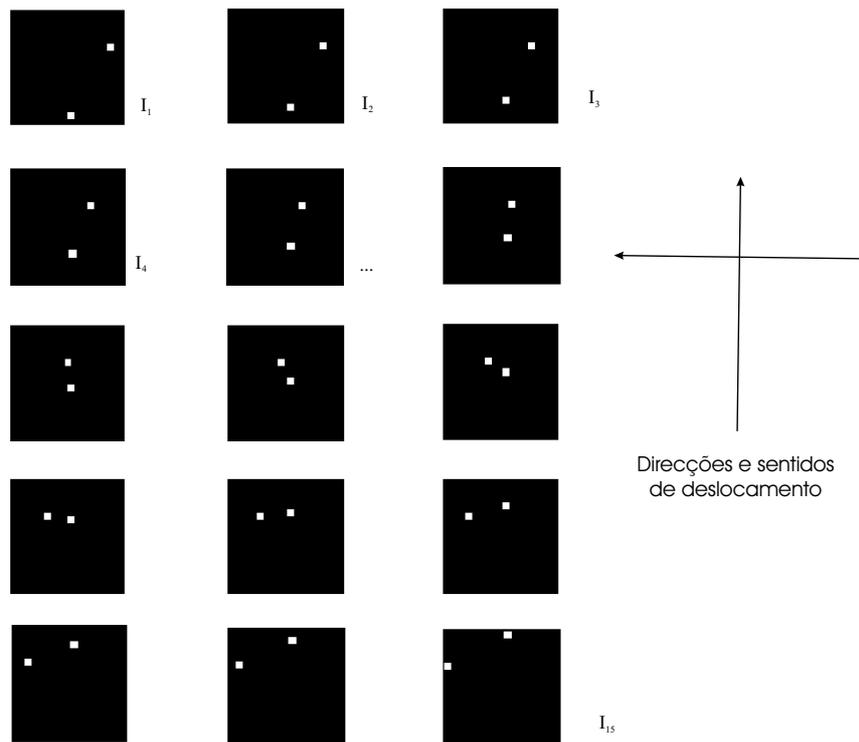


Figura 5.1: Sequência sintetizada para teste

sentido ascendente e de modo a simular o movimento existente nas sequências reais. Na figura 5.1, mostra-se uma das sequências utilizadas nesta fase. Neste caso, a sequência tinha movimento não só vertical mas também horizontal.

Com base nestas sequências sintéticas, validou-se uma primeira versão computacional dos algoritmos anteriormente estabelecidos.

Após o estabelecimento deste primeiro modelo computacional, procedeu-se à sua validação utilizando-se agora sequências de imagens submarinas reais, onde se registava actividade hidrotermal manifestada pela emissão de bolhas gasosas.

A aplicação desenvolvida para implementar os algoritmos de compressão/descompressão corre em ambiente *Windows* e a respectiva interface é apresentada na figura 5.2. Esta aplicação permite um processamento rápido, para além de ter uma interface que possibilita uma fácil selecção das operações e opções.

Também se procurou que a aplicação fosse flexível possibilitando a escolha de vários parâmetros tais como: dimensão das imagens, escolha do número de imagens a comprimir simultaneamente, etc.. Refira-se também que foram implementadas

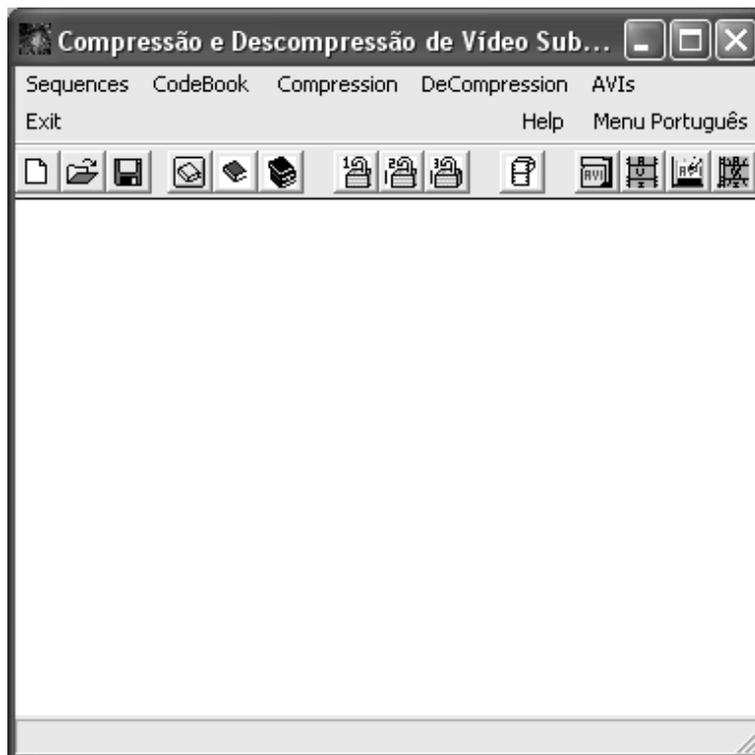


Figura 5.2: Interface da aplicação desenvolvida

funcionalidades de manipulação de sequências AVI (*Audio and Video Interlaced*), que é um dos padrões do *Windows* para a manipulação de sequências de vídeo. Esta implementação foi feita de modo a possibilitar que a aplicação tomasse uma sequência de imagens, a manipulasse comprimindo-a e depois na fase de descompressão pudesse, a partir das imagens comprimidas, descomprimi-las e gerar um AVI, agora da sequência reconstruída.

5.3 Simulações

5.3.1 Sequências Utilizadas

Nas simulações foram usadas sequências reais de imagens submarinas onde se regista actividade hidrotermal manifestada pela emissão de bolhas gasosas, as quais se deslocam para a superfície em colunas.

Foram feitas simulações com 5 sequências diferentes de imagens. Estas sequências não são muito longas, têm cerca de 6 segundos cada, e foram obtidas a partir de um

vídeo de prospecção filmado por operadores humanos no *Banco D. João de Castro*, banco hidrotermal situado nos Açores. Um facto relacionado com estas condições de captação das imagens traduz-se por alguma instabilidade na câmara, a qual se reflecte através de movimentos globais.

As imagens destas sequências foram convertidas para o formato QCIF, com 144×176 *pixels*, e foram amostradas a 15 *frames* por segundo (fps).

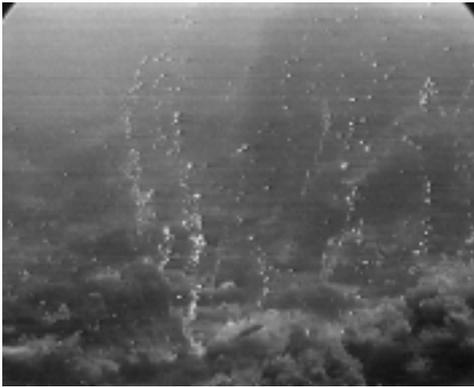
Note-se que foi utilizada esta taxa de amostragem de 15 fps nas simulações, apesar de alguns autores [26, 27] referirem que, para aplicações de *Very Low Bit Rate Video Coding*, se podem utilizar taxas de amostragem mais baixas, da ordem dos 5 a 10 fps.

5.3.2 Segmentação das Imagens

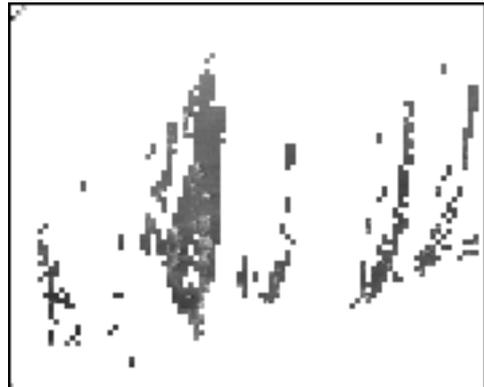
Em todas estas sequências fez-se a segmentação das imagens em zonas contendo blocos com movimento, isto é, zonas onde havia bolhas gasosas e zonas sem movimento, classificado como *background*. Recorde-se que, de acordo com a técnica proposta para a segmentação das imagens baseada em filtros de Gabor, quando se utiliza a expressão 'sem movimento' está-se a referir a movimento respeitante a bolhas e não a outros tipos de movimento, por exemplo, devido a movimentos das algas provocadas por correntes submarinas ou a peixes. Este tipo de movimentos não tem interesse para o fim concreto em vista e foi classificado como irrelevante sendo portanto eliminado pelos filtros.

A técnica desenvolvida apresenta um bom comportamento, tendo-se obtido uma segmentação correcta e precisa das zonas que continham movimento. Nas figuras 5.3 a 5.5, mostram-se alguns exemplos de imagens onde se procedeu a esta segmentação. São mostradas as imagens originais e as respectivas imagens segmentadas, podendo-se constatar o desempenho eficiente do algoritmo na detecção do movimento ascendente.

A reforçar a ideia de que a técnica proposta para a segmentação tem um bom

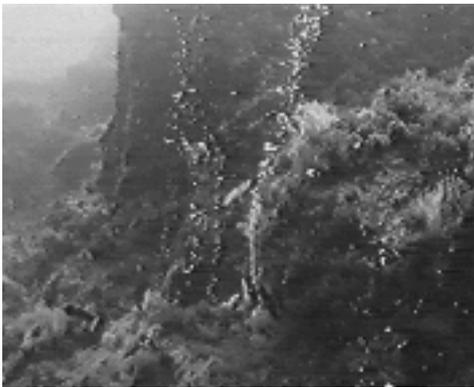


a) Imagem original



b) Imagem segmentada

Figura 5.3: Segmentação - Sequência Açores 1

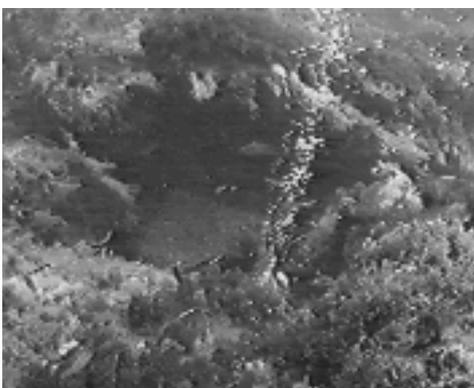


a) Imagem original

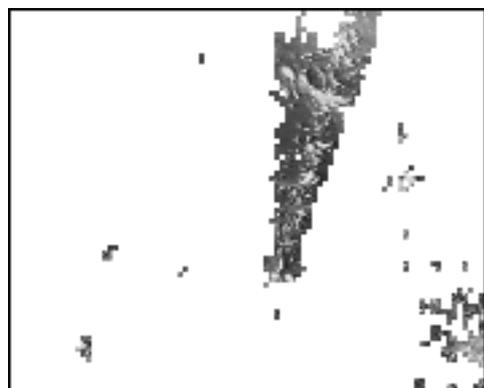


b) Imagem segmentada

Figura 5.4: Segmentação - Sequência Açores 3



a) Imagem original



b) Imagem segmentada

Figura 5.5: Segmentação - Sequência Açores 5



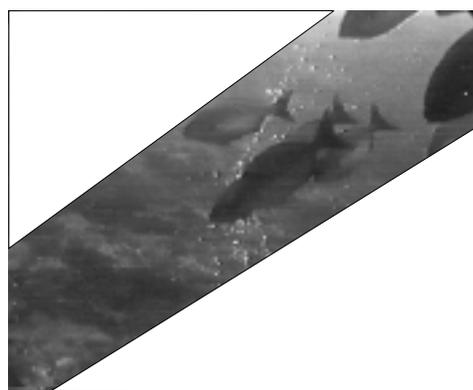
a) Original 12



b) Original 20



c) Original 30



d) Diagonal do movimento do cardume

Figura 5.6: Sequência Açores 2

desempenho, note-se a segmentação efectuada na segunda sequência e da qual se mostram algumas imagens na figura 5.6-a), b) e c).

Nesta sequência aparece um cardume de peixes que se deslocam dentro do campo da imagem quase numa diagonal, isto é, "entrando" pelo canto superior direito das imagens e "saindo" pelo canto inferior esquerdo. Digamos que, a direcção do movimento deste cardume é diagonal e corresponderia aproximadamente à faixa mostrada na figura 5.6-d).

No entanto, da segmentação resulta a faixa de blocos classificados como 'com movimento' que é mostrada na figura 5.7, a qual corresponde à coluna de emissões gasosas como pretendido.

Na tabela 5.1, faz-se um resumo dos resultados obtidos nesta etapa de segmentação para as várias sequências de imagens indicando-se, nomeadamente, o número de blocos classificado como 'com movimento' e calculando-se a taxa de

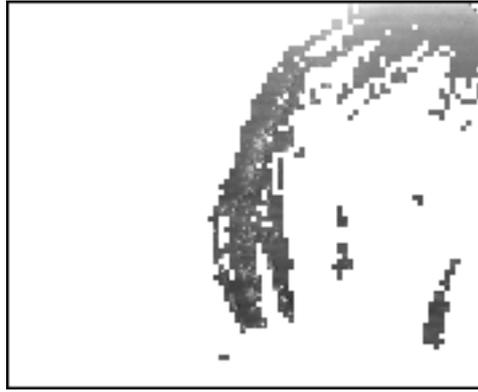


Figura 5.7: Blocos classificados como 'com movimento'

Sequência	Número Total de Imagens	Imagens Processadas em Simultâneo	Número Total de Blocos	Blocos Classificados como Com Movimento	Taxa de Compressão na Segmentação
Açores 1	90	15	6336	736	1/8
Açores 2	90	15	6336	984	1/6
Açores 3	90	15	6336	866	1/7
Açores 4	90	15	6336	615	1/10
Açores 5	90	15	6336	683	1/9

Tabela 5.1: Blocos classificados como 'com movimento' - Sequências Açores

compressão que se pode obter logo nesta etapa de segmentação.

Nesta tabela e de modo a não sobrecarregar a exposição dos resultados, só são apresentados os resultados da segmentação para sequências de imagens em que se fez a segmentação simultânea de 15 imagens, apesar de se terem feito também simulações com um número diferente de imagens em simultâneo.

5.3.3 Compressão dos Blocos

A compressão obtida no passo de segmentação, se bem que significativa, ainda é insuficiente face ao objectivo desejado. Após o passo de segmentação, é então efectuado o passo de compressão dos blocos que foram classificados como 'com movimento'.

A interface da aplicação é flexível permitindo escolher qual o tipo de compressão a ser usado. Na figura 5.8, mostra-se este ecrã de opções, as quais podem ser uma das quatro seguintes e que correspondem aos algoritmos de compressão descritos no



Figura 5.8: Menu de selecção do tipo de compressão

capítulo 4:

- Compressão VQIND
- Compressão VQ2.5DN
- Compressão VQ3DN
- Compressão VQZIP

Na figura 5.9 e 5.10, apresentam-se imagens do ecrã mostrado pela aplicação, após terem sido efectuados os passos de segmentação e de compressão. Na primeira figura, o ecrã refere-se ao processamento utilizando VQIND de um grupo de 90 imagens, considerando 15 imagens em simultâneo da sequência *Açores 1*. A segunda refere-se ao processamento da sequência *Açores 3* com VQ3DN, com as mesmas opções em termos de número de imagens.

Nas tabelas 5.2 a 5.5, mostram-se os resultados obtidos após o passo de segmentação baseada em filtros de Gabor seguido por, respectivamente, compressão

Sequência	Imagens Processadas em Simultâneo	Número Total de Bits	Bits a Transmitir	Taxa de Compressão Nominal	fps	Taxa de Transmissão (bits/s)
Açores 1	15	18.247.680	54.024	1/337	15	9.004
Açores 2	15	18.247.680	73.936	1/246	15	12.322
Açores 3	15	18.247.680	63.776	1/286	15	10.629
Açores 4	15	18.247.680	45.968	1/396	15	7.661
Açores 5	15	18.247.680	46.336	1/393	15	7.722

Tabela 5.2: Taxas de compressão obtidas com compressão VQIND

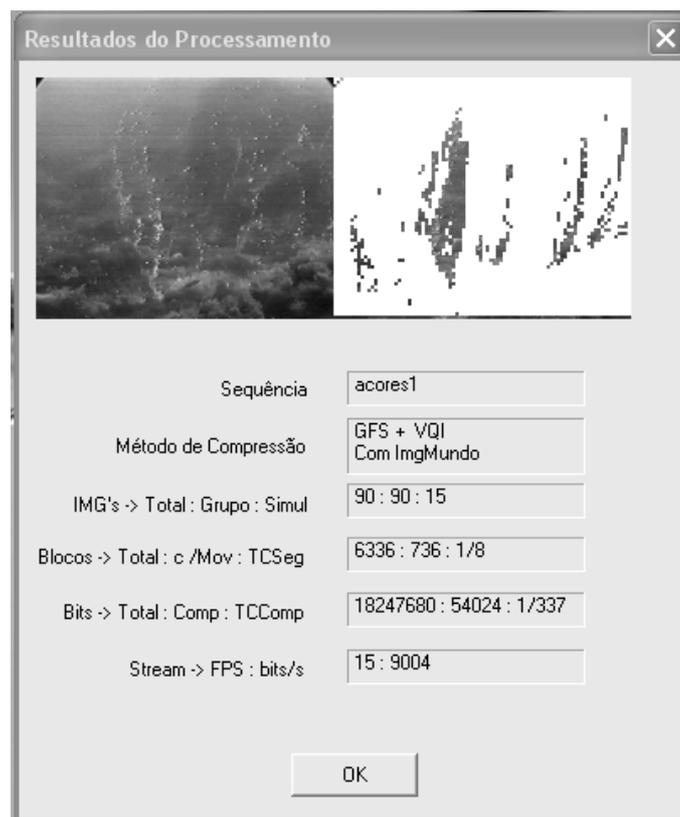


Figura 5.9: Resultados da compressão com VQIND - Açores 1

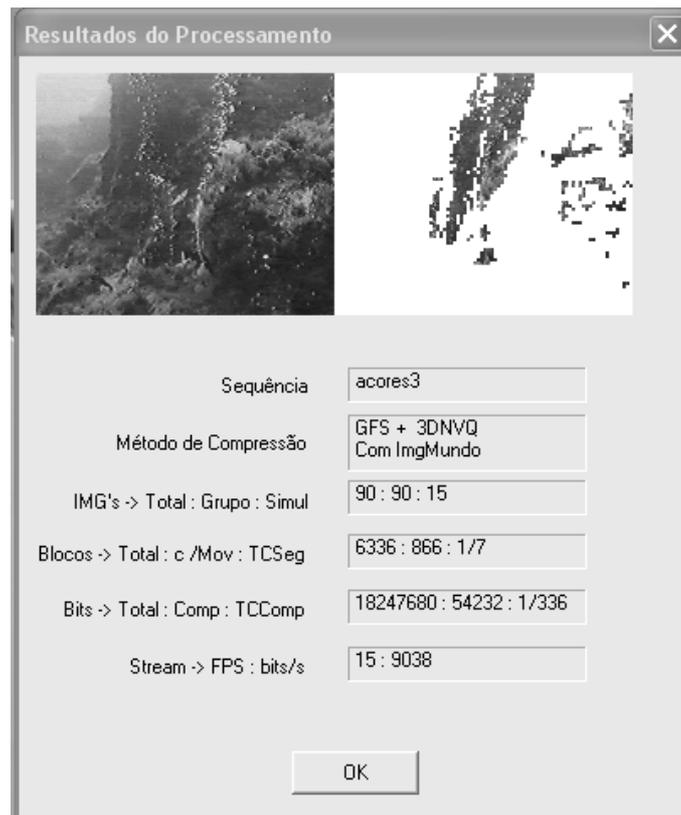


Figura 5.10: Resultados da compressão com VQ3DN - Açores 3

VQIND, VQ2.5DN, VQ3DN e VQZIP. Nestas tabelas, e apesar de esta compressão ser uma compressão com perda, não são indicados propositadamente os valores da distorção existente nas imagens reconstruídas preferindo-se enquadrar a análise dessa distorção na secção 5.4.4 onde se analisam os valores do PSNR absoluto isto é, o referente à distorção total existente nas imagens, e os valores do PSNR relativo, isto é, o referente somente à distorção existente nos blocos classificados como 'com movimento'.

Sequência	Imagens Processadas em Simultâneo	Número Total de Bits	Bits a Transmitir	Taxa de Compressão Nominal	fps	Taxa de Transmissão (bits/s)
Açores 1	15	18.247.680	45.776	1/398	15	7.629
Açores 2	15	18.247.680	63.568	1/287	15	10.594
Açores 3	15	18.247.680	54.152	1/336	15	9.025
Açores 4	15	18.247.680	40.072	1/455	15	6.678
Açores 5	15	18.247.680	41.032	1/444	15	6.838

Tabela 5.3: Taxas de compressão obtidas com compressão VQ2.5DN

Sequência	Imagens Processadas em Simultâneo	Número Total de Bits	Bits a Transmitir	Taxa de Compressão Nominal	fps	Taxa de Transmissão (bits/s)
Açores 1	15	18.247.680	46.400	1/393	15	7.733
Açores 2	15	18.247.680	62.176	1/293	15	10.362
Açores 3	15	18.247.680	54.232	1/336	15	9.038
Açores 4	15	18.247.680	40.544	1/450	15	6.757
Açores 5	15	18.247.680	40.400	1/451	15	6.733

Tabela 5.4: Taxas de compressão obtidas com compressão VQ3DN

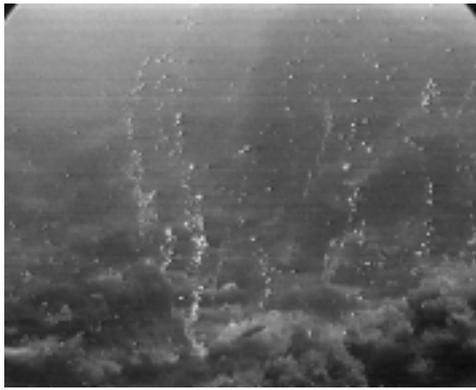
Sequência	Imagens Processadas em Simultâneo	Número Total de Bits	Bits a Transmitir	Taxa de Compressão Nominal	fps	Taxa de Transmissão (bits/s)
Açores 1	15	18.247.680	536.472	1/34	15	89.412
Açores 2	15	18.247.680	769.152	1/23	15	128.192
Açores 3	15	18.247.680	703.368	1/25	15	117.228
Açores 4	15	18.247.680	515.584	1/35	15	85.930
Açores 5	15	18.247.680	557.000	1/32	15	92.833

Tabela 5.5: Taxas de compressão obtidas com compressão VQZIP

Repare-se que, exceptuando o caso da compressão VQZIP, todas as técnicas consideradas conduziram a resultados que cumprem os objectivos iniciais, i.e., exigem taxas de transmissão inferiores a $20Kbps$. Assim, a escolha de qual destas técnicas deve ser utilizada numa aplicação real deverá recair numa das 3 primeiras, devendo "desempatar-se" essa escolha com base noutros critérios como, por exemplo, a qualidade final das sequências reconstruídas. Esta pode ser aferida subjectivamente, com base na qualidade visual das imagens ou, mais objectivamente, com base no PSNR.

5.3.4 Reconstrução das Imagens

Esta fase de reconstrução é a mais leve do ponto de vista computacional. Inicialmente, é consultada uma tabela indicativa da posição relativa dos blocos classificados como 'com movimento'. Depois, pela leitura dos índices de entrada no *codebook* de cinzentos ou no de índices seguido do de cinzentos, faz-se a reconstrução desses blocos pela atribuição dos valores lidos no *codebook* de cinzento a *pixels* dos blocos de

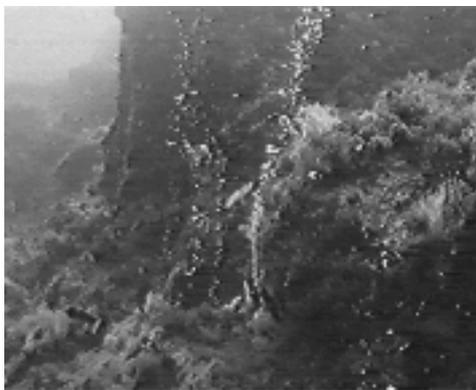


a) Imagem original



b) Imagem reconstruída

Figura 5.11: Reconstrução - Sequência Açores 1



a) Imagem original



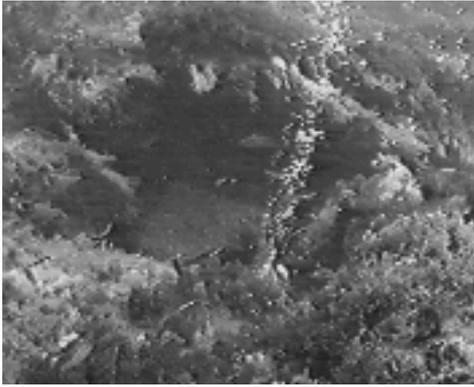
b) Imagem reconstruída

Figura 5.12: Reconstrução - Sequência Açores 3

imagens reconstruídos. Após esta reconstrução de blocos, a reconstrução das imagens é feita pela justaposição na imagem base de referência, a qual foi enviada na sua totalidade, destes blocos na sua posição correcta.

A partir destas imagens reconstruídas é gerado automaticamente um AVI de modo a poder ver-se a sequência.

Nas figuras 5.11 a 5.13 são mostradas as imagens originais e as respectivas imagens reconstruídas de 3 das 5 sequências consideradas. A compressão aqui utilizada foi a VQ3DN.



a) Imagem original



b) Imagem reconstruída

Figura 5.13: Reconstrução - Sequência Açores 5

5.3.5 Informação Lateral a Enviar

Para a correcta reconstrução das imagens é necessário que o decodificador conheça o *n-codebook*, o *codebook*, ou este e o *i-codebook*, consoante a opção de compressão considerada. No entanto, não é necessário estar sempre a enviar estes *codebooks*. Na secção 5.6 incluem-se as conclusões de um estudo efectuado relativo à estabilidade dos codebooks.

Para além destes *codebooks* é necessário, obviamente, enviar também a versão comprimida da imagem base de referência, que pode ser a primeira ou a *imagem do mundo*, isto é, uma imagem que constitui o *background* comum ao conjunto de imagens. Esta imagem é comprimida usando quantização vectorial tradicional ou DCT. Nos exemplos apresentados utilizou-se compressão DCT.

Outro tipo de informação que é preciso enviar também é a informação sobre a posição relativa dentro das imagens dos blocos com movimento. Esta informação é necessária de modo a possibilitar uma correcta justaposição destes blocos na imagem de referência.

A influência na taxa de compressão desta informação lateral relaciona-se, essencialmente, com a informação que é preciso enviar sobre a posição relativa dos blocos. De modo a baixar o *overhead* desta transmissão desenvolveu-se um esquema para a compressão desta informação que, basicamente, faz um RLE (*Run Length Encoding*) dessa informação. O que interessa indicar ao decodificador é se determinada

posição está *off*, isto é, se o bloco nessa posição não tem movimento, ou se está *on*, isto é, se o bloco tem movimento. Esta informação traduz-se, em termos práticos, por uma sequência de zeros e uns, 100111000..... , que é necessário comprimir. O que se faz, é indicar o número de ocorrências de cada um desses bits e depois, caso haja alternância, indica-se o número de ocorrências do novo bit, e assim sucessivamente. Deste modo, e dependendo do tamanho do bloco de imagens a segmentar em simultâneo, o *overhead* devido à necessidade de se transmitir esta informação é bastante reduzido.

Para se ter uma ideia precisa das taxas de compressão obtidas, considerando não só as imagens comprimidas mas também os refrescamentos dos *codebooks* e esta informação lateral, indica-se na secção 5.8 as conclusões de um estudo efectuado sobre a taxa de compressão nominal, ou seja, a taxa de compressão obtida considerando só os blocos de imagem a comprimir, a taxa de compressão efectiva, isto é, a taxa de compressão obtida considerando também as imagens de referência e também a taxa de compressão global, isto é, considerando agora também os *codebooks* a transmitir.

5.4 Análise de Resultados

Os resultados obtidos com os algoritmos propostos podem ser analisados segundo 4 perspectivas diferentes, se bem que relacionadas umas com as outras. Assim, podemos analisar estes resultados dos seguintes pontos de vista:

- Desempenho do algoritmo relativamente à capacidade de segmentar as imagens em zonas de interesse, zonas onde se nota a presença de bolhas gasosas, ou zonas de *background*;
- Desempenho do algoritmo relativamente às taxas de compressão obtidas e ao tamanho das *streams* de vídeo;
- Desempenho do algoritmo relativamente à qualidade visual das imagens reconstruídas;

Sequência	Número Total de Blocos	Blocos classificados como 'com movimento'	Taxa de Compressão na Segmentação
Açores 1	6336	736	1/8
Açores 2	6336	984	1/6
Açores 3	6336	866	1/7
Açores 4	6336	615	1/10
Açores 5	6336	683	1/9

Tabela 5.6: Taxa de compressão obtida na segmentação com filtros de Gabor

- Desempenho do algoritmo relativamente à qualidade das imagens reconstruídas, com base na análise da relação sinal-ruído de pico, PSNR.

5.4.1 Desempenho do Algoritmo na Segmentação

Um dos pressupostos base do algoritmo proposto, considerando a estratégia global adoptada na abordagem do problema, é o de ter a capacidade de proceder a uma segmentação correcta das imagens em zonas com movimento devido às bolhas e zonas sem este movimento.

Para isso, o algoritmo não deve limitar-se a seleccionar indiscriminadamente os movimentos nas imagens mas sim, com base em critérios que permitem considerar apenas os movimentos de interesse, isolá-los de outros que não devem ser considerados. Estes aspectos foram plenamente conseguidos e como se pode ver nas figuras 5.3 a 5.5 onde se apresentam as imagens originais e as respectivas imagens segmentadas, foi feita uma correcta segmentação.

O número de blocos que em cada sequência foram classificados como 'com movimento', bem como a taxa de compressão obtida considerando-se somente a segmentação, são mostrados na tabela 5.6.

Note-se que, para as sequências que foram filmadas mais de perto o número de blocos tende a aumentar, o que é natural, pois nesses casos a zona com bolhas tende a ocupar em maior percentagem o campo das imagens.

Atendendo aos pressupostos de partida e aos resultados obtidos, no que concerne à segmentação, podemos afirmar que o algoritmo tem um desempenho muito bom.

Método	Taxa de Compressão	fps	Taxa de Transmissão (bits/s)
VQIND	1/337	15	9.004
VQ2.5D	1/398	15	7.629
VQ3D	1/393	15	7.733
VQZIP	1/34	15	89.412

Tabela 5.7: Comparação das taxas de compressão - Sequência Açores 1

5.4.2 Desempenho do Algoritmo na Compressão Total

Existindo grandes restrições nos valores das taxas de transmissão máximas possíveis de serem usadas, torna-se necessário obter taxas de compressão muito altas.

Também neste aspecto, os algoritmos propostos satisfazem os objectivos iniciais, como se pode constatar pelos dados constantes na tabela 5.7. Três das opções consideradas permitem obter *streams* de vídeo perfeitamente passíveis de serem transmitidas pelo canal acústico submarino abaixo do débito máximo inicialmente imposto de 20 – 30Kbps. Os dados aqui apresentados referem-se a taxas de compressão e de transmissão nominais, mas mesmo as respectivas taxas efectivas estão dentro destes limites como se mostra na secção 5.6.

Note-se que, os dados apresentados referem-se a uma cadência de transmissão destas sequências de 15 fps, sendo possível baixar esta taxa de transmissão caso se utiliza uma cadência de, por exemplo, 10 fps.

5.4.3 Qualidade Visual das Imagens Reconstruídas

Os bons resultados atingidos, quer em termos de segmentação, quer em termos de taxas de compressão, não teriam qualquer significado e interesse se as imagens reconstruídas não tivessem uma qualidade visual que pudesse ser considerada aceitável ou boa pelo utilizador final.

Dada a subjectividade desta classificação submeteram-se a um conjunto de 6 indivíduos as sequências reconstruídas e foi-lhes pedido que as classificassem do ponto de vista da qualidade visual como: péssima, má, razoável, boa e muito boa.

Sequência	Péssima	Má	Razoável	Boa	Muito Boa
Açores 1			2	4	
Açores 2		1	3	2	
Açores 3			2	4	
Açores 4			4	2	
Açores 5			3	3	

Tabela 5.8: Classificação visual - Sem conhecer original - Açores

Sequência	Péssima	Má	Razoável	Boa	Muito Boa
Açores 1		1	3	2	
Açores 2		3	3		
Açores 3		1	4	1	
Açores 4		2	4		
Açores 5		2	3	1	

Tabela 5.9: Classificação visual - Conhecendo o original - Açores

Numa primeira fase, foram-lhes mostradas as sequências reconstruídas sem terem conhecimento das sequências originais de modo a não os condicionar. As classificações obtidas são as mostradas na tabela 5.8.

Numa segunda fase, mostraram-se aos mesmos indivíduos as sequências originais e as reconstruídas e pediu-se-lhes que as classificassem novamente segundo os mesmos critérios. Os resultados obtidos nesta segunda avaliação são mostrados na tabela 5.9.

Note-se que, as classificações atribuídas nesta segunda fase não foram tão boas como na primeira. No entanto, existe invariância no número de classificações "razoável".

Uma possível explicação para esta diferença prende-se com a estratégia assumida de se transmitir na totalidade uma só imagem e sobre esta ir-se sucessivamente justificando os blocos das outras imagens para as reconstruir. Isto leva a que, por exemplo, movimentos globais nas imagens, geralmente devidos a movimentos da câmara, não sejam rigorosamente captados e como tal possam ser mal classificados na avaliação. Outro aspecto que pode ter influenciado esta segunda avaliação foi o facto, também inerente à estratégia assumida, de não se considerarem os movimentos que não os verticais ou quase verticais. Assim os observadores, analisando as duas sequências

tendiam a fazer uma pior classificação da reconstruída pois não detectavam os movimentos das algas ou o dos peixes que viam nas originais. Um caso paradigmático desta situação é o referente à sequência *Açores 2* onde, como já atrás referido, existe um cardume de peixes deslocando-se na diagonal do campo da imagem. Ora como esse movimento é eliminado na sequência reconstruída esta foi bastante penalizada. No entanto, como também já anteriormente referido, isto está correcto do ponto de vista da especificação inicial e dos movimentos que interessavam detectar.

Observe-se também que estes indivíduos não tinham *a priori* nenhum conhecimento das técnicas que estavam a ser utilizadas, nem conheciam as especificações iniciais que se consideraram na sua implementação, nomeadamente, o facto de que os movimentos a reter deverem ser só os verticais ou quase verticais ou que o fundo devia ser o mesmo. Talvez por este facto, muitos destes avaliadores comentaram frequentemente que as sequências não estavam correctas porque os peixes não se mexiam, isto aquando da primeira avaliação, ou que não se viam nas imagens reconstruídas os peixes que viam nas originais, isto na segunda avaliação.

Parece-nos que esta explicação é válida, até devido aos resultados atribuídos numa avaliação semelhante que foi feita relativamente a sequências padrão em que nem as imagens têm muito movimento global nem o *background* tem também muito movimento. Esta avaliação é referida na secção 5.5.

5.4.4 Qualidade das Imagens Relativamente ao PSNR

Para além da avaliação visual da qualidade das imagens, fez-se um estudo baseado na relação sinal-ruído de pico entre as imagens originais e as reconstruídas. A medida considerada foi a expressa na equação (2.3.4), ou seja, considerando-se a energia máxima na imagem. Para as imagens utilizadas que eram imagens de 8 bits, $x_{max} = 255$.

No entanto, na aplicação desta relação e de modo a separarmos a influência no PSNR das duas operações efectuadas, a operação de segmentação e a de compressão,

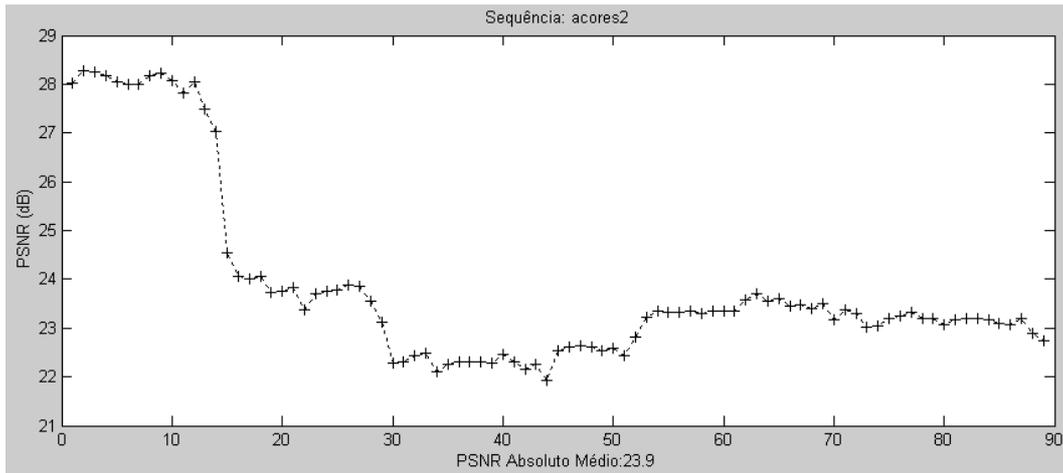


Figura 5.14: PSNR absoluto - Compressão com VQ3DN da Sequência Açores 2

foram feitos dois tipos de considerações. Dado ter sido assumido que só era transmitida na totalidade uma imagem e que todas as outras eram reconstruídas a partir desta, havia por um lado logo à partida, devido a este facto, uma penalização no PSNR devido à influência que o *background* poderia ter; por outro lado, as várias estratégias de compressão vectorial utilizadas também provocam distorção, o que leva a que o PSNR seja também penalizado. Houve interesse em determinar qual era a influência de cada uma destas penalizações e deste modo isto foi considerado no estudo efectuado.

Assim, considera-se um PSNR absoluto e um PSNR relativo. O PSNR absoluto é aquele que influenciará a imagem no seu todo, sofre a influência tanto das penalizações devidas à segmentação como às da compressão, e foi determinado comparando-se as imagens originais e reconstruídas no seu todo. Por sua vez o PSNR relativo, é o PSNR devido apenas às operações de compressão, isto é, não considerando o facto de grande parte das imagens reconstruídas estarem afectadas por um *background* ligeiramente diferente do que o que originalmente tinham. Assim, esse PSNR relativo foi calculado somente considerando os blocos das imagens originais classificados como 'com movimento' com os homólogos nas imagens reconstruídas, ou seja, não entrou aqui o *factor background*.

Nas figuras 5.14 a 5.16, mostram-se alguns gráficos que dão o valor do PSNR

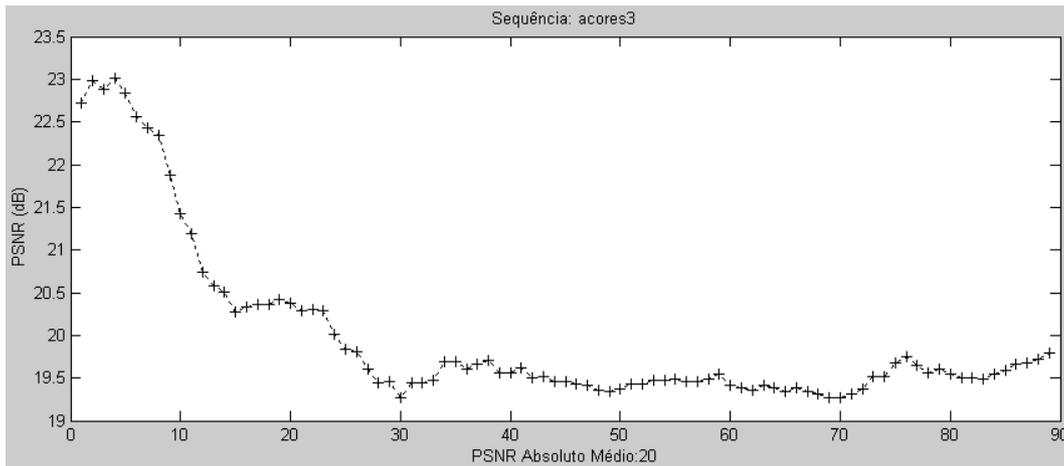


Figura 5.15: PSNR absoluto - Compressão com VQ3DN da Sequência Açores 3

Sequência	Número Total de Imagens	Imagens Processadas em Simultâneo	PSNR Absoluto Médio
Açores 1	90	15	24,2
Açores 2	90	15	23,9
Açores 3	90	15	20,0
Açores 4	90	15	19,0
Açores 5	90	15	18,4

Tabela 5.10: PSNR absoluto médio - Compressão com VQ3DN - Sequências Açores

absoluto em cada uma das imagens das 3 sequências, bem como o valor médio do PSNR absoluto para essas sequências. Estes gráficos referem-se à compressão com 15 imagens em simultâneo e utilizando-se a compressão VQ3DN. Como se constata por estes gráficos, os valores do PSNR mantêm-se para os 6 segundos destas sequências dentro de limites razoáveis. A causa para esta degradação do PSNR pode ser devida, quer à substituição em cada imagem do background original pelo background de referência ou então, ser devida ao facto de se estar a fazer compressão VQ multi-bloco. Somos de opinião de que essa degradação se deve ao *background*, pois considerando-se no PSNR a influência dos blocos classificados como 'com movimento', ou seja o PSNR relativo, os valores que se obtêm são maiores, como se mostra adiante.

Na tabela 5.10, resumem-se os valores médios do PSNR absoluto, obtidos para as sequências consideradas. Estes valores referem-se à compressão de 90 imagens de uma só vez em grupos de 15 imagens simultâneas com VQ3DN.

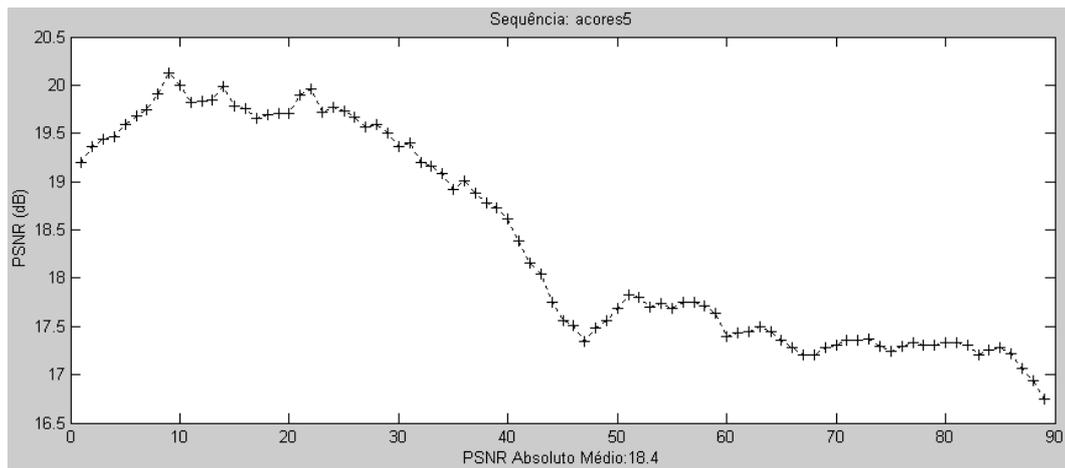


Figura 5.16: PSNR absoluto - Compressão com VQ3DN da Sequência Acores 5

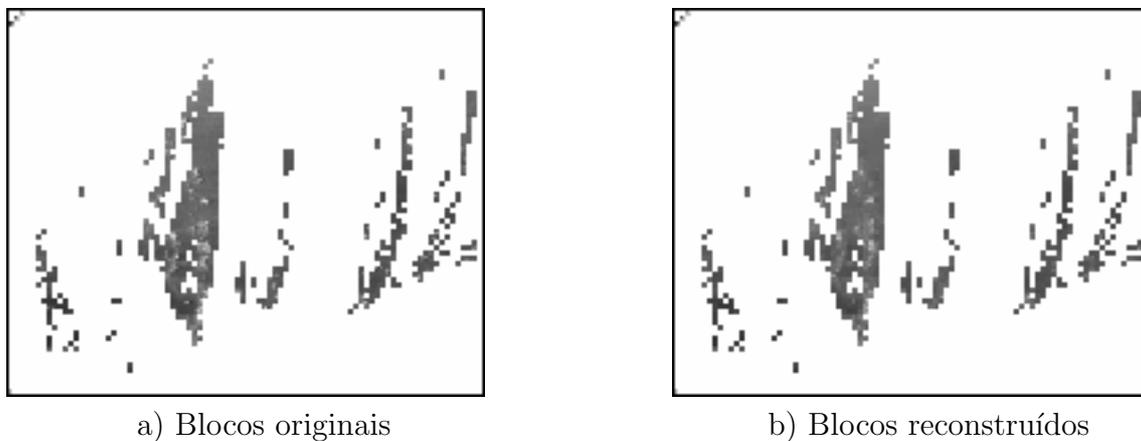


Figura 5.17: Blocos 'com movimento' - Sequência Acores 1

Nas figuras 5.17 a 5.19, mostram-se os blocos originais e os blocos reconstruídos das 3 sequências, isto é, agora só se mostram as zonas com blocos classificados como 'com movimento'. O PSNR relativo foi calculado com base neste tipo de imagens.

Nas figuras 5.20 a 5.22, mostram-se agora os gráficos do PSNR relativo para as mesmas sequências, bem como o valor médio do PSNR relativo para essas sequências.

Na tabela 5.11 são mostrados os valores médios do PSNR relativo considerando-se a compressão de 90 imagens de uma só vez em grupos de 15 imagens simultâneas com VQ3DN.

Como seria de esperar, os valores médios do PSNR relativo são melhores do que os do absoluto, onde a influência do *background* é importante. O interesse deste estudo diferenciado foi o de ter-se uma ideia da influência de cada etapa, pois o

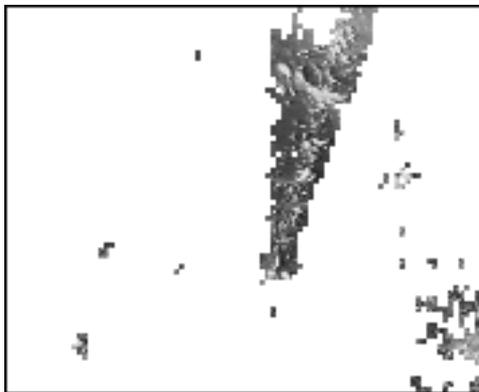


a) Blocos originais

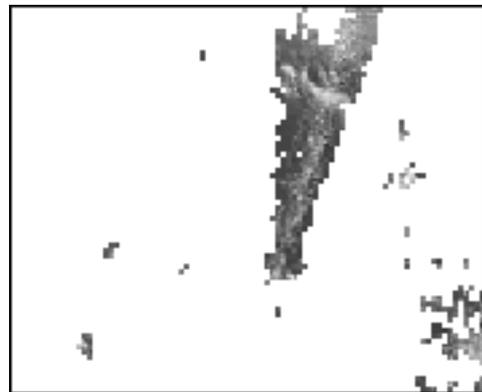


b) Blocos reconstruídos

Figura 5.18: Blocos 'com movimento' - Sequência Açores 3



a) Blocos originais



b) Blocos reconstruídos

Figura 5.19: Blocos 'com movimento' - Sequência Açores 5

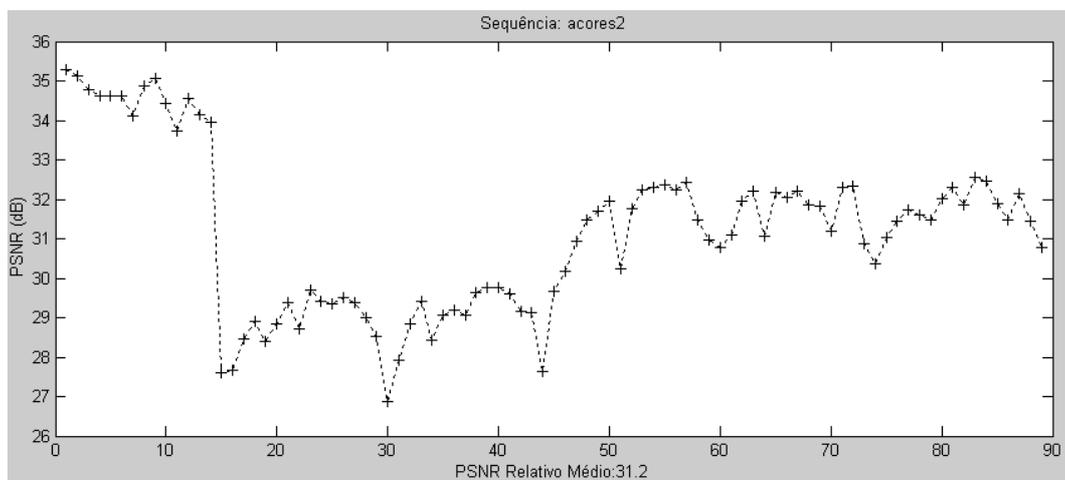


Figura 5.20: PSNR relativo - Compressão com VQ3DN da Sequência Açores 2

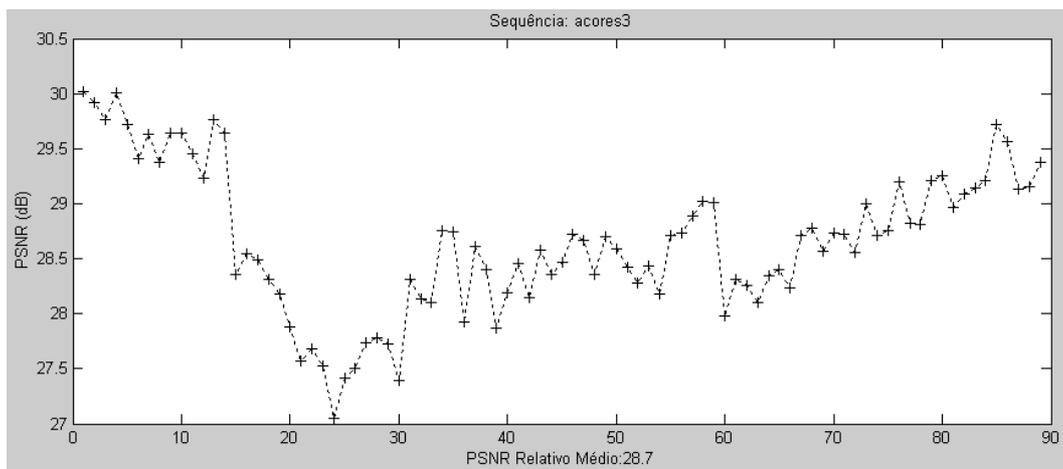


Figura 5.21: PSNR relativo - Compressão com VQ3DN da Sequência Açores 3

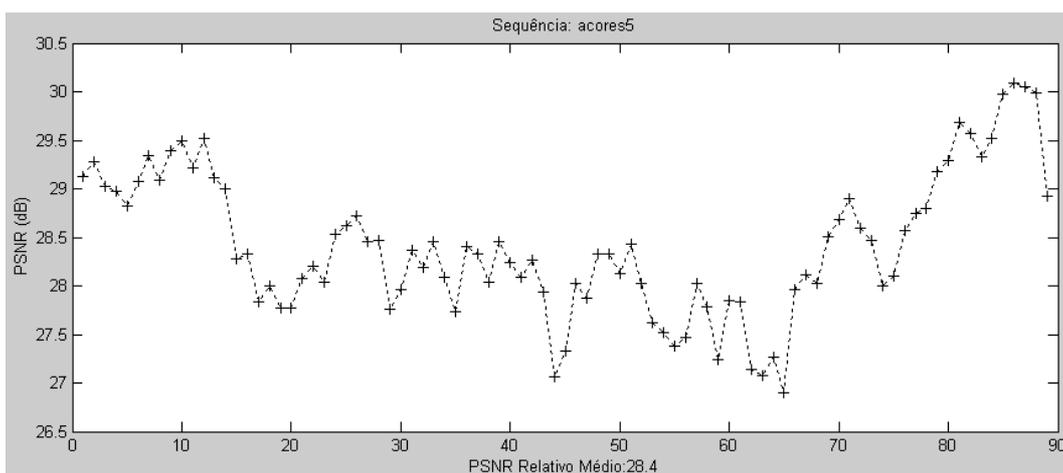


Figura 5.22: PSNR relativo - Compressão com VQ3DN da Sequência Açores 5

Sequência	Número Total de Imagens	Imagens Processadas em Simultâneo	PSNR Relativo Médio
Açores 1	90	15	33,9
Açores 2	90	15	31,2
Açores 3	90	15	28,7
Açores 4	90	15	32,9
Açores 5	90	15	28,4

Tabela 5.11: PSNR relativo médio - Compressão com VQ3DN - Sequências Açores

PSNR absoluto é o que transparecerá para o observador. Como mostrado na secção 5.5.2 e pelas razões já apontadas na secção 5.4.3, esta diferenciação não é tão grande quando se compara o PSNR absoluto e relativo nas sequências padrão de teste.

Também é interessante observar-se que há uma relativa identificação entre a classificação subjectiva, dada em termos de análise visual pelos observadores, e a que se poderia fazer com base no PSNR.

Assim, é necessário fazer-se uma interpretação da degradação do valor do PSNR absoluto. Como já referido atrás, o PSNR absoluto é fortemente penalizado pela substituição do *background* original pelo de referência. Esta penalização será obviamente tanto mais forte quanto mais diferente forem esses *backgrounds*, nomeadamente, se no original existirem, por exemplo, peixes ou movimentos relativamente fortes de algas. Esta constatação é evidente se, por exemplo, considerarmos as sequências *Açores 2* e *Açores 5*, cujos gráficos do PSNR absoluto são mostrados nas figuras 5.14 e 5.16. No caso da sequência 2, podemos ver pelo gráfico que o PSNR sofre uma brusca variação negativa a partir da imagem 12. Ora nesta sequência aparece um cardume de peixes, cuja influência já foi analisada e de que se mostraram imagens na figura 5.6, exactamente a partir das imagens 11-12 da sequência. Daí a degradação do PSNR, pois os peixes presentes no *background* original não aparecem no substituído. Uma consideração semelhante poderá ser feita para a sequência *Açores 5*, onde a degradação do PSNR se torna mais brusca a partir da imagem 30, exactamente aquela onde se começa a notar a presença, em quantidade, de pequenos peixes, que também estando presentes no *background* original não aparecem no de referência. Esta análise é corroborada pelos gráficos do PSNR relativos destas sequências.

5.4.5 Qualidade das Imagens *versus* Metodologia Empregue

Considerando os vários aspectos em que se baseou a análise que fizemos nas secções anteriores, podemos dizer que para uma aplicação real que se implementasse,

o algoritmo a escolher poderia ser qualquer um dos seguintes:

- Compressão VQIND
- Compressão VQ2.5DN
- Compressão VQ3DN

pois os resultados obtidos, utilizando-se qualquer uma destas opções, quer em termos de qualidade visual das imagens, quer em termos de qualidade aferida com base no PSNR, ou mesmo por consideração das taxas de compressão passíveis de serem obtidas, cumprem os objectivos especificados.

No entanto, a estratégia onde se utiliza segmentação por filtros de Gabor seguida por compressão baseada em quantização vectorial 3DN será a que, globalmente, permite a obtenção de melhores resultados. Para além disso, conduz também a uma melhor qualidade visual das imagens, sendo também menor o tempo de processamento.

5.5 Aplicação a Outros Tipos de Imagem

Com o objectivo de se ter uma ideia do comportamento dos algoritmos propostos, quando aplicados a outros tipos de imagens, procedeu-se à compressão de sequências de vídeo *standard*, que normalmente são utilizadas na aferição de algoritmos de compressão de vídeo, e à compressão de duas sequências de tráfego.

Se bem que os algoritmos não tenham sido originalmente desenhados para ser de aplicação geral, o seu comportamento, para qualquer uma destas sequências, é também bastante satisfatório.

5.5.1 Aplicação a Sequências Padrão

As sequências *Claire*, *Miss América* e *Salesman* são habitualmente usadas como sequências de teste.

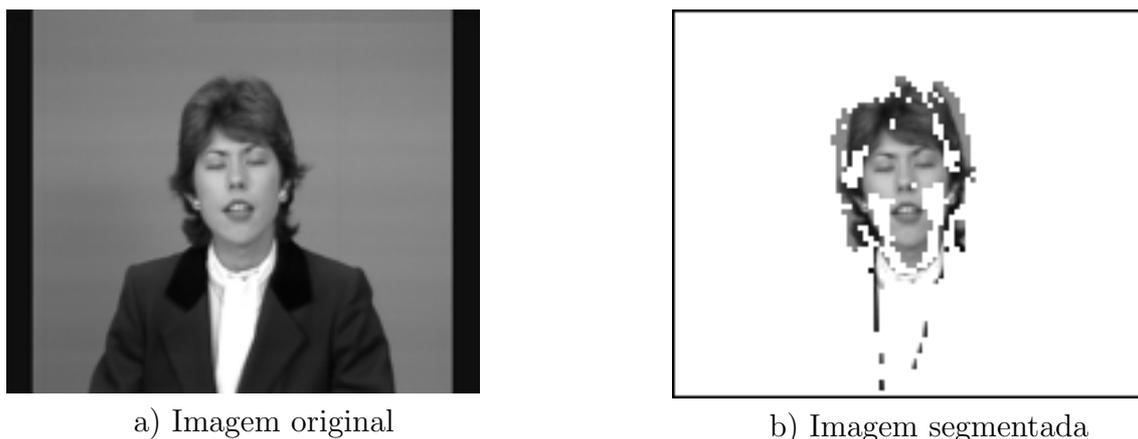


Figura 5.23: Segmentação - Sequência Claire

De modo a ter-se uma comparação mais próxima com as sequências submarinas, a duração da sequências padrão foi limitada a 6 segundos. Todas as imagens tinham 8 bits, a dimensão era QCIF e estavam amostradas a 15 fps.

Aqui também se utilizou o conceito de *imagem do mundo*, isto é, procura-se primeiro obter uma imagem que seja o *background* comum ao conjunto de imagens, e a reconstrução é feita com base nessa imagem. Neste género de imagens, que se caracterizam por terem um fundo estático e serem essencialmente imagens do tipo *cabeça e troncos* ou então com objectos bem definidos a deslocarem-se sobre esse fundo, é mais precisa a reconstituição, a partir dessa *imagem do mundo*. Também em termos de influência na taxa de compressão estaremos numa situação favorável pois essa *imagem do mundo* será relativamente constante durante muito tempo e como tal, não será necessária uma frequência de retransmissão muito grande.

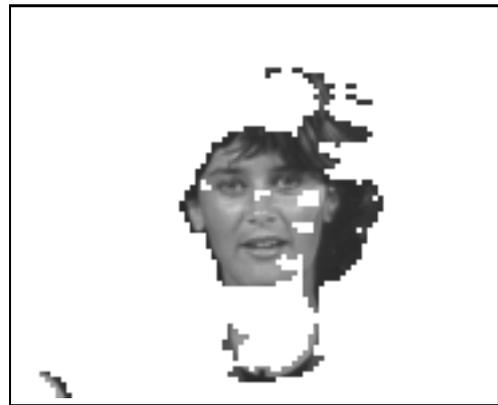
Nas figuras 5.23 a 5.25, mostram-se uma imagem original de cada uma destas sequências e a respectiva imagem segmentada usando filtros de Gabor.

Na tabela 5.12 resume-se, para estas 3 sequências, o número total de blocos e o número de blocos classificados como 'com movimento' em cada uma delas e a taxa de compressão obtida nesta fase de segmentação.

Por sua vez, no que diz respeito às taxas de compressão nominal obtidas, verifica-se a obtenção de elevadas taxas de compressão. É assim possível transmitir estas

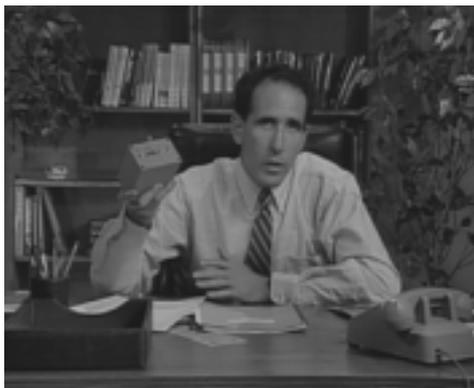


a) Imagem original



b) Imagem segmentada

Figura 5.24: Segmentação - Sequência Miss América



a) Imagem original



b) Imagem segmentada

Figura 5.25: Segmentação - Sequência Salesman

Sequência	Número Total de Blocos	Blocos Classificados como Com Movimento	Taxa de Compressão na Segmentação
Claire	6336	627	1/10
Miss América	6336	950	1/7
Salesman	6336	921	1/7

Tabela 5.12: Blocos classificados como 'com movimento' - Sequências Padrão

Sequência	Método	fps	Taxa de Compressão Nominal	Taxa de Transmissão (bit/s)
Claire	VQIND	15	1/383	7.926
	VQ3DN	15	1/464	6.548
Miss América	VQIND	15	1/296	10.269
	VQ3DN	15	1/336	9.032
Salesman	VQIND	15	1/261	11.624
	VQ3DN	15	1/314	9.676

Tabela 5.13: Taxas de compressão nominal e de transmissão - Sequências Padrão

sequências através de canais de transmissão com uma largura de banda muito pequena.

Nas figuras 5.26 a 5.28 mostra-se, uma imagem de cada uma destas sequências e a correspondente imagem reconstruída. Estas imagens foram comprimidas com a estratégia de compressão VQ3DN anteriormente referida e considerando-se a compressão com 15 imagens em simultâneo.

Os valores da taxa de compressão obtidas para estas sequências, bem como a taxa de transmissão, em *bit/s*, são mostrados na tabela 5.13. Nesta tabela e de modo a facilitar a exposição, só se apresentam os resultados obtidos com compressão VQIND e compressão VQ3DN e para o caso da compressão de 15 imagens em simultâneo.

Como se pode constatar pela análise destes resultados obtém-se altas taxas de compressão e de transmissão efectivas, sendo essas taxas adequadas à transmissão destas sequências por canais de banda estreita.

Quanto à qualidade visual das imagens reconstruídas procederam-se a testes similares aos efectuados para as imagens submarinas. Fizeram-se testes subjectivos de análise da qualidade visual da imagem, utilizando-se os mesmos 6 avaliadores, e obtiveram-se os resultados mostrados na tabela 5.14, para o caso da avaliação não condicionada pelo conhecimento das imagens originais, e os da tabela 5.15, para os da avaliação com conhecimento da sequência original. As imagens reconstruídas que lhes foram mostradas foram as obtidas usando-se VQ3DN.

Note-se que aqui, os resultados destas duas avaliações estão muito mais próximos uns dos outros do que no caso da avaliação das sequências de imagens submarinas.



a) Imagem original



b) Imagem reconstruída

Figura 5.26: Reconstrução - Sequência Claire

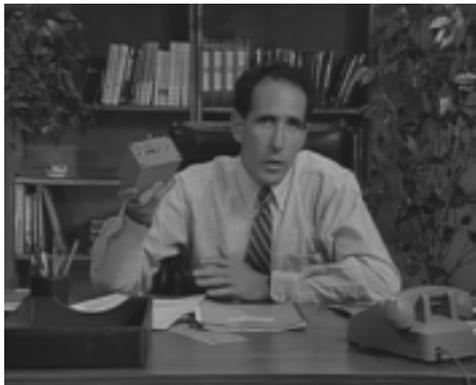


a) Imagem original



b) Imagem reconstruída

Figura 5.27: Reconstrução - Sequência Miss América



a) Imagem original



b) Imagem reconstruída

Figura 5.28: Reconstrução - Sequência Salesman

Sequência	Péssima	Má	Razoável	Boa	Muito Boa
Claire			3	3	
Miss América			2	4	
Salesman		1	3	2	

Tabela 5.14: Classificação visual - Sem conhecer original - Padrão

Sequência	Péssima	Má	Razoável	Boa	Muito Boa
Claire			3	3	
Miss América			4	2	
Salesman		1	4	1	

Tabela 5.15: Classificação visual - Conhecendo Original - Padrão

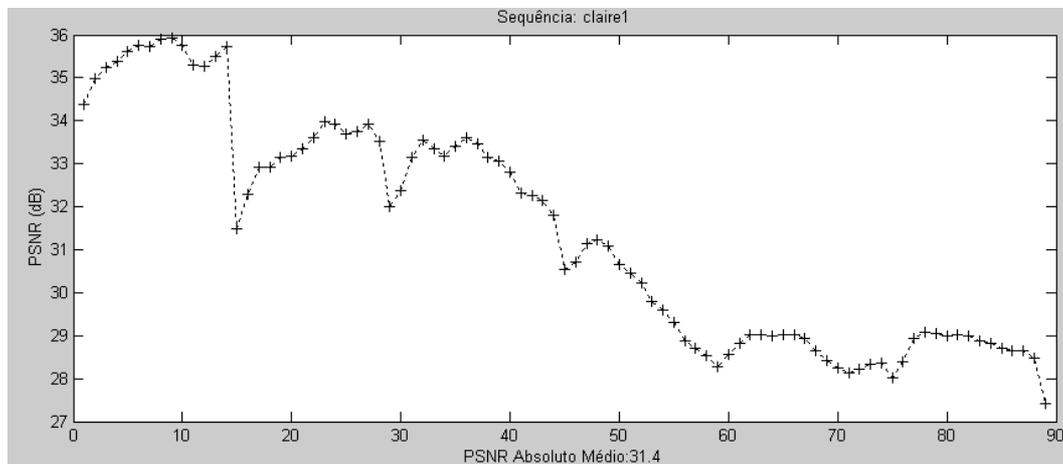


Figura 5.29: PSNR absoluto - Compressão com VQ3DN da Sequência Claire

Já foi dada uma possível explicação para este facto. Como referido, isto pode estar relacionado, por um lado, com o facto de aqui a câmara não ter efectuado movimentos e, por outro lado, o *background* ser quase estático, o que não era o caso das imagens submarinas em que o *background* tinha movimento se bem que de um tipo diferente do movimento de interesse.

Os resultados obtidos na avaliação da qualidade baseada no PSNR, estão representados nas figuras 5.29 a 5.31. Nestas figuras, são indicados o valor do PSNR absoluto obtido para cada uma das imagens de cada uma das sequências, bem como o valor médio obtido para cada uma das sequências.

Nas figuras 5.32 a 5.34, mostra-se o PSNR relativo para essas sequências.

Na tabela 5.16 resumem-se, para estas sequências padrão, os valores do PSNR, quer absoluto quer relativo obtidos após compressão com VQ3DN.

Devido à já referida menor influência do *background*, aqui os valores de PSNR, absoluto e relativo, são mais elevados do que aqueles que foram obtidos com as imagens submarinas.

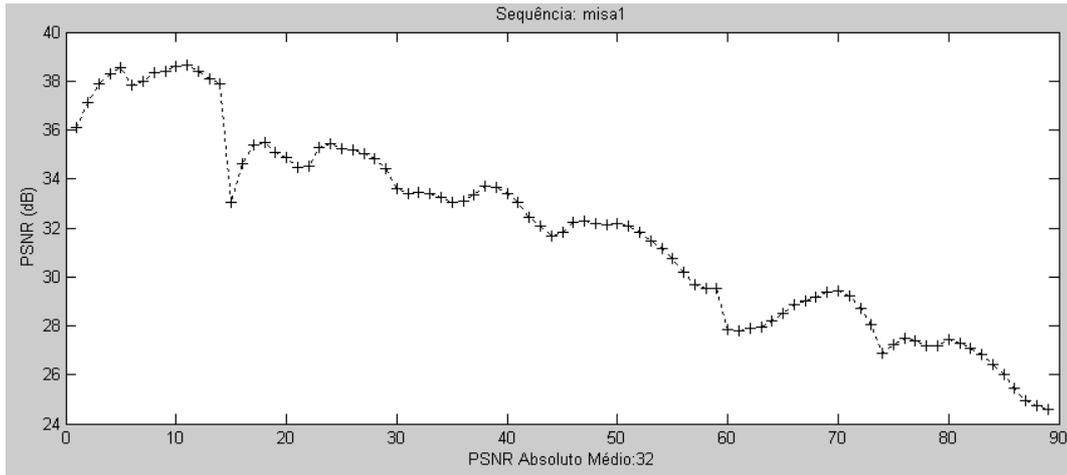


Figura 5.30: PSNR absoluto - Compressão com VQ3DN da Sequência Miss América

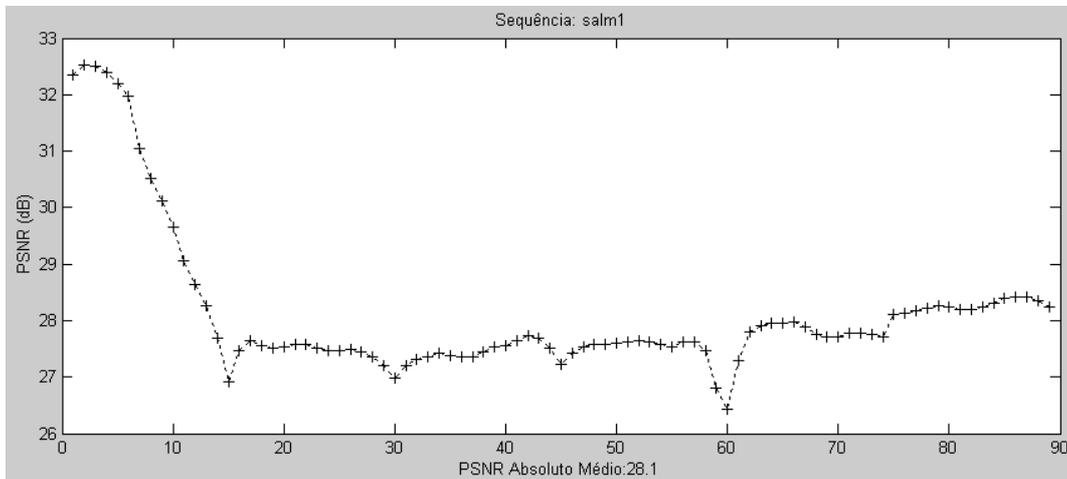


Figura 5.31: PSNR absoluto - Compressão com VQ3DN da Sequência Salesman

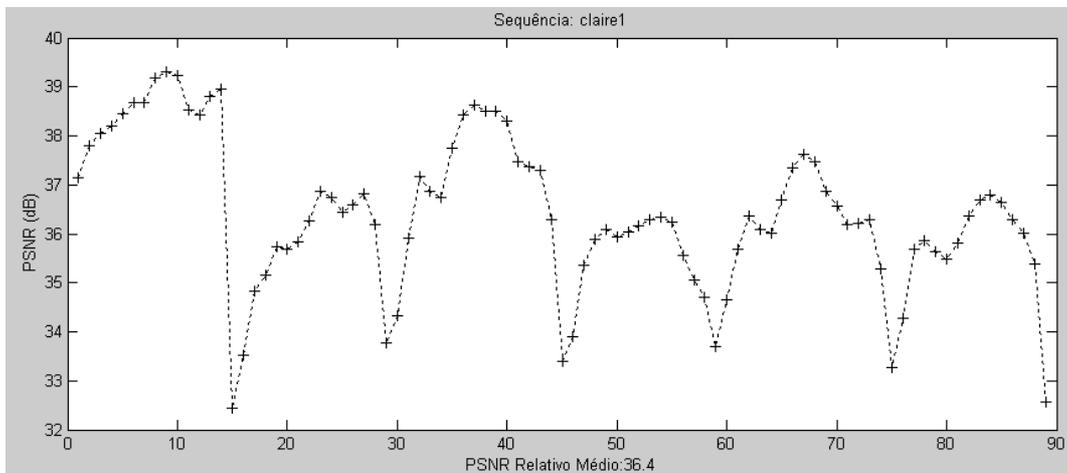


Figura 5.32: PSNR relativo - Compressão com VQ3DN da Sequência Claire

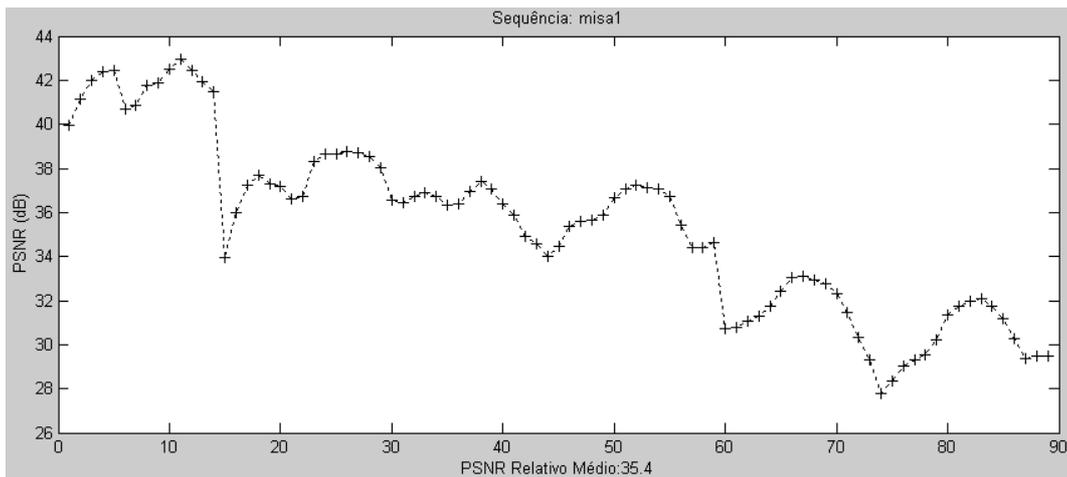


Figura 5.33: PSNR relativo - Compressão com VQ3DN da Sequência Miss América

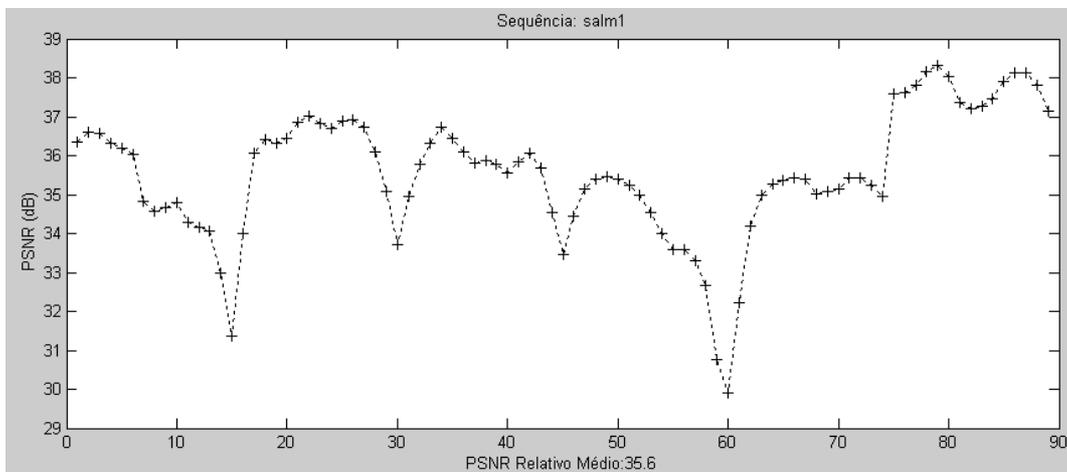


Figura 5.34: PSNR relativo - Compressão com VQ3DN da Sequência Salesman

5.5.2 Aplicação a Sequências de Tráfego

Fez-se uma aplicação similar à descrita na secção anterior a 2 sequências de tráfego.

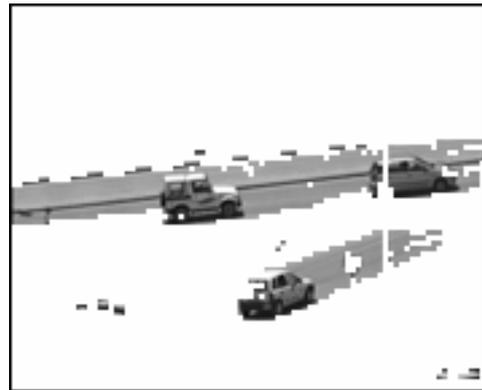
Numa destas sequências, os veículos foram filmados de lado e na outra foram filmados de cima. Também, para efeitos de comparação, ambas as sequências são de 6 segundos e as respectivas imagens são de 8 *bits* com tamanho QCIF e amostradas

Sequência	PSNR Absoluto	PSNR Relativo
Claire	31,4	36,4
Miss América	32,0	35,4
Salesman	28,1	35,6

Tabela 5.16: PSNR absoluto e relativo - Compressão VQ3DN - Sequências Padrão



a) Imagem original

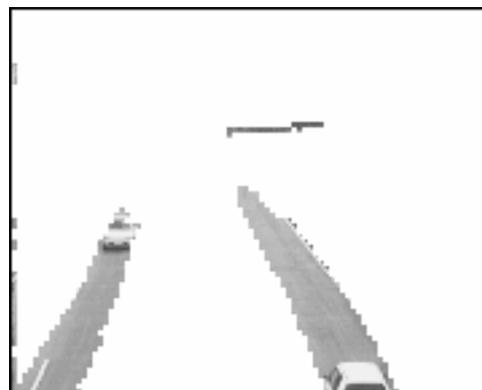


b) Imagem segmentada

Figura 5.35: Segmentação - Sequência Tráfego 1



a) Imagem original



b) Imagem segmentada

Figura 5.36: Segmentação - Sequência Tráfego 2

a 15 fps.

Os resultados aqui obtidos, tal como nos casos anteriores, atingem os objectivos especificados.

Nas figuras 5.35 e 5.36 são mostrados os resultados obtidos com a segmentação.

Na tabela 5.17, resume-se para estas sequências, o número total de blocos, o número de blocos classificados como 'com movimento' e as taxas de compressão obtidas nesta fase.

Sequência	Número Total de Blocos	Blocos Classificados como 'com movimento'	Taxa de Compressão na Segmentação
Tráfego 1	6.336	1.039	1/6
Tráfego 2	6.336	725	1/8

Tabela 5.17: Blocos classificados como 'com movimento' - Sequências Tráfego

Sequência	Método	fps	Taxa de Compressão Nominal	Taxa de Transmissão (bits/s)
Tráfego 1	VQIND	15	1/217	14.002
	VQ3DN	15	1/256	11.850
Tráfego 2	VQIND	15	1/303	10.030
	VQ3DN	15	1/361	8.417

Tabela 5.18: Taxas de compressão nominal e de transmissão - Sequências Tráfego



a) Imagem original



b) Imagem reconstruída

Figura 5.37: Reconstrução - Sequência Tráfego 1

As taxas de compressão obtidas são apresentadas na tabela 5.18, para o caso da compressão VQIND e VQ3DN, para 15 imagens em simultâneo.

A qualidade das imagens reconstruídas é boa também, como se constata pelas figuras 5.37 e 5.38, onde se mostra uma imagem original de cada uma destas sequências e a correspondente imagem reconstruída. Nas figuras 5.39 e 5.40 são mostrados os valores do PSNR absoluto para estas duas sequências.



a) Imagem original



b) Imagem reconstruída

Figura 5.38: Reconstrução - Sequência Tráfego 2

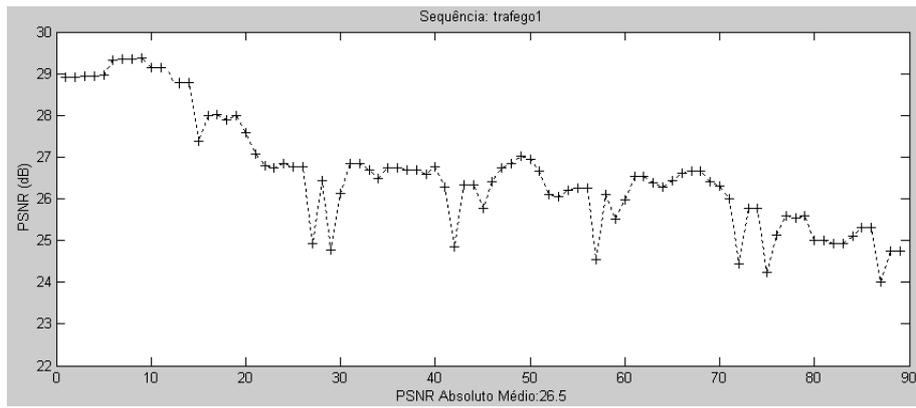


Figura 5.39: PSNR absoluto - Compressão VQ3DN da Sequência Tráfego 1

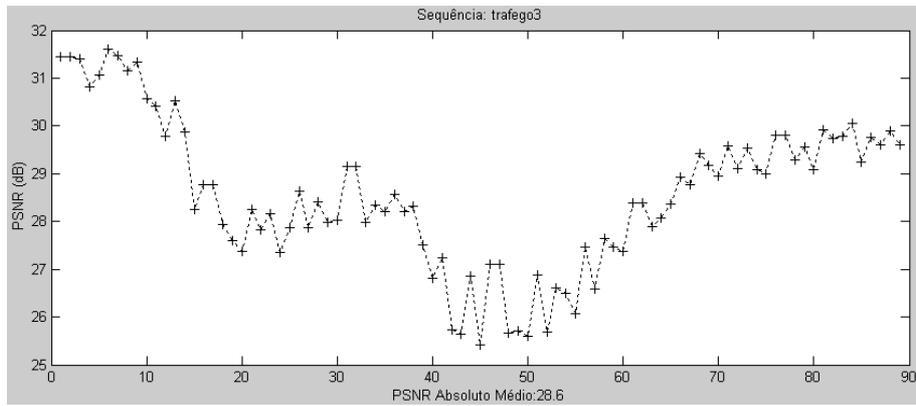


Figura 5.40: PSNR absoluto - Compressão VQ3DN da Sequência Tráfego 2

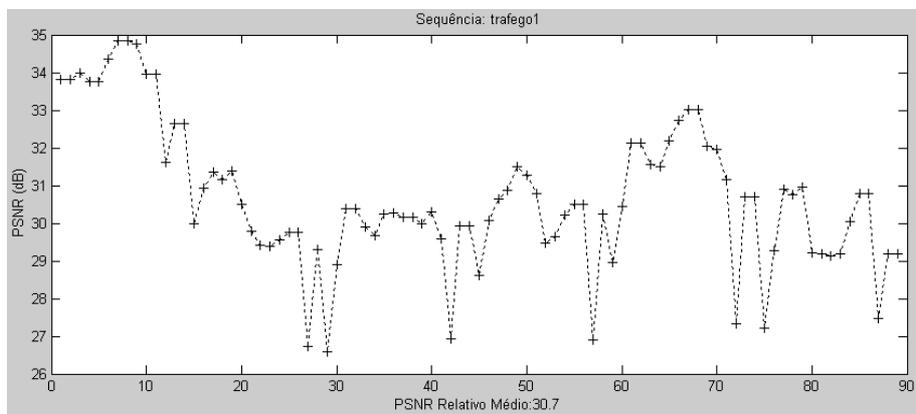


Figura 5.41: PSNR relativo - Compressão VQ3DN da Sequência Tráfego 1

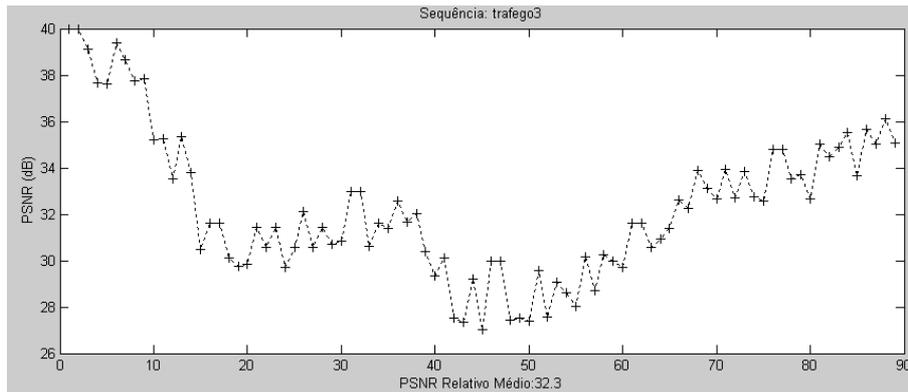


Figura 5.42: PSNR relativo - Compressão VQ3DN da Sequência Tráfego 2

Sequência	PSNR Absoluto	PSNR Relativo
Tráfego 1	26,5	30,7
Tráfego 2	28,6	32,3

Tabela 5.19: PSNR absoluto e relativo - Compressão VQ3DN - Sequências Tráfego

Tal como para o caso das sequências padrão já analisadas atrás, também neste caso, a influência do *background* é menor do que no caso das imagens submarinas, pelo que também aqui os valores do PSNR absoluto são mais elevados, quando comparados com os das sequências submarinas.

Para se poder aferir também a influência somente da compressão, com quantização vectorial multi-bloco, sobre a qualidade das imagens, mostram-se nas figuras 5.41 e 5.42 os gráficos do PSNR relativo para estas imagens.

Na tabela 5.19 resumem-se para estas sequências os valores do PSNR, quer absoluto, quer relativo.

5.6 Estabilidade da Imagem de Referência e Estabilidade dos *Codebooks versus* Taxa de Compressão

Um factor importante e que à primeira vista poderia causar um *overhead* significativo, devendo como tal ser bem estudado, seria a necessidade de se gerarem novos

codebooks, quer de cinzentos, com um só ou com vários planos, quer de índices e retransmiti-los ao decodificador.

Se fosse necessário efectuar frequentemente este refrescamento de *codebooks* o ganho obtido com a opção de se utilizar compressão vectorial multi-bloco perder-se-ia.

Outro importante aspecto a considerar seria a frequência com que tinha de se transmitir uma nova imagem de referência. Recorde-se que, neste algoritmo, uma imagem é transmitida na totalidade e, de modo a fazer-se a reconstrução, sobre ela são justapostos os blocos das outras imagens. Se fosse necessário retransmitir esta imagem frequentemente perdiam-se aqui igualmente as vantagens previstas.

Relacionado com estas questões não foi implementado, mas facilmente o poderia ter sido, um mecanismo baseado num esquema de retroalimentação do codificador que lhe permitisse inferir da necessidade ou de retransmitir a imagem de referência ou de gerar e retransmitir novos *codebooks*. Por exemplo, o codificador poderia regularmente proceder à reconstrução de imagens e com base nelas e nas originais, calcular o PSNR e por comparação com um limiar pré-estabelecido optar ou por continuar ou por transmitir uma nova imagem de referência e/ou então gerar e transmitir novos *codebooks*.

5.6.1 Refrescamento da Imagem de Referência

É evidente que, quanto mais vezes se transmitir a imagem de referência piores serão os resultados obtidos em termos de taxas de compressão, pelo que é desejável, considerando esta questão só deste ponto de vista, fazê-lo o menos frequentemente possível. No entanto, o factor qualidade das imagens não é menos importante e estará inversamente relacionado com o da taxa de compressão, isto é, teoricamente quanto mais vezes se transmitir a imagem de referência, melhor será a qualidade visual das imagens reconstruídas. Assim, é necessário estabelecer um compromisso entre estas duas imposições.

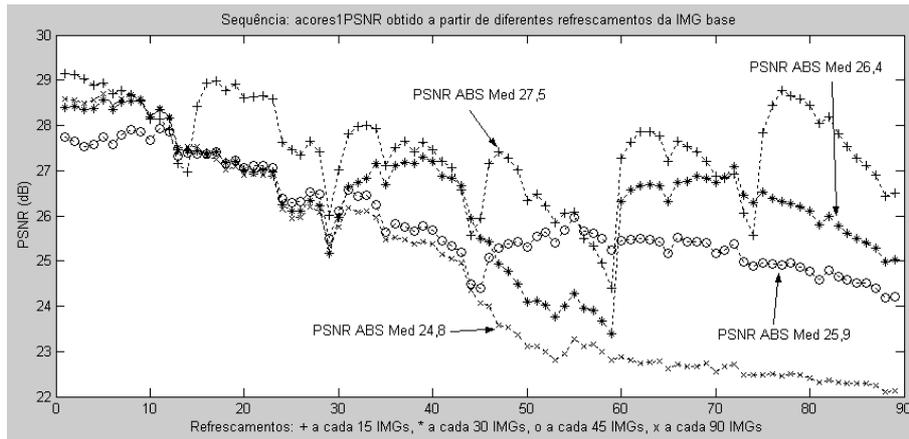


Figura 5.43: PSNR Açores 1 - Diferentes refrescamentos da imagem de referência
Compressão VQ2.5DN e VQ3DN

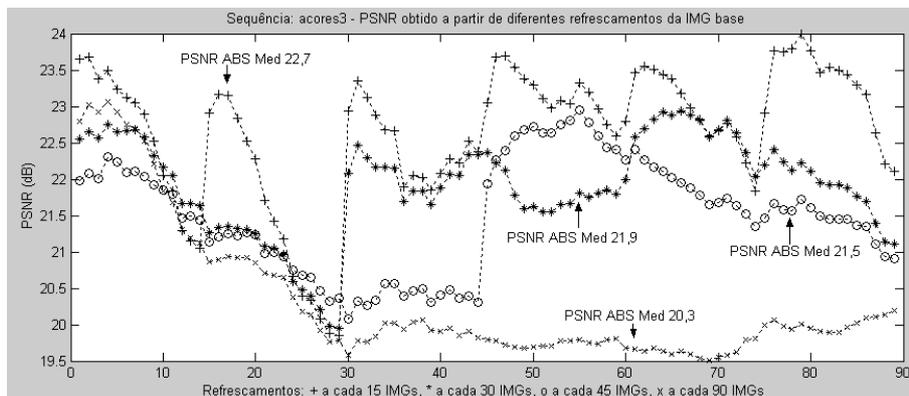


Figura 5.44: PSNR Açores 3 - Diferentes refrescamentos da imagem de referência
Compressão VQ2.5DN e VQ3DN

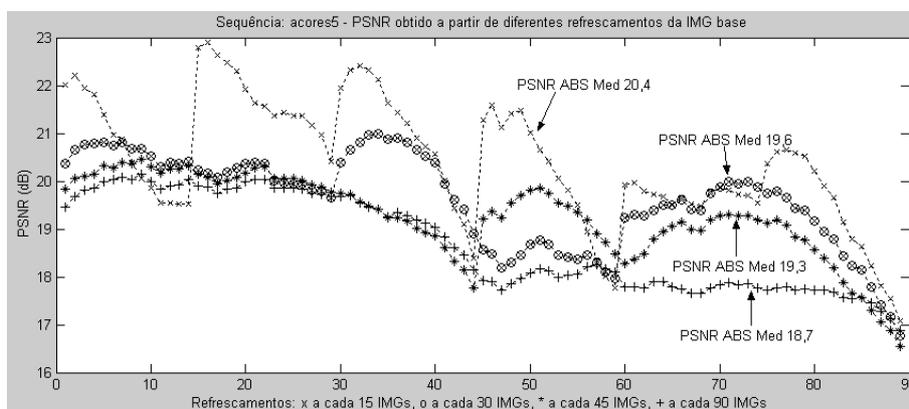


Figura 5.45: PSNR Açores 5 - Diferentes refrescamentos da imagem de referência
Compressão VQ2.5DN e VQ3DN

Refreshamento a cada	Taxa de Compressão Nominal	Taxa de Transmissão (bit/s)	PSNR Absoluto
15	1/361	8.251	27,5
30	1/353	8.609	26,4
45	1/362	8.397	25,9
90	1/393	7.733	24,2

Tabela 5.20: Compressão nominal *versus* refreshamento imagens base - Açores 1
Compressão VQ3DN

Refreshamento a cada	Taxa de Compressão Nominal	Taxa de Transmissão (bit/s)	PSNR Absoluto
15	1/350	8.689	22,7
30	1/335	9.080	21,9
45	1/331	9.180	21,5
90	1/336	9.038	20,0

Tabela 5.21: Compressão nominal *versus* refreshamento imagens base - Açores 3
Compressão VQ3DN

Nos testes realizados com as seqüências de vídeo submarino considerou-se a transmissão de uma nova imagem de referência a cada 15, 30, 45 ou 90 imagens de modo a analisar a influência que este factor tem, quer na taxa de compressão, quer na qualidade das imagens, neste caso aferida com base no PSNR.

Nas figuras 5.43 a 5.45, são mostrados os gráficos em termos do PSNR para 3 seqüências de imagens, considerando na compressão refreshamentos a cada 15, 30, 45 ou 90 imagens. O método aqui empregue foi o VQ3DN para 15 imagens em simultâneo.

Nas tabelas 5.20 a 5.22, mostram-se as taxas de compressão obtidas para essas 3 seqüências e para os 4 casos considerados de refreshamento da imagem de referência. Designamos estas taxas de compressão por taxas de compressão nominais pois elas referem-se só à compressão considerando-se somente as outras imagens que não são de referência.

Por oposição às taxas de compressão nominais, nas tabelas 5.23 a 5.25 consideraram-se as taxas de compressão e de transmissão efectiva, isto é, as obtidas considerando-se também a imagem de referência.

Note-se que, mesmo para o caso de se transmitir uma imagem de referência a

Refrescamento a cada	Taxa de Compressão Nominal	Taxa de Transmissão (bit/s)	PSNR Absoluto
15	1/390	7.803	20,4
30	1/420	7.245	19,6
45	1/463	6.575	19,3
90	1/451	6.733	18,4

Tabela 5.22: Compressão nominal *versus* refrescamento imagens base - Açores 5
Compressão VQ3DN

Refresca a cada	Número Total de bits	Número de bits da Imagem de Referência	Número de bits das Outras Imagens	Número de bits a Transmitir	Taxa de Compressão Efectiva	Taxa de Transmissão Efectiva (bits/s)
15	18247680	271.968	49.504	321.472	1/57	53.579
30	18247680	131.408	51.656	183.064	1/100	30.511
45	18247680	87.176	50.384	137.560	1/133	22.927
90	18247680	40.520	46.736	87.256	1/209	14.543

Tabela 5.23: Compressão efectiva *versus* refrescamento imagens base - Açores 1
Compressão VQ3DN

Refresca a cada	Número Total de bits	Número de bits da Imagem de Referência	Número de bits das Outras Imagens	Número de bits a Transmitir	Taxa de Compressão Efectiva	Taxa de Transmissão Efectiva (bits/s)
15	18247680	379.952	52.136	432.088	1/42	72.015
30	18247680	186.520	54.480	241.000	1/76	40.167
45	18247680	122.264	55.080	177.344	1/103	29.557
90	18247680	60.608	54.000	114.608	1/159	19.101

Tabela 5.24: Compressão efectiva *versus* refrescamento imagens base - Açores 3
Compressão VQ3DN

Refresca a cada	Número Total de bits	Número de bits da Imagem de Referência	Número de bits das Outras Imagens	Número de bits a Transmitir	Taxa de Compressão Efectiva	Taxa de Transmissão Efectiva (bits/s)
15	18247680	455.648	46.816	502.464	1/36	83.744
30	18247680	226.616	43.472	270.088	1/68	45.015
45	18247680	150.264	39.448	189.712	1/96	31.619
90	18247680	74.792	40.424	115.216	1/158	19.202

Tabela 5.25: Compressão efectiva *versus* refrescamento imagens base - Açores 5
Compressão VQ3DN

cada 90 imagens, a qualidade das imagens reconstruídas, considerando o PSNR, mantém-se aceitável. Por outro lado, a taxa de compressão obtida induz uma taxa de transmissão que se mantém abaixo do limite superior especificado. Relembre-se que a análise visual feita pelos 6 avaliadores e referida na secção 5.4.3, incidiu sobre sequências reconstruídas onde o refrescamento foi feito a cada 90 imagens e mesmo assim essa avaliação foi globalmente satisfatória, como constatado.

Esta análise sobre o refrescamento das imagens de base reforça a análise anteriormente feita em termos de conteúdo de imagens, em que referimos que a relativa degradação do PSNR absoluto (e não do relativo) se deve essencialmente às disparidades que possam existir entre as imagens originais e as de referência utilizadas para preencher o *background*.

As observações feitas não serão obviamente válidas se houver uma constante alteração de cena, mas esta situação não se enquadra com o cenário de aplicabilidade que foi inicialmente traçado.

Estas considerações sobre o refrescamento das imagens base não são tão prementes para o caso dos outros tipos de imagens considerados, as imagens padrão e as imagens de tráfego. Principalmente neste último tipo, a imagem de referência pode manter-se por bastante tempo pois o *background* é praticamente imutável.

5.6.2 Refrescamento dos *Codebooks*

Para os *codebooks* podem fazer-se considerações semelhantes às que se fizeram na secção anterior relativamente às imagens de referência. Como se pode ver nos gráficos apresentados, figuras 5.46 e 5.47, essa influência é relativa e mesmo não actualizando o *codebook* muito frequentemente, a relação sinal-ruído mantém-se aceitável, sendo a degradação nos valores do PSNR provocada mais pelas diferenças devidas ao *background* do que propriamente pela utilização do mesmo *codebook*.

O estudo efectuado consistiu em utilizar-se novos *codebooks*, quer de cinzentos, quer *n-codebooks*, quer *i-codebooks*, a cada 45 ou 90 imagens.

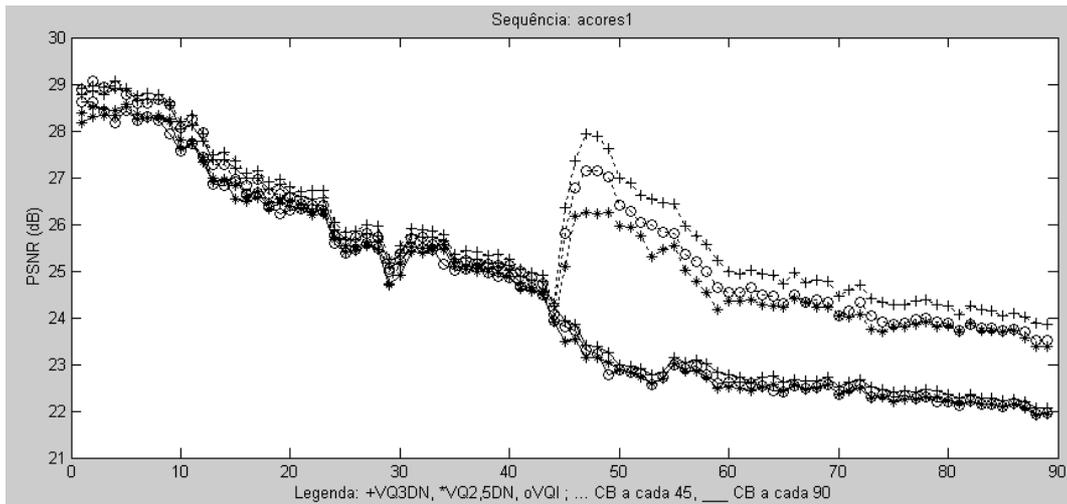


Figura 5.46: PSNR Açores 1 - Diferentes refrescamentos dos *codebooks*
Compressão VQ3DN

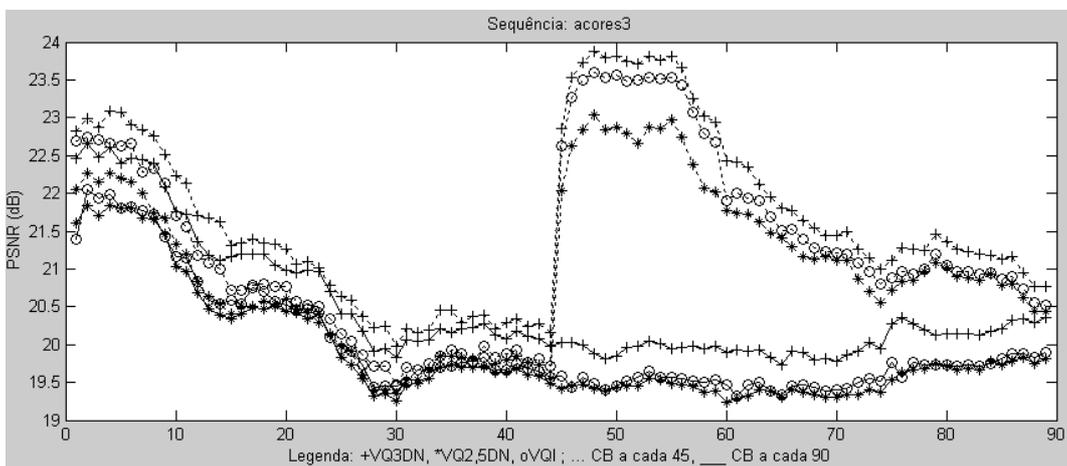


Figura 5.47: PSNR Açores 3 - Diferentes refrescamentos dos *codebooks*
Compressão VQ3DN

Fez-se também uma experiência que consistiu em gerar um *codebook* com base em imagens de três sequências submarinas diferentes e com ele comprimiram-se 18 segundos de vídeo, ou seja 270 imagens dessas 3 sequências. Os resultados foram satisfatórios como se pode ver pelo gráfico do PSNR mostrado na figura 5.48.

Por último, fez-se uma outra experiência que consistiu em utilizar para comprimir uma das sequências submarinas um *codebook* gerado com as outras duas sequências. Os gráficos do PSNR e os respectivos valores médios são mostrados nas figuras 5.49 a 5.51. Também neste caso, os valores obtidos continuam razoáveis ao contrário do que à primeira vista poderíamos ser levados a supor. Isto reforça a ideia de que

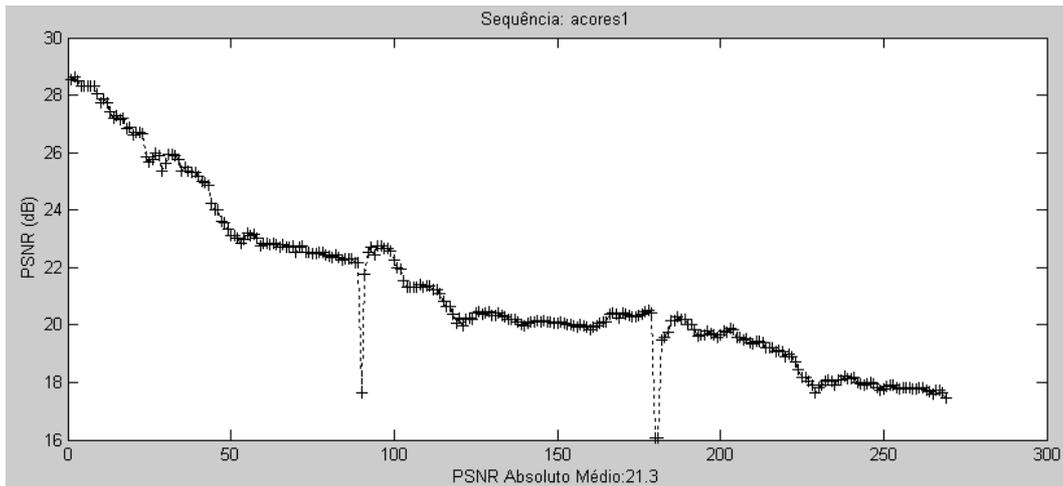


Figura 5.48: PSNR da composição de 3 sequências - Açores 1, 2 e 3
Compressão VQ3DN

a influência do *codebook* na degradação do PSNR não é muito significativa. Caso fosse, haveria uma degradação substancial, quando para uma sequência se utilizasse um *codebook* gerado com imagens de outras sequências do mesmo tipo.

Assim, podemos concluir que o refrescamento do *codebook* não é um factor crítico relativamente à qualidade das imagens reconstruídas.

5.6.3 Compressão Média Total

Nas secções 5.6.1 e 5.6.2, foram introduzidos os conceitos de taxa de compressão nominal e de taxa de compressão efectiva como sendo, respectivamente, a taxa de compressão que se obtém considerando somente as imagens que não as de referência, ou então a que se obtém considerando também as de referência.

Aqui, por analogia, por taxa de compressão média total, estaremos a referirmo-nos à taxa de compressão que se obtém considerando também o *overhead* que pode haver devido à necessidade de reenviar novos *codebooks*.

No entanto, pela análise dos resultados obtidos, feita na secção anterior, concluiu-se não haver necessidade de um refrescamento muito frequente dos *codebooks*. Deste modo, a taxa de compressão média total pode considerar-se equivalente à taxa efectiva referida na secção 5.6.1. Assim, os dados mostrados nas tabelas 5.23 a 5.25

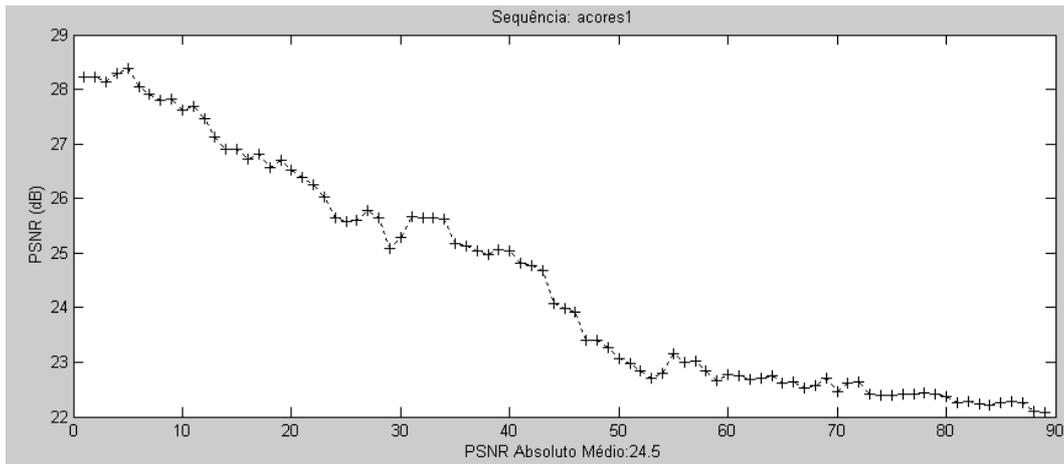


Figura 5.49: PSNR de Açores 1 - Reconstituída com *codebook* de Açores 3 e 5
Compressão VQ3DN

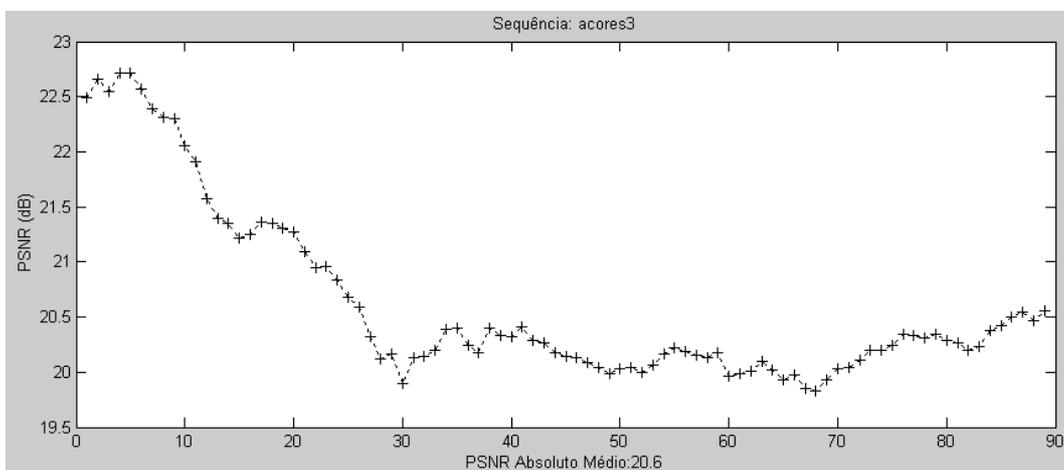


Figura 5.50: PSNR de Açores 3 - Reconstituída com *codebook* de Açores 1 e 5
Compressão VQ3DN

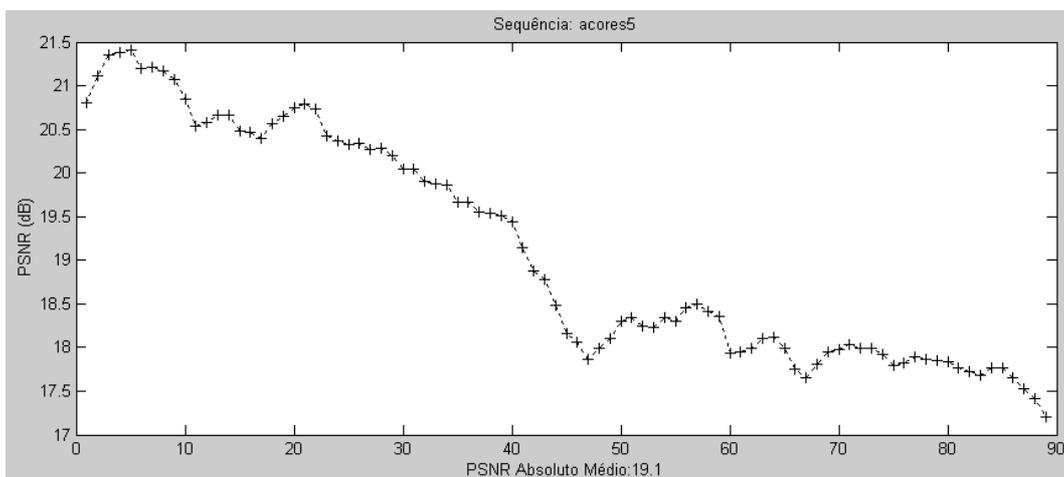


Figura 5.51: PSNR de Açores 5 - Reconstituída com *codebook* de Açores 1 e 3
Compressão VQ3DN

podem ser considerados como a taxa média total.

5.7 *Codebooks* para a Quantização

Dado o interesse central destas simulações ser o de validar os métodos propostos para a segmentação e para a possibilidade de se comprimir mais do que um bloco de imagem em simultâneo, o ênfase dado ao modo como os *codebooks* eram gerados e às eventuais optimizações que se poderiam fazer neste aspecto foi menor. A perspectiva adoptada foi semelhante à que se tomou de não fazer qualquer pré ou pós-processamento das imagens de modo a não mascarar os resultados directamente imputáveis às técnicas propostas.

No entanto, estamos conscientes da importância, já anteriormente referida, dos *codebooks* na qualidade final das sequências reconstruídas, pelo que uma das direcções a explorar como trabalho futuro será exactamente esta. Deverão ser conduzidos estudos no sentido de se melhorar a geração dos *codebooks*, quer sejam os de cinzentos, quer os de índices e no sentido de se estabelecer os melhores parâmetros possíveis para a utilização dos mesmos.

Assim, no que respeita à quantização, e como já referido no capítulo 4, para as simulações efectuadas, os *codebooks* de cinzentos foram treinados utilizando-se o algoritmo LBG [10]. Os *codebooks* de índices foram treinados de um modo ligeiramente diferente, utilizando-se conjuntos de índices para a obtenção do *codebook* final de índices.

Os *codebooks* de cinzentos foram gerados com 256 níveis, tendo cada um dos seus *codevectors* dimensão 2×2 . Esta dimensão dos *codevectors* é a pior do ponto de vista da taxa de compressão, pois conduz a um maior número de *codevectors* e a um conseqüente grande número de índices a transmitir, mas do ponto de vista da qualidade da imagem reconstruída é a melhor passível de se utilizar em quantização vectorial. Como, mesmo usando esta dimensão, o número de bits a transmitir se manteve dentro dos limites de transmissão desejáveis, optou-se por utilizar este

tamanho beneficiando-se deste modo a qualidade das imagens reconstruídas. Caso se pretenda uma maior taxa de compressão, esta pode ser obtida aumentando-se a dimensão dos *codevectors*, o que faz baixar o número de índices a transmitir e consequentemente melhorar a taxa de compressão embora se possa perder qualidade nas imagens reconstruídas.

O algoritmo LBG [5, 10], garante que a distorção de uma iteração para outra não aumenta. No entanto, não garante convergência para uma solução óptima global mas tão somente garante-se uma convergência óptima local para o conjunto de treino utilizado. Atendendo a isto, é evidente que este algoritmo é fortemente dependente do conjunto inicial de treino logo a solução para a qual o algoritmo converge depende bastante das imagens escolhidas para treinar o codebook. De modo a aumentar a variabilidade das combinações possíveis dos níveis de cinzento tendo em atenção esta forte dependência das condições iniciais, as imagens de treino utilizadas foram de dimensão maior que as originais, sendo obtidas pela junção de várias, tipicamente 4, imagens originais.

Com base neste conjunto alargado de treino, para a inicialização do algoritmo utilizou-se a técnica designada por *splitting technique* [10]. Nesta técnica começa-se por considerar um *codebook* de tamanho um, ou seja todo o conjunto de treino, o qual corresponde a um quantizador de um nível, onde o valor calculado para representar esse *codebook*, normalmente designado por *output point*, é o valor médio de todo o conjunto de treino. Este *codebook* inicial vai ser sucessivamente dividido. Para se efectuar esta divisão considera-se um segundo *output point*, o qual é obtido do primeiro pela adição de um vector, fixo ou variável, de perturbação, ϵ . Utiliza-se novamente o algoritmo LBG para obter um quantizador de 2 níveis. Quando o algoritmo convergir, como *codebook* inicial para o quantizador seguinte de 4 níveis, utiliza-se o novo *codebook* de dois níveis e outros dois *output point* obtidos pela adição de ϵ aos dois *output point* resultantes da geração do *codebook* de dois níveis. Deste modo, vai-se duplicando sucessivamente o número de níveis até se atingir o número de níveis final desejado, que neste caso era 256. A inclusão dos sucessivos *codebooks*

na geração por divisão do *codebook* seguinte garante que esse novo *codebook* é pelo menos melhor do que o anterior.

Os *codebooks* de índices foram gerados de um modo similar ao descrito atrás para os *codebooks* de cinzentos, com a diferença de que agora o conjunto de treino são os índices obtidos relativamente a um *codebook* de cinzentos. O tamanho dos *codebooks* de índices é proporcional ao do *codebook* de cinzento, tipicamente 4 vezes maiores, ou seja, de dimensão 1024. Os conjuntos de índices iniciais a partir dos quais se obteve o *codebook* de índices foram obtidos considerando-se 15 imagens da sequência. Dadas as considerações efectuadas secção 4.4.5 sobre a escalabilidade da compressão com índices, os *codevectors* utilizados tinham dimensão 15.

A aplicação desenvolvida está preparada para aceitar outros valores para a geração dos *codebooks*, quer de cinzentos quer de índices.

Normalmente, na geração de *codebooks* a partir de imagens tem-se em consideração o facto de que diferentes valores de iluminação no *background* podem resultar em muitos diferentes *codebooks*. Com vista a atenuar este efeito, pode-se remover a média de cada vector antes da quantização. Esta média e o vector a que se removeu a média são então quantizados separadamente. Para a média utiliza-se normalmente quantização escalar.

No caso presente, consideramos que o ambiente de obtenção das imagens submarinas era constante em termos de iluminação do *background* pelo que não foi utilizada esta possibilidade de codificação separada da média. No entanto esta possibilidade fica em aberto, e em trabalho futuro dever-se-à estudar o interesse da sua utilização.

5.8 Sumário

Neste capítulo, apresentaram-se resultados de simulações efectuadas com a utilização de uma aplicação desenvolvida para implementação dos algoritmos propostos.

Mostraram-se várias figuras onde se pode avaliar a segmentação efectuada e também se mostraram imagens reconstruídas, podendo aferir-se a sua qualidade,

quer em termos visuais, quer em termos da relação sinal-ruído, por análise de gráficos do PSNR dessas sequências. Para apoiar a análise visual, a qual é sempre subjectiva, apresentaram-se os resultados de uma avaliação qualitativa feita por 6 indivíduos.

Dos resultados apresentados, conclui-se que os algoritmos propostos permitem a compressão das sequências submarinas com altas taxas de compressão que possibilitam a sua transmissão através do canal acústico submarino e garantindo a reconstrução dessas imagens com uma qualidade aceitável do ponto de vista do utilizador final.

Se bem que os algoritmos não tenham sido desenhados para serem de aplicação geral, i.e., a qualquer tipo de sequências vídeo foram, a título prospectivo, também aplicados a outras sequências vídeo que não as do ambiente submarino original. Assim, apresentaram-se também alguns resultados da aplicação a sequências padrão e de tráfego. Os resultados obtidos com as sequências padrão e com as sequências de tráfego são encorajadores de um estudo visando a adaptação destes algoritmos a este tipo de sequências vídeo.

Apresentaram-se ainda os resultados de estudos efectuados para analisar a frequência com que é necessário fazer-se o refrescamento das imagens de referência e dos *codebooks*, bem como a influência deste refrescamento e respectiva frequência na qualidade das imagens reconstruídas. Com base neste estudo, definiram-se as taxas de compressão nominal e efectiva. Concluiu-se que, embora sendo estes refrescamentos factores importantes, não têm uma influência tal que façam baixar as taxas de compressão para níveis não compatíveis com a transmissão via canal acústico submarino.

Capítulo 6

Conclusões

6.1 Resumo

Nesta tese foi abordado o tema da compressão de vídeo submarino tendo por objectivo a sua transmissão através do canal acústico submarino. Nas imagens das sequências de vídeo de interesse nota-se actividade hidrotermal, que se manifesta pela emissão de gases deslocando-se para a superfície em colunas ascendentes de bolhas.

Devido ao facto de a taxa de transmissão máxima através deste canal ser bastante baixa, ($20 - 30Kbps$), é necessário proceder-se a uma compressão massiva das sequências. No entanto, como as imagens a transmitir têm características muito específicas, contraste, velocidade e transparência dos elementos que se deslocam, as bolhas gasosas, não é viável a utilização, com bons resultados, de algoritmos clássicos de compressão de vídeo.

Para se ultrapassarem estas dificuldades desenvolveu-se um novo método de compressão, o qual consiste em duas fases:

- Segmentação das imagens em zonas com bolhas e zonas sem bolhas gasosas;
- Compressão das zonas de imagem contendo bolhas.

A segmentação das imagens é efectuada utilizando-se filtros de Gabor adaptados para a detecção das colunas de bolhas. Através desta adaptação consegue-se fazer

uma detecção eficiente dos movimentos verticais relativos às colunas de bolhas e como tal proceder a uma segmentação adequada das zonas de interesse.

A compressão é então efectuada sobre estas zonas, utilizando-se para isso quantização vectorial modificada. Os *codebooks* da quantização vectorial são modificados para possibilitar a compressão de mais do que um bloco de imagem em simultâneo. De acordo com a modificação efectuada, podemos fazer a compressão de 4 maneiras alternativas: compressão com quantização vectorial 2.5D, compressão com quantização vectorial 3D, compressão com quantização vectorial com *codebooks* de cinzentos e de índices e compressão com quantização vectorial e Lempel-Ziv.

Pelos bons resultados obtidos da aplicação desta técnica a sequências de vídeo reais, pode concluir-se que o método proposto funciona adequadamente pois permitiu obter altas taxas de compressão mantendo uma boa qualidade das imagens reconstruídas.

Com o objectivo de se testar a possibilidade de adaptação do método proposto a outro tipo de sequências de vídeo, que não somente as de vídeo submarino, efectuaram-se simulações com sequências de vídeo padrão e de tráfego. Também para estes tipos de sequências se obtiveram bons resultados.

6.2 Principais Contribuições

Neste trabalho propôs-se a utilização dos filtros direccionais de Gabor numa nova perspectiva. Com este tipo de filtros é possível determinar a velocidade de cada *pixel* da imagem e com base nesta informação pode-se, por agrupamento de *pixels* com velocidades similares, segmentar-se as imagens. Na técnica alternativa proposta, determina-se a direcção do movimento principal num dado bloco de imagem e desse modo seleccionam-se a direcção de interesse para a segmentação das imagens. Assim, consegue-se seleccionar somente os blocos da imagem que contém bolhas gasosas, as quais terão, obviamente, um movimento ascendente vertical ou quase vertical. Estes blocos são seleccionados para transmissão para a superfície.

Para compressão dos blocos de imagem seleccionados para transmissão propuseram-se técnicas alternativas de compressão. As técnicas propostas são inovadoras no sentido em que, ao contrário das técnicas de compressão baseadas em quantização vectorial clássica na qual só é possível a compressão de um bloco de imagem de cada vez, nestas é possível a compressão de mais do que um bloco de imagem em simultâneo.

Na primeira destas técnicas utilizam-se *codebooks*, diferentes dos clássicos, pois têm mais do que um plano, que designamos por *n-codebooks*. Com estes *n-codebooks* podem-se comprimir, em simultâneo, tantos blocos de imagem quanto o número de planos do *n-codebook*. Foram implementadas duas variantes desta técnica cuja diferença reside no modo como os *codebooks* são gerados, resultando em compressão baseada em *n-codebooks* 2.5D ou 3D.

Na segunda das técnicas propostas, designada por compressão baseada em *i-codebooks*, para além do *codebook* de níveis de cinzentos propôs-se a utilização de um novo *codebook*, um *codebook* de índices, o *i-codebook*. Este *i-codebook* permite codificar em simultâneo n índices, referentes a n blocos de imagens, obtidos por consulta do *codebook* de níveis de cinzento.

Finalmente foi proposta uma outra técnica, implementada apenas com a finalidade de se obterem dados comparativos, onde, para além da quantização vectorial, se utiliza a compressão por dicionário adaptativo baseada no algoritmo de Lempel-Ziv. Nesta técnica, determinam-se os índices referentes à quantização vectorial de n blocos de imagem e depois aplica-se a esses índices compressão baseada em dicionário adaptativo.

Refira-se que, de modo a poder-se avaliar correctamente o desempenho dos algoritmos propostos, não foi feito nenhum pré ou pós-processamento das imagens originais ou reconstruídas. No entanto, isso é possível e numa aplicação real tal deverá ser feito. Com isso, pode-se em princípio obter ganhos substanciais, sobretudo no que respeita à qualidade visual das imagens reconstruídas.

6.3 Trabalho Futuro

O método proposto poderá obviamente ser melhorado e otimizado, bem como pode ser estendido a outro tipo de imagens que não somente imagens submarinas com registo de actividade hidrotermal.

Assim, como trabalho futuro, perspectiva-se:

- Segmentação das imagens com base em filtros de Gabor, mas possibilitando a segmentação em blocos de tamanho e morfologia diferentes de acordo com características locais da imagem e/ou do movimento;
- Considerar, numa mesma imagem e para diferentes zonas da mesma, blocos de tamanho e/ou morfologia diferente, por exemplo quadrados e/ou rectângulos, o que poderia possibilitar a optimização da compressão;
- Utilizar para a determinação dos blocos com movimento, técnicas de processamento de imagem em pirâmide;
- Utilizar, dado o algoritmo ser altamente paralelizável, processamento paralelo, de modo a melhorar a velocidade de execução, nomeadamente, nas fases de segmentação e de compressão;
- Estudar a utilização de técnicas alternativas de geração dos *codebooks* e estudo dos melhores parâmetros de utilização da quantização vectorial, de modo a tentar melhorar ainda mais a fidelidade das imagens reconstruídas;
- Estudar alternativas para a geração dos *i-codebooks* ou seja para a obtenção das possíveis combinações mais usuais de índices que não só as baseadas no algoritmo LBG como implementado;
- Implementar um mecanismo de auto-controlo da qualidade das imagens reconstruídas, o qual poderia tomar decisões, tais como, o número de imagens a comprimir em simultâneo, o refrescamento das imagens de referência e/ou o refrescamento dos *codebooks*;

- Implementar algoritmos de pré e pós-processamento das imagens originais e reconstruídas, de modo a melhorar as taxas de compressão e a qualidade visual.

Como alternativas de aplicação do método proposto pensamos ser altamente viável e proveitosa a sua aplicação à monitorização de segurança, através da adaptação do algoritmo a situações de vídeo-vigilância, bem como ao controlo de tráfego rodoviário. As experiências bastante positivas que foram efectuadas e que são apresentadas em 5.5, demonstram a viabilidade da utilização do método para outros tipos de imagens, pelo que consideraremos estas aplicações alternativas em futuro trabalho de investigação.

A utilização do algoritmo proposto para a compressão de imagens coloridas será também considerado em futura investigação, a desenvolver no seguimento deste trabalho, pois pensamos que esta extensão é perfeitamente viável.

Bibliografia

- [1] J. A. J. Roufs, "Perceptual image quality: Concept and measurement," *Philips Journal of Research*, vol. 47, pp. 35–62, 1992.
- [2] G. S. Cox , G. de Jager, "Template matching with invariance," in *Proceedings of the Fourth South African Workshop on Pattern Recognition*, pp. 152–6, Nov. 1993.
- [3] D. L. Wilson, A. J. Baddeley , R. A. Owens, "A new metric for grey-scale image comparison," *International Journal of Computer Vision*, vol. 24:, pp. 5–17, 1997.
- [4] R. J. Clarke, *Digital Compression of Still Images and Video*. Academic Press, 1995.
- [5] K. Sayood, *Data Compression*. Morgan Kaufmann Publishers, Inc, 1996.
- [6] K. R. Rao , J. Hwang, *Techniques & Standars for Image, Video and Audio Coding*. Prentice Hall, 1996.
- [7] L. Thede , S. Kwatra, "Image sequence coding by vector quantization," in *Proceedings of ICASSP'89*, 1989.
- [8] N. M. Nasrabadi , R. A. King, "Image coding using vector quantization: A Review," *IEEE Transactions on Communications*, vol. 36, pp. 957–971, 1988.
- [9] W. H. Equitz, "A new vector quantization clustering algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 1568–1575, 1989.
- [10] Y. Linde, A. Buzo , R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, pp. 84–95, 1980.

- [11] S. P. Lloyd, “Least squares quantization in PCM,” *IEEE Transactions on Information Theory*, vol. 28, pp. 127–135, 1982.
- [12] K. N. Ngan , W. L. Chooi, “Very low bit rate video coding using 3D subband approach,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, pp. 309–316, 1994.
- [13] K. Cinkler, “Very low bit-rate wavelet video coding,” *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 4–11, 1998.
- [14] A. Graps, “An introduction to wavelets,” *IEEE Computacional Science and Engineering*, vol. 2, 1995.
- [15] J. Roese, W. Pratt , G. Robinson, “Interframe cosine transform image coding,” *IEEE Transactions on Communications*, vol. 25, 1977.
- [16] J. Miano, *Compressed Image File Formats*. Addison Wesley, 1999.
- [17] C. Shannon , W. Weaver, *The Mathematical Theory of Communication*. University of Illinois Press, 1998.
- [18] S. J. Solari, *Digital and Video Compression*. McGraw-Hill, 1997.
- [19] P. Symes, *Video Compression Demystified*. McGraw-Hill, 2001.
- [20] P. Wayner, *Compression algorithms for real programmers*. Morgan Kaufmann, 2000.
- [21] A. E. Jacquin, “Fractal image coding: A Review,” *Proceedings of the IEEE*, vol. 81, pp. 1451–1465, 1993.
- [22] Y.-Q. Zhang, F. Pereira, T. Sikora, , C. Reader, “Special issue on MPEG-4,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, 1997.
- [23] K. M. Ferreira, “Compressão de vídeo para canais de banda estreita,” Master’s thesis, Instituto Superior Técnico - Universidade Técnica de Lisboa, 1996.
- [24] I. E. G. Richardson, *Video Codec Design*. John Wiley & Sons, 2003.
- [25] A. H. Sadka, *Compressed Video Communications*. John Wiley & Sons, 2002.

- [26] T. Ebrahimi , E. Reusens, “New trends in very low bitrate video coding,” *Proceedings of the IEEE*, vol. 83, pp. 877–891, 1995.
- [27] O. Egger, E. Reusens, T. Ebrahimi, , M. Kunt, “Very low bit rate coding of visual information: A Review,” in *IEEE ICAS’95 - Tutorial Book*, 1995.
- [28] H. Li, A. Lundmark , R. Forcheimer, “Image sequence coding at very low bit rate: A Review,” *IEEE Transactions on Image Processing*, vol. 3, pp. 589–609, 1994.
- [29] C.-W. Ku, *An Object-Based Very Low Bit-Rate Video Coding System*. PhD thesis, Department of Electrical Engineering, National Taiwan University, Taiwan, 1997.
- [30] C. S. Choi, H. Harashima , T. Takebe, “Analysis and synthesis of facial expressions in model-based image coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, 1994.
- [31] D. J. Heeger, “Optical flow using spatiotemporal filters,” *International Journal of Computer Vision*, pp. 279–295, 1988.
- [32] E. H. Adelson , J. R. Bergen, “Spatiotemporal energy models for the perception of motion,” *J. Opt. Soc. Am.*, vol. A/Vol. 2, pp. 284–299, 1985.
- [33] T. R. Reed , A. E. Soohoo, “Very-low-bit-rate coding of image sequences using the Gabor transform,” *Journal of the SID*, pp. 77–81, 1995.
- [34] J. Y. A. Wang , E. H. Adelson, “Spatio-temporal segmentation of video data,” in *Proceedings of SPIE: Image and Video Processing II*, February 1994.
- [35] J. Huguet , L. Torres, “Vector quantization in image sequence coding,” in *Proceedings of EUSIPCO’90*, 1990. Barcelona, Spain, September 18-21.
- [36] M. Nelson , J.-L. Gailly, *The Data Compression Book, 2nd*. Hungry Minds, Inc, 1995.
- [37] R. Lewis, *Practical Digital Image Processing*. Ellis Horwood, 1990.
- [38] M. Rabbani , P. W. Jones, *Digital Image Compression Techniques*. Spie Optical Engineering Press, 1995.

- [39] A. Kokaram, *Motion Picture Restoration*. Springer-Verlag, 1998.
- [40] S. J. Sangwine , E. R. E. N. Horne, *The Colour Image Processing Handbook*. Chapman & Hall, 1998.
- [41] A. M. Tekalp, *Digital Video Processing*. Prentice Hall, 1995.
- [42] M. Waldowski, “A new segmentation algorithm for videophone applications based on stereo image pairs,” *IEEE Transactions on Communications*, vol. 39, pp. 1856–1868, 1991.
- [43] M. Irani, B. Rousso , S. Peleg, “Computing occluding and transparent motions,” *International Journal of Computer Vision*, vol. 12:, pp. 5–16, 1994.
- [44] J. R. Bergen, P. J. Burt , S. Peleg, “A three-frame algorithm for estimating two-componente image motion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 886–896, 1992.
- [45] J. M. F. Moura, R. S. Jasinski, H. Shiojiri , J.-C. Lin, “Video over wireless,” *IEEE Personal Communications*, pp. 44–54, 1996.
- [46] L. Torres , J. Huguet, “An improvement on codebook search for vector quantization,” *IEEE Transactions on Communications*, vol. 42, pp. 208–210, 1994.
- [47] M. Kunt, M. Bénard , R. Leonardi, “Recent results in high-compression image coding,” *IEEE TRansactions on Circuits and Systems*, vol. 34, pp. 1306–1336, 1987.
- [48] D. F. Hoag, V. K. Ingle , R. J. Gaudette, “Low-bit-rate coding of underwater video using wavelet-based compression algorithms,” *IEEE Journal of Oceanic Engineering*, vol. 22, pp. 393–400, 1997.
- [49] S. Yin, “Image compression and transmission trough a low-rate ultrasonic link in subsea telerobotic applications,” *Journal of Mathematical Imaging and Vision*, vol. 7, pp. 41–52, 1997.
- [50] M. Porat , Y. Y. Zeevi, “The generalized Gabor scheme of image representation in biological and machine vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 452–468, 1988.

- [51] K.-H. Tzou, H. G. Husmann , K. Aizawa, “Special issue on very low bit rate video coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, 1994.
- [52] D. F. Hoag , V. K. Ingle, “Underwater video compression using the wavelet transform,” in *Proceedings of Oceans’95*, September 1995. San Diego, C.A., USA.
- [53] V. A. Christopoulos , J. Cornelis, “A low bit rate segmented video codec with hybrid motion estimation and inherent bit rate control capability,” in *Proceedings of ICASSP’98*, May 12-15 1998. Seattle, Washington, USA.
- [54] J. Barbosa , V. Barroso, “Very low bit rate video compression for acoustic transmission of subsea images of hydrothermal vents,” in *Proceedings of Oceans’2000*, September 10-14 2000. Providence, Rhode Island, USA.
- [55] J. Barbosa , V. Barroso, “Underwater video compression based on Gabor filters,” in *Actas da ConfTele’99*, Abril 15-16 1999. Sesimbra, Portugal.
- [56] J. Barbosa , V. Barroso, “Multiple and simultaneous image compression for video sequences,” in *Actas da ConfTele’01*, Abril 23-24 2001. Figueira da Foz, Portugal.
- [57] R. S. Jasinchi , J. M. F. de Moura, “Content-based video sequence representation,” in *Proceedings of ICIP’95*, 1995.
- [58] L. A. P. de Meneses Corte-Real, *Quantificação Vectorial Aplicada a Vídeo Digital de Baixa Cadência*. PhD thesis, Faculdade de Engenharia, Universidade do Porto, 1993.
- [59] G. K. Wallace, “The JPEG still picture compression standard.” Digital Equipment Corporation, Submitted in December 1991 for publication in *IEEE Transactions on Consumer Electronics*, 1991.
- [60] F. Dufaux , F. Moscheni, “Background mosaicking for low bit rate video coding.” Digital Equipment Corporation, Cambridge Research Lab., 1997.