

Mónica Isabel Teixeira da Costa

**TÉCNICAS DE RECONSTRUÇÃO DE SINAL
APLICADAS A SEQUÊNCIAS DE ALTA
RESOLUÇÃO TEMPORAL**

Departamento de Engenharias
Universidade de Trás-os-Montes e Alto Douro
Vila Real, 2005

Dissertação submetida por Mónica Isabel Teixeira da Costa à Universidade de Trás-os-Montes e Alto Douro para obtenção do grau de Mestre em Tecnologias das Engenharias, sob a orientação do Prof. Doutor João Manuel Pereira Barroso, Prof. Auxiliar do Departamento de Engenharias da Universidade de Trás-os-Montes e Alto Douro, e sob a co-orientação do Prof. Doutor Salviano Filipe Silva Pinto Soares, Prof. Auxiliar do Departamento de Engenharias da Universidade de Trás-os-Montes e Alto Douro.

À minha Família

AGRADECIMENTOS

A autora deseja expressar o seu mais sincero agradecimento a todas as instituições e pessoas que, com a sua valiosa colaboração, contribuíram para a realização deste trabalho.

Ao Professor Doutor João Manuel Pereira Barroso e ao Professor Doutor Salviano Filipe Silva Pinto Soares, agradeço a orientação científica, o incentivo, as sugestões e discussões sempre oportunas, a confiança e apoio constantes ao longo de todo o trabalho. Fico-lhes também muito grata pelas sugestões feitas durante a escrita da dissertação e revisão final.

Ao Professor Doutor Victor Manuel de Jesus Filipe da Universidade de Trás-os-Montes e Alto Douro e ao Professor Doutor Paulo Jorge Ferreira da Universidade de Aveiro, agradeço o apoio e colaboração prestada sempre que foi necessário.

Aos colegas do Departamento de Engenharia Informática da EST – IPCB, em especial ao Professor Coordenador Eurico Ribeiro Lopes e ao Professor Adjunto José Luís Cruz agradeço todas as facilidades concedidas, para que este trabalho se tornasse realidade.

À Dra. Sandra Sofia Martinho Pereira agradeço a colaboração prestada durante a fase de verificação ortográfica da dissertação.

Aos meus pais e ao meu marido, agradeço o carinho e o apoio constantes durante a realização deste trabalho e não só.

A todos aqueles que, de forma directa ou indirecta, contribuíram para a realização deste trabalho, o meu sincero obrigado.

Vila Real, Dezembro de 2005

Mónica Isabel Teixeira da Costa

RESUMO

Nesta dissertação conjugam-se técnicas de compressão de sinal, designadamente imagens, com técnicas de reconstrução de sinal. O objectivo consiste em avaliar uma alternativa a métodos de compressão tradicionais baseados em previsão de movimento, através da inclusão de técnicas de reconstrução de sinal, para o caso específico das sequências de alta resolução temporal.

Numa sequência de vídeo, existem *pixels* cujo valor não sofre alteração significativa ao longo da sequência. Isto é verdade para taxas de 25 e 30 *frames* por segundo, e é igualmente verdade para taxas superiores, ou seja, em sequências de alta resolução temporal (taxas de *frame* superiores a 50 *frames* por segundo), como é o caso estudado nesta dissertação.

Se o valor de um *pixel* se mantiver constante ao longo da sequência, apenas é necessária uma amostra para representar esse *pixel*. Por outro lado, se o valor de um *pixel* sofrer alteração ao longo da sequência o número de amostras necessárias para representar esse sinal será maior.

Desta forma analisa-se o sinal representativo de cada um dos *pixels* de uma sequência de *frames* de alta resolução temporal (225 *frames* por segundo), desprezando um determinado número de amostras de cada um dos sinais, com taxa de amostragem inicial igual a n . Ao desprezar amostras, o resultado será uma sequência com menor número de *frames*, e consequentemente um ficheiro com tamanho inferior ao da sequência original ($< n$).

A sequência original (ou sinais originais) é obtida novamente aplicando aos sinais subamostrados um algoritmo de reconstrução de sinal, neste caso específico o algoritmo de Papoulis-Gerchberg. O objectivo consiste em recuperar as amostras perdidas/desconhecidas para permitir a reconstrução da sequência original.

A transição entre o caso tratado nesta dissertação e casos mais gerais é facilitada pelo facto do algoritmo de reconstrução usado permitir tratar distribuições irregulares de amostras perdidas.

ABSTRACT

In this dissertation, signal (more concretely images) compression techniques are combined with signal reconstruction techniques. The main objective is to assess an alternative to the traditional compression methods, based on motion prediction, through the inclusion of signal reconstruction methods, in the specific case of high frame rate sequences.

When looking at a video stream, it becomes noticeable that a part of the information remains more or less the same throughout the entire sequence. This is true for frame rates of 25 and 30 frames per second, and it is also true for higher rates such as the case of high frame rate sequences (frame rates higher than 50 frames per second), which is the case studied in this dissertation.

If a pixel value remains constant during the whole sequence, only a sample will be required to represent it. On the other hand, if throughout the sequence the pixel value changes, more samples will be needed to represent it correctly.

The signals that represent each one of the pixels of a high frame rate sequence (225 frames per second) are analyzed, and a certain number of samples of each one of the signals, with a sampling rate equal to n , are obliterated. By removing a certain number of samples the result will be a sequence with an inferior number of frames, and consequently a file with lower size than the original one ($< n$).

The original sequence (or original signals) is obtained by applying, to the down sampled signals, a signal reconstruction algorithm; in this particular case, the Papoulis-Gerchberg algorithm, whose objective is to recover the lost/missing samples, obtaining by this way the original signals and allowing the reconstruction of the original sequence.

The transition from the case related in this dissertation and more general cases is facilitated by the fact that the reconstruction algorithm used, allows the handling of the irregular lost sample distribution.

ÍNDICE

ÍNDICE DE FIGURAS	XI
ÍNDICE DE TABELAS	XV
CAPÍTULO 1 - INTRODUÇÃO	1
1.1 ENQUADRAMENTO.....	1
1.2 MOTIVAÇÃO.....	2
1.3 OBJECTIVOS	3
1.4 ORGANIZAÇÃO DA DISSERTAÇÃO	4
CAPÍTULO 2 - FORMATOS E QUALIDADE DE VÍDEO	5
2.1 CENAS DE VÍDEO NATURAIS.....	5
2.2 AQUISIÇÃO.....	6
2.2.1 Amostragem Espacial.....	6
2.2.2 Amostragem Temporal.....	7
2.2.3 Frames e Campos.....	8
2.3 ESPAÇOS DE COR	9
2.3.1 RGB.....	10
2.3.2 YCbCr.....	10
2.4 FORMATOS DE VÍDEO	11
2.5 QUALIDADE	13
2.5.1 Medida Subjectiva.....	14
2.5.2 Medida Objectiva.....	16
CAPÍTULO 3 - COMPRESSÃO DE IMAGEM E VÍDEO.....	19
3.1 JPEG E JPEG2000	19
3.1.1 JPEG	20
3.1.1.1 DCT – Discrete Cosine Transform.....	21
3.1.1.2 Modo Lossless	23
3.1.1.3 JPEG – LS	23
3.1.1.4 JBIG	23
3.1.1.5 JBIG2	25
3.1.2 JPEG2000.....	26
3.1.2.1 Aplicações, Requerimentos e Características	27
3.1.2.2 Arquitectura do Standard	29
3.2 MPEG	30
3.2.1 MPEG-1	32
3.2.2 MPEG-2	33
3.2.3 MPEG-4	33
3.3 H.261 E H.263.....	36
3.4 H.264/MPEG-4 PART 10	36
3.5 COMPARAÇÃO ENTRE O MPEG-4 VISUAL E O H.264.....	37
3.6 RADIUS CINEPAK CODEC	38
3.7 MICROSOFT RUN LENGTH ENCODING - RLE.....	40
3.7.1 Modo Encoded	40
3.7.2 Modo Absoluto	41
3.8 MICROSOFT VÍDEO – 1.....	41

3.8.1	<i>Dados Video-1 de 8 bits</i>	42
3.8.1.1	Codificação de 2 cores e 8 cores.....	42
3.9	INTEL IYUV	42
CAPÍTULO 4 - ALGORITMO DE RECONSTRUÇÃO		45
4.1	NOTAÇÃO E TERMINOLOGIA	45
4.2	OBJECTIVOS	46
4.3	SINAIS DE BANDA-LIMITADA	48
4.4	O ALGORITMO DE PAPOULIS-GERCHBERG	49
4.5	CONVERGÊNCIA DO ALGORITMO	53
CAPÍTULO 5 - METODOLOGIA APLICADA		55
5.1	EXTRACÇÃO DOS SINAIS DE UMA SEQUÊNCIA	56
5.2	CARACTERÍSTICAS DAS SEQUÊNCIAS	58
5.3	FILTRAGEM DOS SINAIS EXTRAÍDOS.....	58
5.4	DECIMAÇÃO DOS SINAIS	60
5.5	CONDIÇÕES DE CONVERGÊNCIA DO ALGORITMO DE RECONSTRUÇÃO	61
5.6	ANÁLISE DO DESEMPENHO E CRITÉRIOS DE PARAGEM DO ALGORITMO DE RECONSTRUÇÃO	63
5.7	METODOLOGIA UTILIZADA NA GERAÇÃO DOS FICHEIROS AVI	67
CAPÍTULO 6 - RESULTADOS		71
6.1	SEQUÊNCIA 1 – RESULTADOS PARA BANDA DE 32 AMOSTRAS	71
6.1.1	<i>Sequência 1 – Reconstrução a partir da sequência de 128 frames</i>	73
6.1.2	<i>Sequência 1 – Reconstrução a partir da sequência de 256 frames</i>	77
6.1.3	<i>Sequência 1 – Reconstrução a partir da sequência de 512 frames</i>	80
6.2	SEQUÊNCIA 1 – RESULTADOS PARA BANDA DE 64 AMOSTRAS	83
6.2.1	<i>Sequência 1 – Reconstrução a partir da sequência de 256 frames</i>	83
6.2.2	<i>Sequência 1 – Reconstrução a partir da sequência de 512 frames</i>	85
6.3	SEQUÊNCIA 1 – RESULTADOS PARA BANDA DE 128 AMOSTRAS	86
6.3.1	<i>Sequência 1 – Reconstrução a partir da sequência de 512 frames</i>	87
6.4	FICHEIROS AVI	89
6.4.1	<i>Sequência 1 Codificada com o método RCC – Resultados para banda de 32 Amostras</i>	90
6.4.1.1	<i>Sequência 1 Codificada com o RCC – Reconstrução a partir da sequência de 128 frames</i>	91
6.4.1.2	<i>Sequência 1 Codificada com o RCC – Reconstrução a partir da sequência de 256 frames</i>	92
6.4.1.3	<i>Sequência 1 Codificada com o RCC – Reconstrução a partir da sequência de 512 frames</i>	93
6.4.2	<i>Sequência 1 Codificada com o método IYUV – Resultados para banda de 32 Amostras</i>	95
6.4.2.1	<i>Sequência 1 Codificada com o IYUV – Reconstrução a partir da sequência de 128 frames</i>	96
6.4.2.2	<i>Sequência 1 Codificada com o IYUV – Reconstrução a partir da sequência de 256 frames</i>	97
6.4.2.3	<i>Sequência 1 Codificada com o IYUV – Reconstrução a partir da sequência de 512 frames</i>	98
6.4.3	<i>Sequência 1 Codificada com o método MV1 – Resultados para banda de 32 Amostras</i>	100
6.4.3.1	<i>Sequência 1 Codificada com o MV1 – Reconstrução a partir da sequência de 128 frames</i>	101
6.4.3.2	<i>Sequência 1 Codificada com o MV1 – Reconstrução a partir da sequência de 256 frames</i>	102
6.4.3.3	<i>Sequência 1 Codificada com o MV1 – Reconstrução a partir da sequência de 512 frames</i>	104
6.4.4	<i>Sequência 1 Codificada com o método RLE – Resultados para banda de 32 Amostras</i>	105
6.4.4.1	<i>Sequência 1 Codificada com o RLE – Reconstrução a partir da sequência de 128 frames</i>	106
6.4.4.2	<i>Sequência 1 Codificada com o RLE – Reconstrução a partir da sequência de 256 frames</i>	107
6.4.4.3	<i>Sequência 1 Codificada com o RLE – Reconstrução a partir da sequência de 512 frames</i>	109
CAPÍTULO 7 - CONCLUSÕES E TRABALHO FUTURO.....		111
ACRÓNIMOS		117
REFERÊNCIAS BIBLIOGRÁFICAS.....		119

ANEXO.....	125
SEQUÊNCIA 2 – RESULTADOS PARA BANDA DE 32 AMOSTRAS	127
<i>Sequência 2 – Reconstrução a partir da sequência de 128 frames.....</i>	<i>129</i>
<i>Sequência 2 – Reconstrução a partir da sequência de 256 frames.....</i>	<i>132</i>
<i>Sequência 2 – Reconstrução a partir da sequência de 512 frames.....</i>	<i>134</i>

ÍNDICE DE FIGURAS

Figura 2.1 – Imagem com 2 grelhas de amostragem.....	6
Figura 2.2 – Imagem amostrada com baixa resolução (grelha de amostragem preta)	7
Figura 2.3 – Imagem amostrada com resolução ligeiramente superior (grelha de amostragem branca) 7	
Figura 2.4 – Sequência de vídeo entrelaçada	8
Figura 2.5 – (a) Campo superior e (b) campo inferior	9
Figura 2.6 – (a) Imagem original e suas componentes (b) vermelha, (c) verde e (d) azul	10
Figura 2.7 – <i>Frame</i> de vídeo amostrado com diversas gamas de resolução	12
Figura 3.1 – Diagramas de blocos do codificador (a) e decodificador (b) JPEG2000.	29
Figura 3.2 – Ladrilhos da imagem, Deslocamento a nível DC para cada ladrilho componente da imagem	30
Figura 3.3 – Formato da <i>source</i>	31
Figura 3.4 – Organização do decodificador MPEG.....	32
Figura 3.5 – Estrutura de um <i>frame</i> típico de uma sequência de vídeo Cinepak	40
Figura 4.1 – Sinal não limitado na frequência (a) e respectiva FFT (b).....	48
Figura 4.2 – Limitação de banda de um sinal utilizando o operador B	49
Figura 4.3 – (a) Sinal $x(t)$ contínuo limitado em frequência; (b) Sinal $x(t)$ discretizado	50
Figura 4.4 – (a) Trem de impulsos para amostrar o sinal $x(n)$; (b) Sinal $x(n)$ corrompido.....	50
Figura 4.5 – Limitação de banda de um sinal.....	52
Figura 4.6 – Substituição das amostras conhecidas no sinal de banda-limitada	53
Figura 5.1 – Dois <i>frames</i> consecutivos e respectiva diferença entre ambos	55
Figura 5.2 – Processo de extração dos sinais de uma sequência	56
Figura 5.3 – Processo para guardar todos os <i>frames</i> da sequência numa única matriz.....	57
Figura 5.4 – Sinal correspondente aos níveis de cinzento de um <i>pixel</i> na sequência.....	57
Figura 5.5 – (a) Sinal original; (b) Sinal observado	59
Figura 5.6 – Resultado da decimação com entrelaçamento igual a dois, sinais com, (a)512 amostras; (b) 256 amostras; (c) 128 amostras.....	60
Figura 5.7 – Processo de decimação da sequência de acordo com a banda de amostras do filtro passa-baixo	63
Figura 5.8 – Resultados da reconstrução de um sinal de: (a) 512 amostras; (b)128 amostras	64
Figura 5.9 – Processo iterativo do algoritmo de reconstrução de sinal.....	66
Figura 5.10 – <i>Frame</i> comprimido e descomprimido com diferentes métodos de compressão	68
Figura 5.11 – Codificação/decodificação de sequências utilizando o formato AVI.....	68

Figura 5.12 – Metodologia de criação dos ficheiros AVI e das novas sequências utilizando métodos de compressão	69
Figura 5.13 – Proposta para aplicação, da metodologia apresentada, em sequências codificadas/descodificadas utilizando o formato AVI	70
Figura 6.1 – Alguns <i>frames</i> da sequência 1 observada, banda de 32 amostras no FPB	72
Figura 6.2 – Sequência 1, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 128 <i>frames</i> (banda de 32 amostras)	74
Figura 6.3 – Sequência 1, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 128 <i>frames</i> (banda de 32 amostras)	74
Figura 6.4 – Alguns <i>frames</i> da sequência 1 reconstruída a partir da sequência de 128 <i>frames</i> , banda de 32 amostras no FPB	76
Figura 6.5 – Sequência 1, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 256 <i>frames</i> (banda de 32 amostras)	77
Figura 6.6 – Sequência 1, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 256 <i>frames</i> (banda de 32 amostras)	77
Figura 6.7 – Alguns <i>frames</i> da sequência 1 reconstruída a partir da sequência de 256 <i>frames</i> , banda de 32 amostras no FPB	79
Figura 6.8 – Sequência 1, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 512 <i>frames</i> (banda de 32 amostras)	80
Figura 6.9 – Sequência 1, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 512 <i>frames</i> (banda de 32 amostras)	80
Figura 6.10 – Alguns <i>frames</i> da sequência 1 reconstruída a partir da sequência de 512 <i>frames</i> , banda de 32 amostras no FPB	82
Figura 6.11 – Sequência 1, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 256 <i>frames</i> (banda de 64 amostras)	84
Figura 6.12 – Sequência 1, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 256 <i>frames</i> (banda de 64 amostras)	84
Figura 6.13 – Sequência 1, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 512 <i>frames</i> (banda de 64 amostras)	85
Figura 6.14 – Sequência 1, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 512 <i>frames</i> (banda de 64 amostras)	85
Figura 6.15 – Sequência 1, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 512 <i>frames</i> (banda de 128 amostras)	87
Figura 6.16 – Sequência 1, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 512 <i>frames</i> (banda de 128 amostras)	87
Figura 6.17 – Sequência 1 codificada com o RCC, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 128 <i>frames</i> (banda de 32 amostras)	91

Figura 6.18 – Sequência 1 codificada com o RCC, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 128 <i>frames</i> (banda de 32 amostras)	91
Figura 6.19 – Sequência 1 codificada com o RCC, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 256 <i>frames</i> (banda de 32 amostras).....	92
Figura 6.20 – Sequência 1 codificada com o RCC, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 256 <i>frames</i> (banda de 32 amostras)	93
Figura 6.21 – Sequência 1 codificada com o RCC, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 512 <i>frames</i> (banda de 32 amostras).....	94
Figura 6.22 – Sequência 1 codificada com o RCC, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 512 <i>frames</i> (banda de 32 amostras)	94
Figura 6.23 – Sequência 1 codificada com o IYUV, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 128 <i>frames</i> (banda de 32 amostras).....	96
Figura 6.24 – Sequência 1 codificada com o IYUV, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 128 <i>frames</i> (banda de 32 amostras)	96
Figura 6.25 – Sequência 1 codificada com o IYUV, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 256 <i>frames</i> (banda de 32 amostras).....	97
Figura 6.26 – Sequência 1 codificada com o IYUV, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 256 <i>frames</i> (banda de 32 amostras)	98
Figura 6.27 – Sequência 1 codificada com o IYUV, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 512 <i>frames</i> (banda de 32 amostras).....	99
Figura 6.28 – Sequência 1 codificada com o IYUV, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 512 <i>frames</i> (banda de 32 amostras)	99
Figura 6.29 – Sequência 1 codificada com o MV1, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 128 <i>frames</i> (banda de 32 amostras).....	101
Figura 6.30 – Sequência 1 codificada com o MV1, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 128 <i>frames</i> (banda de 32 amostras)	102
Figura 6.31 – Sequência 1 codificada com o MV1, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 256 <i>frames</i> (banda de 32 amostras).....	103
Figura 6.32 – Sequência 1 codificada com o MV1, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 256 <i>frames</i> (banda de 32 amostras)	103
Figura 6.33 – Sequência 1 codificada com o MV1, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 512 <i>frames</i> (banda de 32 amostras).....	104
Figura 6.34 – Sequência 1 codificada com o MV1, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 512 <i>frames</i> (banda de 32 amostras)	104
Figura 6.35 – Sequência 1 codificada com o RLE, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 128 <i>frames</i> (banda de 32 amostras).....	106

Figura 6.36 – Sequência 1 codificada com o RLE, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 128 <i>frames</i> (banda de 32 amostras).....	107
Figura 6.37 – Sequência 1 codificada com o RLE, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 256 <i>frames</i> (banda de 32 amostras)	108
Figura 6.38 – Sequência 1 codificada com o RLE, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 256 <i>frames</i> (banda de 32 amostras).....	108
Figura 6.39 – Sequência 1 codificada com o RLE, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 512 <i>frames</i> (banda de 32 amostras)	109
Figura 6.40 – Sequência 1 codificada com o RLE, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 512 <i>frames</i> (banda de 32 amostras).....	109
Figura 7.1 – Três tipos de sinais possíveis	113
Figura 7.2 – Sequência dividida em zonas	114
Figura 7.3 – Metodologia proposta para a amostragem dos sinais.....	115
Figura A.1 – Alguns <i>frames</i> da sequência 2 observada, banda de 32 amostras no FPB	128
Figura A.2 – Sequência 2, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 128 <i>frames</i> (banda de 32 amostras).....	130
Figura A.3 – Sequência 2, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 128 <i>frames</i> (banda de 32 amostras)	130
Figura A.4 – Alguns <i>frames</i> da sequência 2 reconstruída a partir da sequência de 128 <i>frames</i> , banda de 32 amostras no FPB	131
Figura A.5 – Sequência 2, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 256 <i>frames</i> (banda de 32 amostras).....	132
Figura A.6 – Sequência 2, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 256 <i>frames</i> (banda de 32 amostras)	132
Figura A.7 – Alguns <i>frames</i> da sequência 2 reconstruída a partir da sequência de 256 <i>frames</i> , banda de 32 amostras no FPB	133
Figura A.8 – Sequência 2, RMSE, resultados por <i>pixel</i> a 2D obtidos na reconstrução dos sinais da sequência de 512 <i>frames</i> (banda de 32 amostras).....	134
Figura A.9 – Sequência 2, PSNR, resultados por <i>pixel</i> a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 512 <i>frames</i> (banda de 32 amostras)	134
Figura A.10 – Alguns <i>frames</i> da sequência 2 reconstruída a partir da sequência de 256 <i>frames</i> , banda de 32 amostras no FPB	136

ÍNDICE DE TABELAS

Tabela 2.1 – Formatos de <i>frame</i> de vídeo	12
Tabela 2.2 – Parâmetros ITU-R BT.601-5	13
Tabela 3.1 – Apresentação sumária das diferenças entre o MPEG-4 Visual e o H.264.....	37
Tabela 3.2 – Possíveis significados do segundo byte.....	40
Tabela 3.3 – Exemplo de um <i>bitmap</i> RLE de 8 bits.....	41
Tabela 3.4 – Período de amostragem das componentes YUV	43
Tabela 5.1 – Características das sequências utilizadas.....	58
Tabela 5.2 – Verificação da condição da condição de convergência do algoritmo de Papoulis-Gerchberg	62
Tabela 5.3 – Tabela comparativa dos ficheiros AVI criados utilizando ou não métodos de codificação	70
Tabela 6.1 – Sequência 1 – Resultados obtidos na reconstrução dos sinais de 128, 256 e 512 <i>frames</i> sem compressão (banda de 32 amostras).....	73
Tabela 6.2 – Sequência 1 – Resultados obtidos na reconstrução dos sinais de 256 e 512 <i>frames</i> sem compressão (banda de 64 amostras).....	83
Tabela 6.3 – Sequência 1 – Resultados obtidos na reconstrução dos sinais de 512 <i>frames</i> sem compressão (banda de 128 amostras).....	86
Tabela 6.4 – Tabela comparativa dos ficheiros AVI criados utilizando ou não métodos de codificação	89
Tabela 6.5 – Sequência 1 codificada com o RCC, Resultados da reconstrução dos sinais de 128, 256 e 512 <i>frames</i> (banda de 32 amostras)	90
Tabela 6.6 – Sequência 1 codificada com o IYUV, Resultados da reconstrução dos sinais de 128, 256 e 512 <i>frames</i> (banda de 32 amostras)	95
Tabela 6.7 – Sequência 1 codificada com o MV1, Resultados da reconstrução dos sinais de 128, 256 e 512 <i>frames</i> (banda de 32 amostras)	100
Tabela 6.8 – Sequência 1 codificada com o RLE, Resultados da reconstrução dos sinais de 128, 256 e 512 <i>frames</i> (banda de 32 amostras)	105
Tabela A.1 – Sequência 2, Resultados da reconstrução dos sinais de 128, 256 e 512 <i>frames</i> (banda de 32 amostras)	129

CAPÍTULO 1 - INTRODUÇÃO

1.1 ENQUADRAMENTO

Ultimamente, as taxas de transmissão em rede (por exemplo: Internet ou Sistemas móveis) têm aumentado, tornando possível ao utilizador comum o acesso a diversos tipos de informação num espaço de tempo reduzido. Simultaneamente o aumento da capacidade de armazenamento em *hard disks*, memórias *flash*, dispositivos ópticos, entre outros, torna possível o armazenamento de grandes volumes de informação.

Com o decréscimo da relação custo/bit transmitido e/ou armazenado, torna-se aparentemente menos significativo o porquê da compressão de imagem, som e vídeo, e as razões pelas quais se tentam melhorar as técnicas já existentes [Richardson, 2003]. No entanto, a compressão de vídeo permite dois importantes benefícios. Em primeiro lugar torna possível a transmissão e armazenamento de vídeo digital em ambientes que não suportam vídeo sem compressão (vídeo bruto): por exemplo, as taxas de débito da Internet são insuficientes para lidar com vídeo sem compressão em tempo real (mesmo a baixas taxas e tamanhos de *frame*). Um *Digital Versatile Disk* (DVD) pode armazenar apenas alguns segundos de vídeo “bruto” com resolução e taxas de *frame* idênticas à qualidade de televisão, pelo que se conclui que o armazenamento de vídeo em DVD com essa qualidade não seria possível sem compressão. Em segundo lugar, a compressão de vídeo permite uma maior eficiência na utilização dos recursos de transmissão e armazenamento. Se tivermos ao nosso dispor um canal de transmissão com elevado bit *rate*, então torna-se mais eficiente enviar um único vídeo de alta resolução comprimido em múltiplos canais de vídeo comprimidos, do que enviar um único *stream* não comprimido de baixa resolução. Mesmo com os constantes avanços na capacidade de armazenamento e transmissão, a compressão é um componente essencial em serviços multimédia [Richardson, 2003].

A compressão de dados é conseguida através de dois métodos: a redução da *redundância estatística* e a redução de *redundância perceptual* (eliminação de informação pouco relevante). É possível classificar as técnicas de compressão de dados das seguintes formas: *compressão sem perdas*, *compressão com perdas* e *compressão quase sem perdas*. Um sistema de compressão sem perdas, baseia-se unicamente na redução de redundância estatística. Neste tipo de sistema as taxas de compressão são reduzidas. Um sistema de

compressão com perdas baseia-se na redução de redundância estatística e na eliminação de informação perceptivelmente pouco relevante. Neste tipo de sistemas as taxas de compressão são elevadas. Finalmente, sistemas de compressão quase sem perdas baseiam-se na redução de redundância estatística e na eliminação controlada de informação, e possuem taxas de compressão médias [Salomon, 2000]. As técnicas de compressão de vídeo mais utilizadas são baseadas em compressão com perdas, que permitem taxas mais elevadas de compressão com a desvantagem do sinal decodificado não ser igual ao sinal original [Richardson, 2003].

Em geral, o principal objectivo dos algoritmos de compressão com perdas é reduzir o volume dos dados de forma eficiente, mas minimizando simultaneamente a distorção introduzida pelo processo de compressão. Os algoritmos de compressão de vídeo baseiam-se na remoção da redundância nos domínios temporal, espacial e/ou frequência. Quando uma sequência, em que não existem movimentos rápidos, é adquirida através de uma câmara de 25 *frames* por segundo, existem poucas diferenças entre duas *frames* consecutivas no pequeno intervalo de 0.04 (1/25) segundos. Existe claramente uma redundância temporal significativa, isto é, a maior parte da imagem permanece inalterada entre duas *frames* consecutivas.

Removendo os diferentes tipos de redundância (espaço, frequência e/ou tempo) é possível comprimir significativamente os dados, perdendo, no entanto, alguma informação e provocando distorção. É possível alcançar maior compressão codificando os dados processados, utilizando, para tal, um esquema de codificação de entropia, tais como, a codificação de Huffman ou a codificação Aritmética [Data Compression, 2005; Hankersson *et al.*, 1998].

A compressão de imagem e vídeo tem sido uma área de pesquisa e desenvolvimento bastante activa nas últimas duas décadas. Foram muitos os algoritmos/sistemas propostos e desenvolvidos para efectuar codificação de vídeo. De maneira a promover o trabalho de cooperação, competição e aumentar a oferta, foi necessário definir métodos *standard* para a compressão, permitindo assim que produtos de fabricantes diferentes sejam compatíveis. Isto levou ao desenvolvimento de vários *Standards* Internacionais para compressão de imagem e vídeo, tais como as séries dos *standards* JPEG, MPEG e H.26x [Richardson, 2003].

1.2 MOTIVAÇÃO

São várias as áreas onde se utilizam imagens com alta resolução espacial e temporal (Biomecânica, *crash tests*, estudo da vida animal, imagens médicas, entre outras). Hoje em dia, com a tecnologia disponível, é possível adquirir imagens de alta resolução espacial e a

elevada taxa de aquisição de *frame* que pode ir desde os 100 *frames* por segundo (fps) aos 1000 fps, que em alguns casos podem ir até aos 100.000 fps utilizando resolução espacial mais baixa [Redlake, 2005; Photosonics, 2005].

No caso de uma sequência de vídeo ter sido adquirida a 1000 fps, as 1000 *frames* que se obtêm em cada segundo terão de ser guardadas com determinado método de compressão ocupando um espaço considerável em disco. Pensou-se então numa forma de diminuir o espaço de armazenamento em disco, utilizando algoritmos de reconstrução de sinal, armazenando apenas a quantidade de amostras suficiente, de modo a garantir a reconstrução dos sinais com uma relação sinal/ruído adequada à aplicação final. Ou seja, guardar a sequência com um menor número de *frames* do que a taxa de *frame* utilizada na aquisição, podendo a partir da sequência com menor número de *frames* obter a original. Existem duas maneiras de reduzir a taxa de *frame*. A primeira, e mais simples, é por subamostragem regular, omitir por exemplo todos os *frames* com índice ímpar. A segunda consiste em omitir um conjunto de M *frames* em cada N ($M < N$) de forma irregular. Este último processo permitiria em principio uma maior adaptabilidade às características do sinal de vídeo, mas neste trabalho considera-se apenas a subamostragem regular. Faz-se contudo notar que a transição para o caso mais geral fica facilitada devido ao tipo de algoritmos de reconstrução usados, que já permitem abordar o caso mais geral.

1.3 OBJECTIVOS

Com o trabalho apresentado nesta dissertação pretende-se conjugar técnicas de compressão de sinal, mais concretamente, imagens, com técnicas de reconstrução de sinal. O objectivo é avaliar uma alternativa aos métodos de compressão tradicionais baseados em previsão de movimento, através da inclusão de técnicas de reconstrução de sinal, para o caso específico das sequências de alta resolução temporal.

Para tal analisa-se o sinal representativo de cada um dos *pixels* de uma sequência de *frames* de alta resolução temporal (225 *frames* por segundo), desprezando um determinado número de amostras de cada um dos sinais, com taxa de amostragem inicial igual a n . Ao desprezarmos um determinado número de amostras, o resultado será uma sequência com menor número de *frames*, e por conseguinte um ficheiro com tamanho inferior ao da sequência original ($< n$).

Para tentar obter novamente a sequência, ou sinais originais, é aplicado o algoritmo de reconstrução de sinal aos sinais subamostrados; neste caso, o algoritmo de

Papoulis-Gerchberg [Papoulis, 1975] [Ferreira_A, 1994], cujo objectivo consiste em recuperar as amostras perdidas/desconhecidas, obtendo assim os sinais originais, permitindo, deste modo, a reconstrução da sequência original.

1.4 ORGANIZAÇÃO DA DISSERTAÇÃO

A presente dissertação encontra-se estruturada em sete capítulos.

No Capítulo 1, é feita uma introdução ao tema abordado nesta dissertação, indicando qual a motivação e objectivos do trabalho desenvolvido.

No Capítulo 2, é feita uma pequena abordagem aos formatos e qualidade de vídeo, e no Capítulo 3, são apresentadas as principais características de alguns dos *standards* de codificação de imagem e de codificação de vídeo actualmente mais utilizados.

No Capítulo 4, é explicado o algoritmo de reconstrução escolhido, qual o seu funcionamento e quais os seus requisitos e critérios de convergência.

No Capítulo 5, é apresentada a metodologia aplicada para conjugar ambas as técnicas de codificação e reconstrução.

No Capítulo 6, são apresentados os resultados obtidos após os testes realizados com a metodologia aplicada.

Finalmente, no Capítulo 7, apresentam-se as conclusões do trabalho realizado, apontando algumas direcções possíveis para futuros desenvolvimentos e extensões da metodologia aqui apresentada.

CAPÍTULO 2 - FORMATOS E QUALIDADE DE VÍDEO

Neste capítulo, são analisadas a estrutura e características de imagens digitais e sinais de vídeo. São, igualmente, introduzidos conceitos úteis para compreender a codificação de vídeos tais como, formatos de amostragem e qualidade métrica. O vídeo digital representa uma cena natural visual (mundo real), amostrada espacialmente e temporalmente. Uma cena é amostrada num determinado instante de tempo para produzir um *frame* ou quadro (a representação completa da cena visual nesse instante de tempo), ou um campo (que consiste num conjunto de linhas numeradas, pares ou ímpares, das amostras espaciais). A amostragem é repetida em intervalos regulares (i.e. 1/25 ou 1/30 intervalos de segundo, normalmente) para produzir um sinal de vídeo em movimento. Tipicamente, são necessários três conjuntos de amostras (componentes) para representar uma cena com cor.

Os formatos geralmente usados para representar vídeo na forma digital incluem o *standard* ITU-R BT.601¹ e um conjunto de “formatos intermediários”. A fidelidade da reprodução de uma cena visual tem que ser medida para determinar o desempenho de um sistema visual de comunicação, um processo notoriamente difícil e impreciso. Medições subjectivas consomem tempo e estão sujeitas a variações nos espectadores humanos. Medições objectivas (automáticas) são mais fáceis de implementar, mas ainda não se comparam com exactidão à opinião de um humano [Richardson, 2003].

2.1 CENAS DE VÍDEO NATURAIS

Uma cena de vídeo *real* ou *natural* é composta por múltiplos objectos, cada qual com as suas características de forma, profundidade, textura e iluminação. A cor e o brilho de uma cena de vídeo natural mudam com os diversos graus de suavidade (regularidade) através da cena (*tom contínuo*). As características de uma cena de vídeo natural que são relevantes para o processamento e compressão de vídeo incluem características espaciais (variação de textura dentro da cena, número e forma dos objectos, cor, entre outros) e características temporais (movimento de objectos, mudanças na iluminação, movimentos da câmara ou ponto de vista, entre outros).

¹ ITU-R BT.601 *Standard* para componente de vídeo digital

2.2 AQUISIÇÃO

Uma cena visual natural é espacialmente e temporalmente contínua. Representar na forma digital uma cena visual, envolve amostrar a cena real espacialmente (usualmente numa grelha rectangular no plano da imagem de vídeo) e temporalmente (como uma série de *frames* fixos ou componentes de *frames* amostrados em intervalos de tempo rectangulares). O vídeo digital é a representação de uma cena de vídeo amostrada de forma digital. Cada amostra espaço-temporal (elemento da imagem ou *pixel*) é representada com um número ou conjunto de números que descreve o brilho (luminância) e a cor da amostra.

Para obter imagens amostradas a duas dimensões, 2D, a câmara foca uma projecção da cena de vídeo a 2D para um sensor, como por exemplo um *Charge Coupled Device* (CCD) *array*. No caso da aquisição de imagens a cor, cada componente de cor é filtrada separadamente e projectada num *CCD array* [Richardson, 2003; Shapiro e Stockman, 2001].

2.2.1 AMOSTRAGEM ESPACIAL

A saída de um *CCD array* é um sinal de vídeo analógico, um sinal eléctrico variável que representa uma imagem de vídeo. Amostrar o sinal num determinado instante de tempo, produz uma imagem ou *frame* que define valores de um conjunto de pontos amostrados. O formato mais comum para uma imagem amostrada é um rectângulo com os pontos amostrados posicionados numa grelha quadrada ou rectangular. Na Figura 2.1 é possível observar um *frame* de tom contínuo com duas grelhas de amostragem diferentes sobrepostas sobre ele.

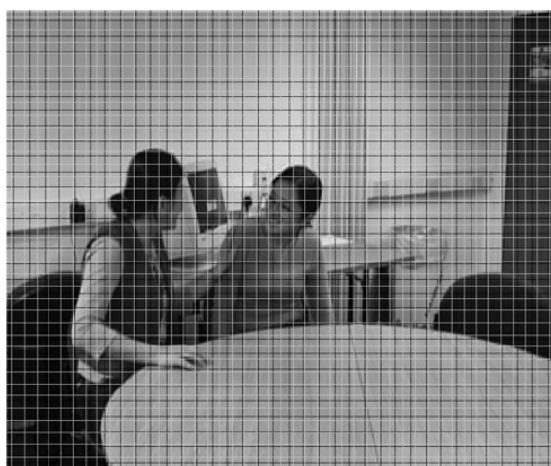


Figura 2.1 – Imagem com 2 grelhas de amostragem²

²Imagem retirada de [Richardson 2003]

A amostragem ocorre em cada um dos pontos de intersecção da grelha e a imagem amostrada pode ser reconstruída, representando cada amostra como um elemento quadrado da imagem (*pixel*). A qualidade visual da imagem é influenciada pelo número de pontos de amostragem. Optando por uma grelha de amostragem grosseira (rude), (grelha preta da Figura 2.1), obtém-se uma imagem amostrada a baixa resolução (Figura 2.2) enquanto que, aumentando ligeiramente o número de pontos de amostragem (grelha branca da Figura 2.1) aumenta-se a resolução da imagem amostrada (Figura 2.3) [Richardson, 2003].



Figura 2.2 – Imagem amostrada com baixa resolução (grelha de amostragem preta)³



Figura 2.3 – Imagem amostrada com resolução ligeiramente superior (grelha de amostragem branca)⁴

2.2.2 AMOSTRAGEM TEMPORAL

Captar uma imagem de vídeo em movimento é equivalente a obter uma fotografia do sinal durante intervalos de tempo periódicos. Efectuando o *playback* da série de *frames*

³ Imagem retirada de [Richardson 2003]

⁴ Imagem retirada de [Richardson 2003]

produz a ideia de movimento. Uma taxa de amostragem temporal mais elevada (*frame rate*) dá uma aparente suavidade ao movimento na cena de vídeo mas requer um maior número de amostras adquiridas e armazenadas. *Frame rates* inferiores a 10 *frames* por segundo, são utilizadas em alguns casos, para comunicações de vídeo de *bit rate* muito inferior (porque a quantidade de dados é relativamente baixa), no entanto, o movimento a esta taxa é claramente irregular e pouco natural. Entre 10 e 20 *frames* por segundo é um *bit rate* mais típico para comunicações vídeo de taxa baixa, a imagem é mais suave mas o movimento pode parecer irregular nas zonas de movimento rápido da sequência. Amostrar a 25 ou 30 *frames* por segundo é o *standard* para imagens de televisão (com entrelaçamento para melhorar o aspecto do movimento); 50 ou 60 *frames* por segundo produzem uma aparência suave de movimento (o custo é uma taxa de dados muito elevada) [Richardson, 2003].

2.2.3 FRAMES E CAMPOS

Um sinal de vídeo pode ser amostrado como uma série de *frames* completos (*progressive sampling* - amostragem progressiva) ou como uma sequência de campos entrelaçados (*interlaced sampling* - amostragem entrelaçada). Numa sequência de vídeo entrelaçada, metade dos dados dum *frame* (um campo) é amostrado a cada intervalo de tempo de amostragem. Um campo consiste nas linhas de número ímpar, ou par, existentes num *frame* de vídeo completo. Uma sequência de vídeo entrelaçada (Figura 2.4) contém uma série de campos, cada um representa metade da informação contida num *frame* de vídeo completo (Figura 2.5).

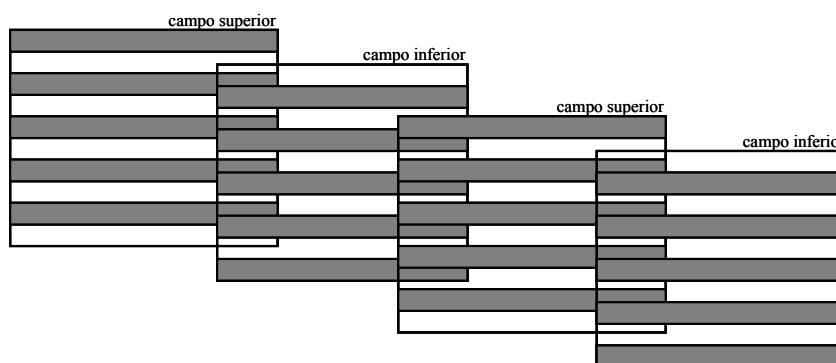


Figura 2.4 – Sequência de vídeo entrelaçada⁵

A vantagem deste método de amostragem é que com a mesma taxa de dados, se podem enviar duas vezes mais campos por segundo, em relação ao número de *frames* numa

⁵ Adaptado de [Richardson 2003]

sequência progressiva equivalente, dando a aparência de movimento suave. Por exemplo, uma sequência de vídeo PAL (*Phase Alternation by Line*) é constituída por 50 campos por segundo, e quando se efectua o *play*, o movimento parece mais suave do que numa sequência progressiva equivalente de 25 *frames* por segundo.



Figura 2.5 – (a) Campo superior e (b) campo inferior⁶

2.3 ESPAÇOS DE COR

A principal organização internacional dedicada à luz e cor é a *Commission International de l'Éclairage* (CIE). Esta organização é responsável por desenvolver *standards* e definições nesta área.

A CIE define cor como sendo o resultado perceptível da luz na região visível do espectro, possuindo comprimentos de onda de 400 nm até 700 nm, incidentes na retina. A grandeza física (ou radiação) é expressa numa distribuição espectral de potência, (*Spectral Power Distribution - SPD*), na maioria das vezes com 31 componentes, que representam cada um, uma banda de 10 nm [Salomon, 2000].

A CIE define brilho como o atributo de uma sensação visual segundo a qual uma área emite mais ou menos luz. É impossível definir a percepção cerebral do brilho, o que levou a CIE a definir uma quantidade denominada luminância, que normalmente é representada por *Y*. Esta é definida como sendo a potência da radiação medida através de uma função espectral de sensibilidade que é característica da visão. A luminância é uma quantidade importante nas áreas de processamento digital de imagem e da compressão [Salomon, 2000].

⁶Imagens retiradas de [Richardson, 2003]

2.3.1 RGB

No espaço de cor RGB (*Red, Green, Blue*), uma imagem a cores é representada através de três quantidades que indicam as proporções relativas de vermelho, verde e azul. Qualquer cor pode ser obtida através da combinação destes três componentes. Na Figura 2.6 é possível observar as componentes vermelha, verde e azul de uma imagem.

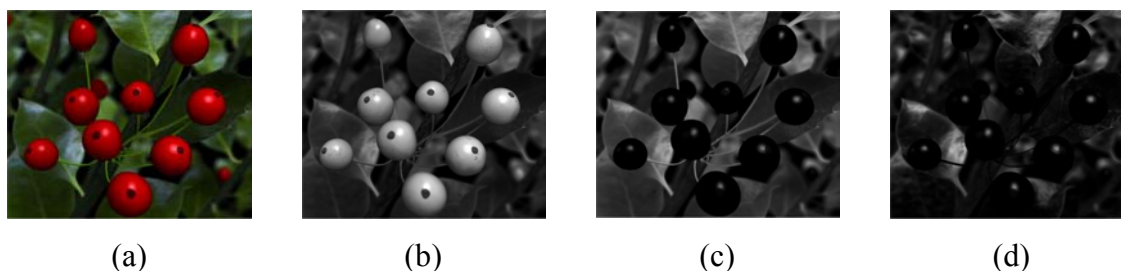


Figura 2.6 – (a) Imagem original e suas componentes (b) vermelha, (c) verde e (d) azul

Por exemplo, tendo por base a Figura 2.6 (a) é possível verificar que os objectos vermelhos aparecem mais brilhantes na componente vermelha (Figura 2.6 (b)), o mesmo se passa com os objectos verdes, que aparecem mais claros na componente verde (Figura 2.6 (c)). O espaço de cor RGB é bom para capturar e exibir imagens a cor. Fazer a aquisição de imagens a cor envolve filtrar cada uma das componentes: vermelha, verde e azul da cena, e capturar separadamente cada uma dessas componentes com sensores distintos [Richardson, 2003; Rao e Hwang, 1996].

2.3.2 YCbCr

O olho humano é mais sensível a pequenas mudanças na luminância, do que a mudanças de cor, razão pela qual, é útil ter espaços de cor que utilizam Y como um dos seus três parâmetros. No espaço RGB, as três cores possuem igual importância e são usualmente armazenadas com a mesma resolução. É, no entanto, possível representar mais eficientemente uma imagem a cores separando a luminância da informação da cor, representando desta forma a luminância com maior resolução, do que a cor. Uma forma simples de o conseguir é subtraindo Y das componentes azul e vermelho, e utilizar três componentes Y , $B-Y$, e $R-Y$, como novo espaço de cor. As duas últimas componentes são denominadas crominância. Estas representam, para uma dada intensidade luminosa, a cor em termos de presença ou ausência de azul (Cb) e vermelho (Cr) [Rao e Hwang, 1996].

Para diferentes aplicações existem várias gamas de valores utilizadas em $B-Y$ e $R-Y$. As gamas YPbPr são optimizadas para componentes de vídeo analógico. Por sua vez, as gamas

YCbCr são mais adequadas para componentes de vídeo digital, tais como: estúdio de vídeo, JPEG, JPEG2000 e MPEG.

O espaço de cor YCbCr foi desenvolvido como parte da recomendação ITU-R BT.601 durante o desenvolvimento de um componente de vídeo mundial *standard*. *Y* pode tomar valores que vão desde 16 até 235; *Cb* e *Cr* podem tomar valores na gama 16 a 240, com 128 igual a zero. Existem diversas amostras de formatos YCbCr, tais como 4:4:4, 4:2:2, 4:1:1, e 4:2:0, que estão igualmente descritos na recomendação anteriormente mencionada [Salomon, 2000].

Converter uma imagem no espaço RGB com valores entre 16 e 235 para o espaço YCbCr é uma operação linear simples. Essa transformação é obtida através das seguintes equações, (2.1).

$$\begin{aligned} Y &= (77/256)R + (150/256)G + (29/256)B \\ Cb &= -(44/256)R - (87/256)G + (131/256)B + 128 \\ Cr &= (131/256)R - (110/256)G - (21/256)B + 128 \end{aligned} \quad (2.1)$$

A transformação oposta, ou seja a passagem do espaço YCbCr para RGB pode ser efectuada através das equações (2.2).

$$\begin{aligned} R &= Y + 1.371(Cr - 128) \\ G &= Y - 0.698(Cr - 128) - 0.336(Cb - 128) \\ B &= Y + 1.732(Cb - 128) \end{aligned} \quad (2.2)$$

Quando se efectua a transformação de YCbCr para RGB, os valores RGB resultantes possuem um alcance nominal de 16-235, com valores ocasionais possíveis entre 0-15 e 236-25 [Salomon, 2000].

2.4 FORMATOS DE VÍDEO

Os *standards* de compressão de vídeo podem comprimir uma larga variedade de formatos de *frame* de vídeo. Na prática, é comum capturar ou converter para um formato intermédio antes da compressão e transmissão. O formato intermédio comum *Common Intermediate Format* (CIF) é a base para um conjunto de formatos conhecidos apresentados na Tabela 2.1. A Figura 2.7 mostra a componente de luminância de um *frame* de vídeo amostrado numa gama de resoluções decrescentes, desde 4CIF até à Sub-QCIF. A escolha da

resolução do *frame* depende da aplicação e da capacidade de armazenamento ou de transmissão. Por exemplo, 4CIF é adequado para as definições *standard* de televisão e DVD-vídeo; o CIF e o *Quarter Common Intermediate Format* (QCIF) são populares em aplicações de videoconferência; QCIF ou SQCIF são adequados para aplicações multimídia móveis nas quais a resolução de *display* bem como o *bit rate* são limitados [Richardson, 2003].

Tabela 2.1 – Formatos de *frame* de vídeo⁷

Formato	Resolução da Luminância (horizontal × vertical)	Bits por <i>Frame</i> (4:2:0, oito bits por amostra)
Sub-QCIF	128×96	147456
Quarter CIF (QCIF)	176×144	304128
CIF	352×288	1216512
4CIF	704×576	4866048

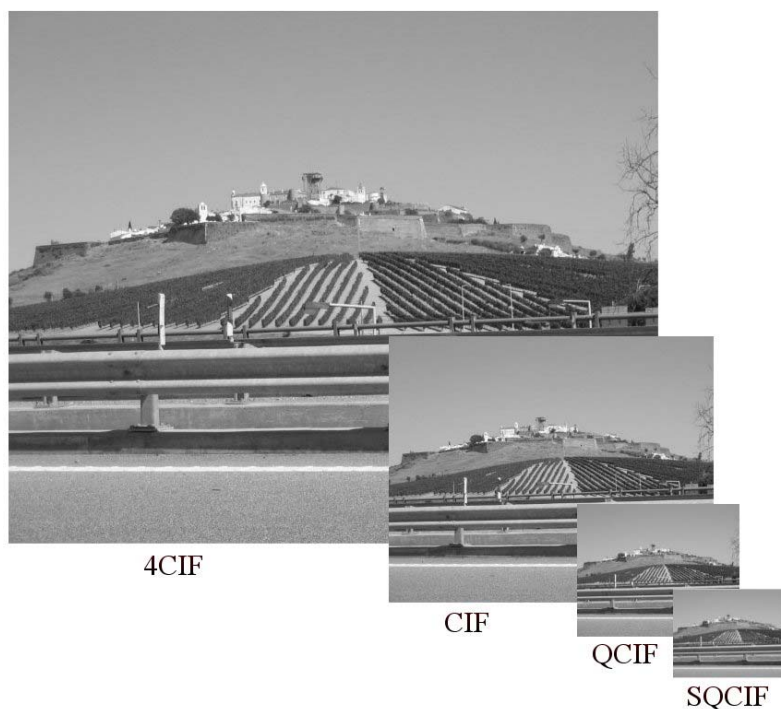


Figura 2.7 – *Frame* de vídeo amostrado com diversas gamas de resolução

Na Tabela 2.1 está apresentado o número de bits necessários para representar um *frame* não-comprimido em cada um dos formatos (assumindo uma amostragem 4:2:0 e 8 bits por cada amostra de luminância e crominância).

⁷ Retirado de [Richardson, 2003]

A recomendação ITU-R BT.601-5⁸ é um formato bastante utilizado na codificação digital de sinais de vídeo para produção de televisão (o termo codificação na recomendação, significa conversão para o formato digital e não implica compressão). A componente de luminância do sinal de vídeo é amostrada a 13.5 MHz e a crominância a 6.75 MHz, para produzir um sinal 4:2:2 de componentes Y:Cb:Cr. Os parâmetros do sinal digital de vídeo amostrado dependem da taxa de *frame* do vídeo (30 Hz para uma sinal NTSC (*National Television System(s) Committee*) e 25 Hz para um sinal PAL/SECAM (*Sequentiel Couleur avec Mémoire*)) e estão apresentados na Tabela 2.2. A taxa de *frame* mais elevada no NTSC é compensada por uma resolução espacial mais baixa, logo o bit *rate* total em cada um dos casos é o mesmo (216 Mbps). A área actual mostrada no *display*, área activa, é mais pequena que a área total, esta exclui os intervalos de *blanking* que existem “fora” dos limites do *frame*.

Cada amostra possui uma gama de 0-255. Os níveis 0 e 255 estão reservados para sincronização e o sinal activo de luminância é restringido a uma gama de 16 (preto) e 235 (branco).

Tabela 2.2 – Parâmetros ITU-R BT.601-5⁹

	Taxa de <i>frame</i> 30 Hz	Taxa de <i>frame</i> 25 Hz
Campos por segundo	60	50
Linhas por <i>frame</i> completo	525	625
Amostras de luminância por linha	858	864
Amostras de crominância por linha	429	432
Bits por amostra	8	8
Bit <i>rate</i> total	216 Mbps	216 Mbps
Linhas activas por <i>frame</i>	480	576
Amostras activas por linha (Y)	720	720
Amostras activas por linha (Cr, Cb)	360	360

2.5 QUALIDADE

A qualidade de imagem e vídeo é um factor de grande importância quando se fala de codificação de imagem e vídeo. Se for necessário, por exemplo, avaliar dois métodos de compressão diferentes, a nossa avaliação pode ser baseada na medida da qualidade da imagem

⁸ Recomendação ITU-R BT.601-5, Parâmetros de codificação em estúdio da televisão digital para os *standards* 4:3 e ecrã panorâmico 16:9, ITU-T, 1995.

⁹ Retirado de [Richardson, 2003]

ou do vídeo. Quando ambos os métodos atingem a mesma qualidade nas imagens ou vídeo reconstruídas, aquele que requer menor volume de dados é considerado superior. Sendo assim, de entre dois métodos que requerem a mesma quantidade de dados, é considerado melhor método, aquele que fornecer maior qualidade nas imagens ou vídeo reconstruídos. É importante referir que não se considera como critério de avaliação o desempenho, como por exemplo, a complexidade computacional [Shi e Sun, 2000].

Medir a qualidade visual de uma imagem é um processo difícil e impreciso, isto porque, são muitos os factores que podem afectar o resultado desta medição. Obter uma medida de qualidade precisa é difícil, uma vez que a qualidade visual é influenciada por um elevado número de factores e é subjectiva por inerência. A opinião de um espectador em relação à qualidade visual, depende muito da tarefa que está a executar, que pode ser a visualização de um filme em DVD, participar numa sessão de videoconferência, tentar identificar uma pessoa numa cena obtida a partir de um vídeo de vigilância, entre outros [Richardson, 2003].

Existem dois tipos de avaliação da qualidade visual. Um é a avaliação objectiva (recorrendo a medidas eléctricas), e a outra é a avaliação subjectiva (recorrendo a observadores humanos). Cada tipo possui a sua métrica e as suas desvantagens. Hoje em dia, a combinação dos dois métodos é bastante utilizada [Shi e Sun, 2000].

Medir a qualidade visual através de critérios objectivos, obtêm-se resultados precisos que se podem repetir. No entanto, não existem ainda sistemas de medição objectivos capazes de reproduzir fidedignamente a experiência subjectiva de um humano a visualizar um vídeo [Richardson, 2003].

2.5.1 MEDIDA SUBJECTIVA

A percepção visual humana de um *frame* de uma sequência de vídeo, resulta de uma interacção complexa entre o olho e o cérebro. A percepção da qualidade visual é influenciada pela fidelidade espacial e pela fidelidade temporal. No entanto, existem outros factores que influenciam a opinião do espectador em relação à qualidade da cena visualizada, são eles o ambiente de visualização, o estado de espírito do espectador e o nível de interacção do espectador com a cena visual. Quando um espectador está a realizar uma tarefa que requer maior concentração na cena visual, este requer melhor qualidade do que um utilizador que vê calmamente um filme. Em muitos casos, a opinião visual de um espectador é elevada se, o ambiente de visualização é confortável e não possui distrações, isto em detrimento da própria qualidade de imagem.

Outras influências importantes para distinguir a qualidade são a atenção com a qual o espectador visualiza a cena e o “efeito recente” (a nossa opinião de uma cena visual depende em grande parte das últimas cenas visualizadas). Estes factores tornam difícil medir qualidade visual de forma fidedigna, bem como, quantificar essa medida.

Num sistema de medida de qualidade subjectivo, é gerado um conjunto de *frames* de vídeo variando alguns parâmetros de codificação. Os observadores são posteriormente chamados a avaliar a qualidade visual dos *frames*. Para tal, é-lhes pedido que avaliem quantitativamente a qualidade das imagens. É igualmente pedido aos observadores que, de forma alternada, forneçam uma medida da perda de qualidade das imagens. Seguidamente, apresenta-se um sistema de cinco escalas de grau de perda, utilizado pelos laboratórios Bell [Shi e Sun 2000]. A escala possui os seguintes pontos:

1. Não se nota perda de qualidade.
2. A perda de qualidade é ligeiramente notada.
3. A perda de qualidade é definitivamente notada, mas não objectável.
4. A perda de qualidade é objectável.
5. A perda de qualidade é extremamente objectável.

Relativamente à avaliação subjectiva, convém ainda mencionar algumas coisas. Na maioria das aplicações existem um conjunto de vectores de imagens, disponíveis simultaneamente para avaliação. Estas imagens são criadas com diferentes parâmetros de codificação. A alguns desses parâmetros são atribuídos valores fixos, enquanto que um (ou subconjunto de parâmetros) possui um valor variável, a qualidade resultante desta codificação, pode ser utilizada para avaliar o efeito de um determinado parâmetro (ou subconjunto de parâmetros) [Shi e Sun, 2000].

Outra possibilidade para estudar o efeito, consiste em identificar, no vector de imagens total, as imagens com a mesma qualidade subjectiva. Deste subconjunto de imagens teste podem criar-se, no espaço de codificação de parâmetros, curvas de preferência que podem ser utilizadas para estudar o efeito do parâmetro, ou parâmetros, sob investigação. Nesta quantificação, um vector completo de imagens é dividido em colunas, e cada coluna partilha algumas condições comuns. O processo de avaliação começa dentro de cada coluna com uma comparação par a par, isto porque este tipo de comparação é relativamente fácil para os nossos olhos. Deste modo, as imagens de uma coluna são ordenadas de acordo com a qualidade visual, e, posteriormente, são atribuídas às imagens dessa coluna quantificações de qualidade ou medidas de deficiência (diminuição de qualidade). Depois de cada uma dessas colunas ter sido quantificada, é necessária uma unificação entre as colunas. Ou seja, colunas

diferentes devem possuir uma medida de qualidade única. Esta tarefa não é fácil, uma vez que pode ser necessário igualar perdas que resultem de diferentes tipos de erros [Shi e Sun, 2000].

Uma conclusão que se pode obter é que a avaliação subjectiva de qualidade possui custos elevados. Esta requer um elevado número de imagens e de observadores. Este tipo de avaliação demora algum tempo, uma vez que os nossos olhos cansam-se ou aborrecem-se facilmente. Torna-se, pois, necessário, tomar em consideração algumas medidas para chegar a uma quantificação precisa de qualidade subjectiva. Alguns procedimentos de teste para avaliação subjectiva de qualidade estão definidos na recomendação ITU-R BT.500-11¹⁰. Um procedimento *standard* para efectuar medida subjectiva de qualidade é o método *Double Stimulus Continuous Quality Scale* (DSCQS) [Richardson, 2003; Boch *et al.*, 1997].

Para mais informação sobre medição subjectiva de qualidade consultar [Webster *et al.*, 1993; Wade e Swanson, 2001].

2.5.2 MEDIDA OBJECTIVA

Medir automaticamente a qualidade usando um algoritmo, tornou-se atractivo, devido à complexidade e custo dos sistemas de medição subjectivos. As entidades ligadas ao desenvolvimento de sistemas de compressão de vídeo e sistemas de processamento de vídeo utilizam muitas das vezes os denominados medidores de qualidade objectivos. O exemplo mais utilizado na medição objectiva de qualidade é o *Peak Signal-to-Noise Ratio* (PSNR) [Richardson, 2003; Lu *et al.*, 2002; Shi e Sun, 2000]. O PSNR possui, no entanto, algumas limitações, que fizeram com que os esforços para desenvolver medidas mais eficazes, que se aproximam da resposta de um observador real, não tenham cessado [Richardson, 2003].

O PSNR, equação (2.3), é medido numa escala logarítmica, o seu valor depende directamente da razão entre $(2^n - 1)^2$ (que representa o quadrado do maior valor de sinal possível na imagem, n é o número de bits por *pixel*), e o *Mean Square Error* (MSE). O MSE, equação (2.4), é calculado entre a imagem original e a imagem degradada (ou *frame* de vídeo).

$$PSNR_{(dB)} = 10 \log_{10} \frac{(2^n - 1)^2}{MSE} \quad (2.3)$$

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N [f(i, j) - \hat{f}(i, j)]^2 \quad (2.4)$$

¹⁰ Recomendação ITU-T BT.500-11, Metodologia para avaliação subjectiva da qualidade de imagens de televisão, ITU-T, 2002

O PSNR poder ser calculado facilmente e de forma rápida, razão pela qual é uma medida de qualidade muito usada. Este valor é utilizado para comparar a qualidade de imagens de vídeo codificadas e decodificadas.

Para uma dada imagem ou sequência de imagens, um PSNR elevado indica usualmente alta qualidade e um PSNR baixo indica usualmente qualidade baixa. Uma das limitações do PSNR prende-se com o facto de este necessitar da imagem original para efectuar a comparação, que muitas vezes não está disponível. Uma outra limitação deve-se ao facto de que o PSNR não se correlacionar bem com as medidas de qualidade de vídeo subjectivas tais como, as definidas na ITU-R BT.500.

Devido às limitações do PSNR, têm sido efectuados esforços para desenvolver um sistema objectivo de teste que mais se aproxime dos resultados obtidos em testes subjectivos. No entanto, ainda não existe nenhum sistema de medição de qualidade objectivo, preciso e *standard*, adequado ao vídeo digital codificado. Reconhecendo esta realidade, a *International Telecommunication Union – Telecommunications Standardization Sector* (ITU-T) *Video Quality Experts Group* (VQEG) [VQEG, 2005] pretende desenvolver *standards* para avaliação objectiva da qualidade de vídeo.

Pode-se afirmar que a medição objectiva da qualidade de imagens nem sempre fornece resultados fidedignos. No entanto, a sua implementação é muito mais rápida e fácil em comparação com a medição subjectiva da qualidade. Uma outra vantagem prende-se com o facto desta poder ser repetida. Devido a esta realidade, a avaliação objectiva da qualidade é ainda muito utilizada. Combinar as avaliações: subjectiva e objectiva tem sido prática comum nas actividades dos *standards* de codificação internacionais [Shi e Sun, 2000].

CAPÍTULO 3 - COMPRESSÃO DE IMAGEM E VÍDEO

O *Joint Photographic Experts Group* (JPEG) e o *Moving Pictures Experts Group* (MPEG) são grupos de trabalho da *International Standards Organisation* (ISO) responsáveis, respectivamente, pelo desenvolvimento de alguns *standards* de codificação de imagens fixas e codificação de vídeo. De entre eles referimos os *standards* de codificação de imagens fixas, JPEG [JPEG, 1992] e JPEG 2000 [JPEG2000, 2000], e os *standards* de codificação de vídeo, MPEG-1 [MPEG1, 1993], MPEG-2 [MPEG2, 1995] e MPEG-4 Visual [MPEG4 – 2, 2001].

O *Video Coding Experts Group* (VCEG) é um grupo de trabalho do ITU-T. O ITU-T desenvolve *standards* (ou recomendações) para a área das telecomunicações. O VCEG tem sido responsável pelo desenvolvimento de uma série de *standards* relacionados com comunicação de vídeo sobre redes de telecomunicações e de computadores. O *standard* de videoconferência H.261 [H.261, 1993] foi seguido pelo mais eficiente H.263 [H.263, 1998], e este foi posteriormente superado pelas suas posteriores versões (denominadas H.263+ e H.263++) que vieram aumentar as capacidades do H.263. O anteriormente denominado “H.26L”, levou ao desenvolvimento e publicação da recomendação H.264.

Desde 2001, este esforço levou à cooperação entre o VCEG e o MPEG, e o novo *standard*, denominado *Advanced Video Coding* (AVC), foi juntamente publicado como ITU-T H.264 e ISO/IEC MPEG-4 Part 10 [H.264, 2003].

Neste capítulo serão abordados quais os principais objectivos e características dos *standards* mencionados anteriormente. Apresenta-se também uma descrição sucinta dos codificadores utilizados na metodologia descrita no capítulo 5.

3.1 JPEG E JPEG2000

O grupo JPEG é responsável pelo desenvolvimento de alguns dos *standards* de codificação de imagens fixas, sendo os mais relevantes os *standards* JPEG e o JPEG2000. Estes *standards* possuem algumas características comuns com o MPEG-4 Visual e/ou o H.264, e embora o seu objectivo seja a compressão de imagens fixas, os *standards* JPEG tiveram bastante impacto na codificação de imagens em movimento [Richardson, 2003].

O *standard* JPEG original suporta a compressão de imagens fotográficas utilizando a *Discrete Cosine Transform* (DCT) 8×8 seguida de quantificação, reordenação, codificação

run-length e codificação de entropia *variable-length*. Quando o MPEG-4 Visual é utilizado no modo *Intra-coding*, os *standards* possuem muitas semelhanças.

O *standard* JPEG2000 foi desenvolvido com o objectivo de ser um sucessor mais eficiente do JPEG original. Este utiliza a *Discrete Wavelet Transform* (DWT), esta é o seu método básico de codificação, razão pela qual, possui semelhanças com as ferramentas de codificação de textura fixa do MPEG-4 Visual. O JPEG2000 fornece desempenho de compressão superior ao do JPEG e não apresenta as limitações da compressão baseada na DCT.

Apesar do facto de se considerar o JPEG como sendo uma tecnologia antiga e com desempenho inferior ao JPEG2000 e a outros formatos de compressão proprietários, este é ainda muito utilizado no armazenamento de imagens nas câmaras digitais, nos computadores pessoais e nas páginas *web*.

O *Motion JPEG* (um método não-*standard* de comprimir uma sequência de *frames* de vídeo utilizando o JPEG) é utilizado em aplicações tais como a aquisição de vídeo, edição de vídeo em PC e vigilância.

3.1.1 JPEG

O JPEG é um método sofisticado de compressão *lossy/lossless* para imagens fixas, a cores ou em níveis de cinzento (NDC). Este método não apresenta um bom desempenho com imagens a dois níveis (a preto e branco). O método é mais eficiente com imagens de tons contínuos onde os *pixels* adjacentes possuem cores similares. Uma vantagem do JPEG é que este utiliza muitos parâmetros, permitindo que o utilizador ajuste a quantidade de dados perdidos (e assim a taxa de compressão). Muitas vezes, o olho humano não consegue ver nenhuma degradação na imagem, mesmo para taxas de compressão na ordem de 10:1 e 20:1. Existem no JPEG dois modos principais: o *lossy* (também denominado *baseline*) e o *lossless* (que produz tipicamente taxas de compressão de aproximadamente 0.5). A maioria das implementações suporta apenas o modo *lossy*. Este modo inclui codificação hierárquica progressiva.

O JPEG é um método de compressão e não um *standard* para representação de imagem. Esta é a razão pela qual o JPEG não especifica características da imagem tais como, taxa de aspecto do *pixel* (*pixel aspect ratio*), espaço cor (*color space*) ou entrelaçamento nas linhas do *bitmap* (BMP).

Os principais objectivos da compressão JPEG são os seguintes:

1. Taxas de compressão elevadas, especialmente nos casos em que a qualidade de imagem é classificada como muito boa ou excelente.
2. O uso de muitos parâmetros, permitindo que utilizadores sofisticados experimentem e alcancem o compromisso compressão/qualidade desejado.
3. Obter bons resultados com qualquer tipo de imagens de tom contínuo, independentemente das dimensões da imagem, espaços de cor, taxas de aspecto de *pixel*, ou outras características da imagem.
4. É um método de compressão sofisticado mas não demasiadamente complexo, permite a implementação *software* e *hardware* em diversas plataformas.
5. Possui diversos modos de operação: (a) Modo Sequencial: cada componente (cor) da imagem é comprimido num único varrimento da esquerda para a direita e de cima para baixo; (b) Modo Progressivo: a imagem é comprimida em múltiplos blocos (conhecidos como “varrimentos”) para serem vistos com detalhe grosseiro até chegar a perfeito; (c) Modo *Lossless*: é importante para os casos em que o utilizador decide que não deverá ocorrer perda de *pixels* (obtendo um menor taxa de compressão comparativamente com os modos *lossy*); e (d) Modo Hierárquico: a imagem é comprimida com múltiplas resoluções, permitindo que blocos com menor resolução sejam vistos primeiro, sem que para isso seja necessário descomprimir os blocos seguintes com maior resolução.

O JPEG é o resultado de um esforço conjunto entre o CCITT (*Comité Consultatif International Télégraphique et Téléphonique*) e a ISO (*International Standards Organization*) que começou em Junho de 1987 e resultou no primeiro esboço de proposta em 1991. O *standard* JPEG provou o seu sucesso e é extensamente utilizado em compressão de imagem, especialmente em páginas *Web* [Salomon, 2000].

3.1.1.1 DCT – DISCRETE COSINE TRANSFORM

A DCT é a transformada mais utilizada na compressão de imagem, sendo adoptada por um grande número de normas de compressão. É muito similar à transformada de Fourier, excepto pelo facto de não utilizar números complexos. A transformada em duas dimensões para um sinal de dimensão $N \times N$ é definida por:

$$y_{u,v} = \frac{C(v)}{2} \frac{C(u)}{2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x_{i,j} \cos\left[\frac{(2i+1)u\pi}{2N}\right] \cos\left[\frac{(2j+1)v\pi}{2N}\right] \quad (\text{Eq.3.1})$$

onde $y_{u,v}$ representa os coeficientes da DCT para a frequência vertical u e frequência horizontal v . O valor amostrado na posição i,j da imagem é representado por $x_{i,j}$. As constantes $C(u)$ e $C(v)$ tomam o valor $1/\sqrt{2}$ para $u=0$ e $v=0$ respectivamente, caso contrário tomam o valor um [Castleman, 1996].

Em implementações comuns, a imagem de entrada é dividida em blocos de dimensão 8×8 . Uma vez que a transformada é separável, o seu cálculo é simplificado. A transformada unidimensional é aplicada oito vezes ao longo das linhas e oito vezes ao longo das colunas da imagem. A norma para compressão suave de imagens de tons contínuos foi desenvolvida pelo JPEG. O algoritmo base divide a imagem em blocos 8×8 . Seguidamente cada bloco é transformado para o domínio da DCT utilizando a equação (3.1). Os coeficientes DCT são quantificados utilizando uma tabela de quantificação, que é constante ao longo da imagem e enviada ao decodificador. O coeficiente $y_{0,0}$, também denominado de coeficiente DC, é processado separadamente dos outros coeficientes, a sua diferença com o termo DC do bloco anterior é codificada utilizando o código de Huffman. Os restantes coeficientes (também denominados coeficientes AC) são depois enviados utilizando um *run-length* e uma codificação Huffman. A descodificação do *bitstream* é feita através a aplicação dos mesmos passos, mas em ordem inversa.

O decodificador JPEG tem que ser capaz de descodificar os *bitstreams* codificados com o algoritmo base. No entanto, muitos módulos suplementares estão disponíveis no *standard*. Para melhorar o desempenho, especialmente a baixos bit *rates*, um codificador aritmético pode ser utilizado em detrimento do codificador de Huffman. É igualmente possível ter *bitstreams* de qualidade progressiva e resolução progressiva.

Como é o caso da maioria das normas de compressão, apenas o decodificador é especificado. Este quantifica a zero alguns coeficientes para aumentar o desempenho taxa/distorção do codificador. Outra perspectiva foi tomada no algoritmo DCTune, onde o codificador tenta otimizar a qualidade visual das imagens comprimidas [Gall, 1991].

O comité JPEG utilizou a DCT, porque esta possui um bom desempenho, e não assume nada em relação à estrutura dos dados, por exemplo a *Discrete Fourier Transform* (DFT) assume que os dados a serem transformados são periódicos. A DCT foi igualmente escolhida porque existem formas de a tornar mais rápida [Salomon, 2000].

3.1.1.2 MODO *LOSSLESS*

O modo *lossless* no JPEG utiliza diferenciação para reduzir os valores dos *pixels* antes que estes sejam comprimidos. Esta forma particular de diferenciação é denominada *predicting*. Os valores de alguns dos vizinhos do *pixel* são subtraídos ao *pixel* de forma a obter um valor mais pequeno, o qual é mais tarde comprimido utilizando codificação Huffman ou Aritmética.

É de referir que o modo *lossless* do JPEG não obteve muito sucesso. Este modo produz um factor de compressão de dois, que é inferior a outros métodos de compressão de imagem *lossless*. Devido a esta característica, as implementações mais populares do JPEG não implementam este modo. O modo *lossy* (*baseline*) do JPEG não tem bom desempenho quando lhe é pedido que limite a quantidade de perdas ao valor mínimo. Como consequência, algumas implementações do JPEG não permitem o ajuste de parâmetros para obter o mínimo de perdas. A eficácia do JPEG reside na sua capacidade de gerar imagens bastante comprimidas, que após serem descomprimidas são indistinguíveis das originais. Reconhecendo este facto a ISO decidiu criar um novo *standard* para a compressão *lossless* de imagens de tons contínuos. Esse *standard* é vulgarmente conhecido como JPEG-LS e será descrito na secção seguinte.

3.1.1.3 JPEG – LS

O JPEG-LS (*JPEG – Lossless*) é o *standard* da ISO/ITU-T para compressão *lossless* de imagens. Este fornece também compressão “*near-lossless*” (quase sem perdas). É baseado em predição adaptativa, modelação de contexto e codificação Golomb. Como complemento, este *standard* possui um detector de regiões planas, para depois as codificar em *run-length*. A compressão *near-lossless* é atingida permitindo na amostragem um erro máximo, fixo. Este algoritmo foi projectado com uma complexidade mínima fornecendo no entanto taxas de compressão elevadas. Contudo este *standard* não fornece suporte para escalabilidade, robustez ao erro ou qualquer funcionalidade semelhante [Cruz e Ebrahimi, 2000].

3.1.1.4 JBIG

Não existe um único método de compressão capaz de comprimir todo o tipo de dados, razão pela qual se desenvolvem novos métodos para fins específicos. O *Joint Bi-Level Image Processing Group* (JBIG) foi especificamente desenvolvido para efectuar compressão progressiva de imagens a dois níveis. Este tipo de imagens, também denominadas monocromáticas ou a preto e branco, são comuns em aplicações onde desenhos (técnicos ou

artísticos), com ou sem texto, precisam de ser guardados em bases de dados e recuperados. É comum utilizar termos com primeiro plano e fundo em vez de preto e branco, respectivamente.

A expressão “compressão progressiva” significa que a imagem é guardada em diversos níveis no *stream* comprimido, com resoluções cada vez mais elevadas. Quando este tipo de imagem é descomprimida e visualizada, o observador primeiro vê uma imagem imprecisa e rude (correspondente ao primeiro nível) seguida de versões melhoradas da mesma (níveis seguintes). Desta forma se a imagem não for a pretendida, esta pode ser rejeitada numa fase inicial, sem que esta seja totalmente recuperada e descomprimida.

Apesar de o JBIG ter sido desenvolvido para imagens a dois níveis, nas quais cada *pixel* é um bit, este pode ser aplicado em imagens a níveis de cinzento separando os *bitplanes* e comprimindo-os individualmente, como se fossem imagens a dois níveis. Neste caso deverá ser usado o *Reflected Gray Code* (RGC), em vez do código binário *standard* [Salomon, 2000].

Uma importante característica da definição do JBIG é que a operação do decodificador não está definida em detalhe. O *standard* JBIG debate os detalhes do decodificador e do formato do ficheiro comprimido. Está implícito que qualquer codificador que gera um ficheiro JBIG é um decodificador JBIG válido.

Uma vantagem do método JBIG é a sua habilidade para gerar versões (ou diversas níveis) da imagem de baixa resolução no *stream* comprimido. O decodificador descomprime progressivamente estes níveis, desde a resolução mais baixa até à mais alta.

A imagem total (nível de maior resolução) é entregue ao codificador, este constrói os níveis desde a resolução alta para a mais baixa e escreve no ficheiro comprimido os níveis (imagens) pela mesma ordem. Por seu lado, o decodificador, começa por descomprimir a camada de resolução mais baixa, daí que seja mais simples para ele ler este nível em primeiro lugar. Como resultado, o codificador ou o decodificador deverão utilizar um armazenador para reverter a ordem dos níveis. É importante a sua rápida descodificação, o codificador deverá utilizar armazenadores para acumular todos os níveis, e depois guardá-los no ficheiro comprimido desde a resolução mais baixa até à mais alta. Seguidamente o decodificador lê os níveis na ordem correcta. Nos casos em que é importante a codificação rápida (tal como quando um arquivo é actualizado frequentemente mas raramente é descomprimido e utilizado), o codificador deve escrever os níveis pela ordem em que são gerados (alta para baixa resolução) e o decodificador deve utilizar armazenadores. O *standard* JBIG suporta qualquer um dos métodos, o codificador coloca no ficheiro comprimido um bit denominado

HITOLO a zero (quando os níveis estão em ordem crescente de resolução) ou um (na ordem oposta). É o codificador que escolhe a resolução do nível mais baixo. Esta decisão pode ser baseada em informação fornecida pelo utilizador ou em informação embutida no codificador sobre necessidades específicas de um ambiente particular [Salomon, 2000].

A compressão progressiva no JBIG envolve igualmente o conceito de *stripes*. Uma *stripe* é uma banda horizontal estreita que consiste em L linhas de varrimento da imagem, onde L é um parâmetro JBIG controlado pelo utilizador.

3.1.1.5 JBIG2

O *Joint Bi-Level Image Processing Group* desenvolveu um novo *standard* de compressão de imagens a dois níveis, o JBIG2. Este *standard* oferece [Salomon, 2000; JGIB2, 2005]:

- Um aumento significativo no desempenho de compressão (tipicamente 3 a 5 vezes menor que Group 4/ MMR, e 2 a 4 vezes menor que o JBIG1 (JBIG));
- Métodos de compressão especiais para texto, *halftones*, e outras partes de imagem a dois níveis;
- Compressão com e sem perdas;
- Dois modos de compressão progressiva. Modo 1, é compressão progressiva de qualidade, onde a imagem descodificado progride de baixa para alta qualidade. O Modo 2, é compressão progressiva de conteúdo, onde partes importantes da imagem (tais como texto) são descodificadas em primeiro lugar, seguidas pela descodificação das partes menos importantes (tais como padrões *halftone*);
- Compressão multi-página de documentos;
- Formato flexível, desenhado para uma fácil inclusão noutros formatos de ficheiro, como o *Tag(ged) Image File Format* (TIFF);
- Descompressão de alto desempenho: utilizando alguns modos de codificação, as imagens podem ser descomprimidas a uma taxa de 250 milhões de *pixels* por segundo.

O *standard* JBIG2 descreve os princípios da compressão e o formato do ficheiro comprimido. Qualquer codificador que produza um ficheiro comprimido JBIG2 é um codificador JBIG2 válido. O descodificador JBIG2 lê o ficheiro comprimido, o qual contém, dicionários e informação da imagem, e descomprime-o para um *page buffer*. Essa imagem é depois apresentada ou imprimida do *page buffer*. Vários *buffers* auxiliares podem ser utilizados pelo descodificador.

Um documento a ser comprimido pelo JBIG2 pode consistir em mais do que uma página. A principal característica do JBIG2 é que este distingue de uma página, texto, imagens *halftone*, e tudo o resto existente na página. O codificador JBIG2 deve em primeiro lugar varrer a página antes de efectuar qualquer tipo de codificação, e identificar regiões dos três seguintes tipos:

- Regiões de texto – estas contêm texto, normalmente arranjado em linhas;
- Regiões *halftone* – uma imagem a dois níveis (preto/branco) pode conter imagem a níveis de cinzento, feita com *halftone*;
- Regiões genéricas – qualquer região não identificada pelo codificador como não sendo texto ou *halftone*.

O JBIG2 introduziu o conceito de aperfeiçoamento de região. O ficheiro comprimido pode conter instruções indicando ao descodificador que descodifique a região A do ficheiro comprimido para um *buffer* auxiliar. Esse *buffer* pode ser mais tarde utilizado para refinar a descodificação de uma outra região B. Quando B é encontrada no ficheiro comprimido e descodificada para o *page buffer*, cada *pixel* escrito no *page buffer* é determinado pelos *pixels* descodificados do ficheiro comprimido e pelos *pixels* no *buffer* auxiliar. Um codificador JBIG2 simples pode nunca utilizar refinação de região, mas um codificador sofisticado pode utilizar este conceito para adicionar efeitos especiais em algumas regiões da imagem.

3.1.2 JPEG2000

Com a contínua expansão da multimédia e das aplicações na Internet, as necessidades e requisitos das tecnologias utilizadas cresceram e evoluíram. Em Março de 1997 foi lançado o desafio para o desenvolvimento de um novo *standard* para compressão de imagens estáticas, o JPEG2000. Este projecto, JTC 1.29.14 (15444), tinha por objectivo criar um novo sistema de codificação de imagem para diferentes tipos de imagens estáticas (monocromáticas, a níveis de cinzento, a cores, ou com múltiplos componentes), com diferentes características (imagens naturais, científicas, médicas, detecção remota, texto, tradução de gráficos, entre outras), permitindo assim diferentes modelos de manuseamento de imagem (cliente/servidor, transmissão em tempo real, arquivo de livrarias de imagem, barramento limitado e recursos de largura de banda, entre outros) preferencialmente num sistema unificado. Este sistema de codificação deveria fornecer operações a baixo bit *rate* com taxa de distorção e desempenho de qualidade de imagem subjectiva superior à dos *standards* existentes, isto sem sacrificar o desempenho nos outros pontos do mesmo espectro taxa de distorção, e incorporando simultaneamente muitas características contemporâneas [Skodras *et al.*, 2000].

3.1.2.1 APLICAÇÕES, REQUERIMENTOS E CARACTERÍSTICAS

O *standard* JPEG2000 fornece um conjunto de características que são de vital importância para aplicações emergentes e de ponta, tirando partido de novas tecnologias. Este abrange áreas onde os *standards* actuais não conseguiram produzir a melhor qualidade ou desempenho e fornece capacidades para entrar em mercados que actualmente não utilizam compressão. Os mercados e aplicações que mais beneficiam do *standard* JPEG2000 são a Internet, fac-símile, impressão, *scanning* (consumidor e pré-impressão), fotografia digital, detecção remota (*remote sensing*), dispositivos móveis, imagens médicas, livrarias digitais/arquivos e comércio electrónico (*E-commerce*) [Christopoulos *et al.*, 2000]. Cada área de aplicação impõe alguns requerimentos que o *standard* deve cumprir. As características que o *standard* deve possuir são as seguintes:

- **Desempenho superior a baixo bit rate:** Este *standard* deve oferecer desempenho superior ao dos *standards* actuais a baixo bit rate (i.e. abaixo dos 0.25 bpp para imagens a níveis de cinzento altamente detalhadas). Isto aumentou significativamente o desempenho a baixo bit rate que é alcançado sem sacrificar o desempenho no espectro taxa de distorção restante. Exemplos de aplicações que necessitam desta característica são a transmissão de imagem em rede e detecção remota. Esta é a característica de maior prioridade.
- **Compressão a tom contínuo e a dois níveis:** É desejável ter um *standard* de codificação capaz de comprimir imagens de tom contínuo e a dois níveis. Se realizável, o *standard* deverá procurar atingir este objectivo com recursos similares. O sistema deverá comprimir e descomprimir imagens com várias escalas (gamas) dinâmicas por cada componente de cor (i.e. 1 bit a 16 bit). Exemplos de aplicações que podem utilizar esta característica incluem documentos compostos por imagens e texto, imagens médicas com camadas de anotações, gráficos e imagens geradas por computador com regiões binárias e perto de binárias, alfa e planos de transparência, e fac-símile.
- **Compressão *lossless* (sem perdas) e *lossy* (com perdas):** É desejável fornecer naturalmente a compressão *lossless* no curso da descodificação progressiva. Exemplos de aplicações que podem utilizar esta característica são imagens médicas, onde as perdas não são sempre toleráveis, aplicações de arquivo de imagens, onde a melhor qualidade é vital para a preservação mas desnecessária para exposição, aplicações em

rede que fornecem dispositivos com diferentes capacidades e recursos, e imagens de pré-impressão.

- **Transmissão progressiva por exactidão de *pixel* e resolução:** Transmissão progressiva que permite a reconstrução de imagens com crescente perfeição exactidão de *pixel* ou resolução espacial é essencial para muitas aplicações. Esta característica permite a reconstrução de imagens com diferentes resoluções e exactidão de *pixel*, necessitada ou desejada, para diferentes dispositivos alvo. Exemplos de aplicações incluem a *World Wide Web*, aplicações de arquivo de imagens e impressoras.
- **Acesso e processamento aleatório no fluxo de código:** Frequentemente existem partes da imagem que são mais importantes do que outras. Esta característica permite ao utilizador definir regiões de interesse (ROI – *Regions Of-Interest*) na imagem a ser aleatoriamente acedida e/ou descomprimida com menor distorção do que no resto da imagem. O processamento aleatório do *codestream* poderá permitir operações tais como rotação, translação, filtragem, extracção de características e escalonamento.
- **Robustez a erros:** É desejável ter em atenção a ocorrência de erros aquando da criação do *codestream*. Uma aplicação onde este assunto é de extrema importância é a comunicação por canais *wireless* (sem fios). Para determinar a qualidade da imagem descodificada, existem porções do *codestream* que podem ser mais importantes do que outras. O projecto de códigos de detecção e correcção de erros pode ajudar os sistemas a evitar falhas de descodificação.
- **Arquitectura aberta:** É importante possuir uma arquitectura aberta para permitir a optimização do sistema para diferentes tipos de imagens e aplicações. Com esta característica, é pedido ao descodificador que implemente apenas o conjunto de ferramentas nucleares e um analisador que interprete o *codestream*. Se necessárias, as ferramentas desconhecidas são pedidas pelo descodificador e enviadas da fonte.
- **Capacidade *build-up* sequencial (codificação em tempo real):** O *standard* deverá ser capaz de comprimir e descomprimir imagens apenas com um único passo sequencial. Este deverá igualmente ser capaz de processar uma imagem utilizando componentes *interleave order* e *non-interleave order*. Durante a compressão e descompressão, o *standard* deve utilizar contexto limitado e um número de linhas razoável.

3.1.2.2 ARQUITECTURA DO STANDARD

O diagrama de blocos do codificador JPEG2000 está apresentado na Figura 3.1. É idêntico a qualquer esquema de codificação baseado em transformadas. Em primeiro lugar é aplicada à imagem de dados fonte a transformada discreta. Seguidamente são quantificados os coeficientes da transformada e codificados pela sua entropia, antes de formado o *bitstream*. O decodificador é o inverso do codificador (Figura 3.1b). Primeiro o *bitstream* é decodificado por entropia, seguidamente é efectuada a operação inversa da quantificação e posteriormente é aplicada a transformada discreta inversa, obtendo deste modo a imagem reconstruída. É importante referir que, ao contrário dos outros esquemas de codificação, o JPEG2000 pode ser *lossy* e *lossless*. Isto depende apenas da transformada de *wavelet* e da quantificação aplicada [Skodras *et al.*, 2000].

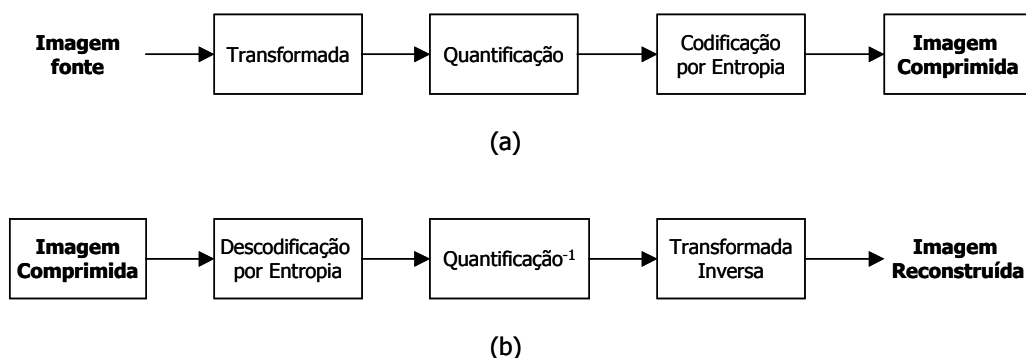


Figura 3.1 – Diagramas de blocos do codificador (a) e decodificador (b) JPEG2000.

O *standard* JPEG2000 utiliza o conceito de ladrilhos (*tiles*) de imagem. O termo *tiling* (ladrilhar) refere-se ao acto de dividir a imagem original (fonte) em blocos rectangulares (ladrilhos) que não se sobrepõem, que são comprimidos independentemente, como se representassem imagens distintas. Todas as operações, incluindo a mistura de componentes, transformada de *wavelet*, quantificação e codificação por entropia, são efectuadas independentemente nos ladrilhos da imagem. O ladrilhar da imagem reduz os requisitos de memória e uma vez que os ladrilhos são reconstruídos independentemente, estes podem ser utilizados para decodificar partes específicas da imagem em vez da imagem total. Todos os ladrilhos têm exactamente a mesma dimensão, excepto talvez os da camada direita mais baixa, no limite da imagem. É possível ter ladrilhos com tamanhos diferentes, até perfazer a totalidade da imagem (i.e. a imagem total é vista como um ladrilho) [Christopoulos *et al.*, 2000]. As dimensões nominais do ladrilho são potências de dois, exactas. Após aplicar a *Discrete Wavelet Transform* (DWT) a cada um dos ladrilhos, cada uma das amostras do

ladrilho componente da imagem são deslocadas a nível DC, isto através da subtracção da mesma quantidade (i.e. a profundidade do componente) de cada amostra (Figura 3.2) [Skodras *et al.*, 2000].

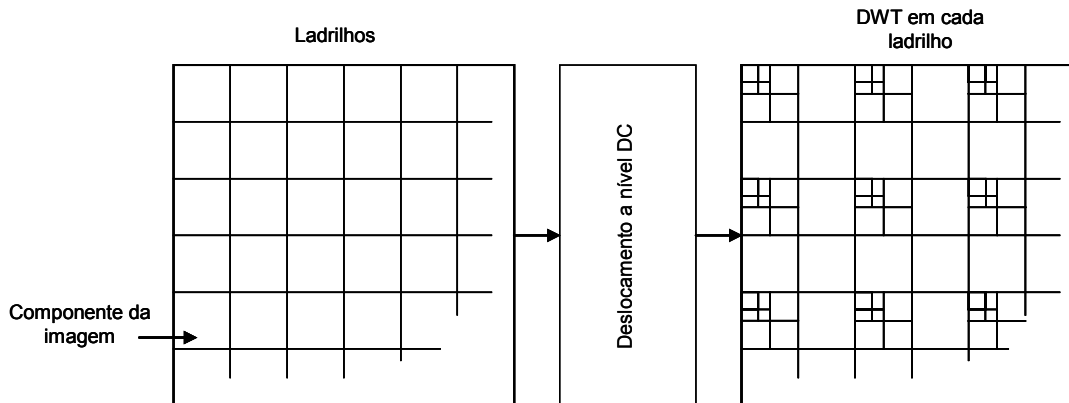


Figura 3.2 – Ladrilhos da imagem, Deslocamento a nível DC para cada ladrilho componente da imagem

É através da transformada *wavelet* que os ladrilhos componentes da imagem são decompostos em diferentes níveis de decomposição. Esses níveis de decomposição contêm um determinado número de sub-bandas, que consistem em coeficientes que descrevem as características da frequência espacial horizontal e vertical do ladrilho original. O nível de decomposição está relacionado com o próximo nível de decomposição através de potências de dois espaciais [Christopoulos *et al.*, 2000].

3.2 MPEG

O MPEG é o *Moving Picture Experts Group*, este trabalha sobre a direcção da ISO e da *International Electro-Technical Commission* (IEC). Este grupo desenvolve *standards* de codificação de imagens em movimento e áudio associado. Existem actualmente vários *standards* MPEG, o MPEG-1 foi projectado para taxas de dados intermédias, na ordem dos 1.5 Mbits/s. O MPEG-2 foi projectado para taxas de dados elevadas de pelo menos 10 Mbits/s. O MPEG-3 foi projectado para compressão de *High Definition Television* (HDTV), mas posteriormente foi dado como redundante e mais tarde incorporado no MPEG-2. O MPEG-4 foi projectado para taxas de dados muito baixas, inferiores a 64 Kbit/s. Uma terceira organização internacional, o ITU-T, tem estado envolvido do desenvolvimento do MPEG-2 e do MPEG-4 [Salomon, 2000].

O *standard* MPEG consiste em codificar completamente os *frames* chave de uma sequência, utilizando para tal um algoritmo semelhante ao JPEG, e estimar as mudanças (resultantes de movimento) entre esses *frames* chave. Uma vez que é enviada uma quantidade mínima de informação entre cada quatro ou cinco *frames*, o número de bits necessário para descrever essas imagens é significativamente reduzido. Assim, as taxas de compressão na ordem de 100:1 são muito comuns neste *standard*. O codificador MPEG é muito complexo e implica uma carga computacional muito elevada para fazer a estimativa de movimento. Já o decodificador MPEG é mais simples e pode ser efectuado pelo *Central Processing Unit* (CPU) de um computador pessoal ou através de *chips* decodificadores de baixo custo [Techonline, 2005].

O codificador MPEG pode efectuar uma predição sobre uma imagem e transformar e codificar a diferença entre a predição e a imagem. A predição tem em consideração o movimento dentro da imagem, utilizando, para tal, a estimativa de movimento. Uma vez que a predição de uma imagem pode ser baseada em imagens futuras bem como em imagens passadas, o codificador tem que reorganizar as imagens de modo a colocar imagens de referência antes das imagens preditas. O decodificador coloca novamente as imagens na ordem correcta de *display*. São necessárias na ordem de 1.1 a 1.5 mil milhões de operações por segundo para a codificação MPEG em tempo real [Techonline, 2005].

A entrada de um codificador MPEG é denominada *source data*, e a saída de um decodificador MPEG é denominada *reconstructed data*. A *source data* é organizada em pacotes (Figura 3.3), cada pacote começa com um código de início (32 bits) seguido de um cabeçalho, termina com um código de fim de 32 bits, entre os pacotes de início e fim este possui um determinado número de pacotes. Um pacote contém dados (vídeo ou áudio) comprimidos. O tamanho de um pacote é determinado pelo codificador MPEG de acordo com os requisitos dos meios de armazenamento e transmissão, razão pela qual um pacote não é necessariamente uma imagem de vídeo completa. Este pode ser qualquer parte de uma imagem vídeo ou qualquer parte de áudio [Salomon 2000].

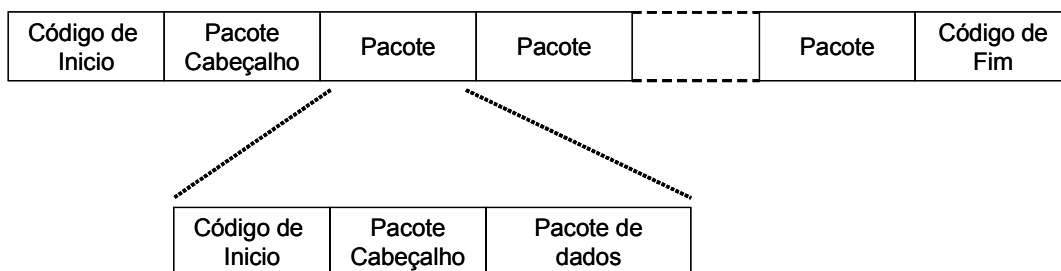


Figura 3.3 – Formato da *source*

O decodificador MPEG possui três partes principais, denominadas *layers* (camadas), para decodificar o áudio, o vídeo, e os dados do sistema. A *layer* do sistema lê e interpreta os diversos códigos e cabeçalhos da *source data*, e direcciona os pacotes para as *layers* de áudio ou de vídeo (Figura 3.4) para serem armazenados e mais tarde decodificados. Estas duas *layers* são constituídas por vários decodificadores a trabalharem em simultâneo [Salomon, 2000].

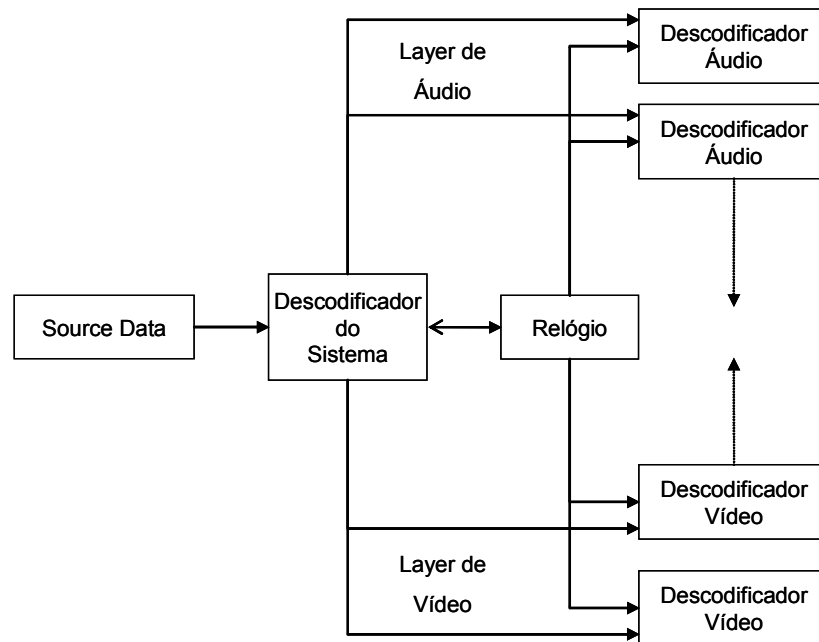


Figura 3.4 – Organização do decodificador MPEG

3.2.1 MPEG-1

O primeiro *standard* MPEG foi o MPEG-1; este foi desenvolvido para aplicações específicas de armazenamento e *playback* de vídeo em *Compact Disks* (CD). Consegue-se ler 70 minutos de um CD, a velocidade normal, com taxas de transferência de 1.4 Mbit/s. O MPEG-1 foi desenvolvido para suportar vídeo CD, um formato de armazenamento e *playback* de vídeo que fosse capaz de competir com as cassetes *Video Home System* (VHS). O MPEG-1 Vídeo utiliza compensação de movimento baseada em blocos, DCT e a quantificação (o codificador DPCM (*Differential Pulse-Code Modulation*)/DCT), e é optimizado para um *bit rate* de vídeo comprimido por volta dos 1.2 Mbit/s. Para além do MPEG-1 Vídeo, as partes áudio e *sistema* do *standard* foram desenvolvidas, respectivamente, para suportar compressão de áudio e a criação de *bitstreams* multiplexados [Richardson, 2003]. Com o *standard* MPEG-1 pode-se armazenar num único CD-ROM, 72 minutos de vídeo e áudio com qualidade VHS (640×480 com 30 fps) [Techonline, 2005]. O formato de vídeo CD não teve

êxito comercialmente talvez porque, a qualidade de vídeo não era suficientemente melhor que a de uma cassete VHS, e sendo assim não encorajou os consumidores a aderirem a esta nova tecnologia. No entanto, o MPEG-1 é ainda muito utilizado no armazenamento de ficheiros de vídeo comprimidos, em computadores pessoais e em rede.

3.2.2 MPEG-2

Depois do MPEG-1, surgiu o *standard* MPEG-2, cujo principal objectivo era abranger um mercado potencial, mais concretamente a radiodifusão digital de televisão comprimida. Este *standard* é baseado no MPEG-1 mas com diversas mudanças significativas para suportar a aplicação alvo, estas incluem o suporte para codificação eficiente de vídeo entrelaçado (bem como progressivo), uma sintaxe mais flexível, alguns melhoramentos na eficiência de codificação e uma parte do *sistema* significativamente mais flexível e poderosa [Richardson, 2003].

Resumindo, o *standard* MPEG-2 tem por alvo a transmissão de TV e outras aplicações com taxas de dados superiores ou iguais 4 Mbps. O MPEG-2 suporta formatos de vídeo entrelaçado, qualidade de imagem elevada, e outras características direccionadas para a HDTV. O MPEG-2 é uma extensão compatível com o MPEG-1, ou seja, um decodificador MPEG-2 também pode decodificar *streams* MPEG-1. O áudio no MPEG-2 fornece até cinco canais de largura de banda total (esquerda, direita, central, e dois canais *surround*), e ainda um canal adicional de realçamento de baixa frequência, ou até sete canais de comentário. O sistema do *standard* MPEG-2 especifica como combinar múltiplos *streams* de áudio, vídeo e dados privados, num único *stream* multiplexado e suporta uma larga gama de radiodifusão, telecomunicações, computação e aplicações de armazenamento [Techonline, 2005]. O *standard* MPEG-3 foi englobado no MPEG-2 e já não existe [Salomon, 2000; Techonline, 2005].

3.2.3 MPEG-4

O projecto do *standard* MPEG-4 começou em Julho de 1993, o seu objectivo era fornecer tecnologia base para permitir de forma eficiente o armazenamento baseado em conteúdo, transmissão, e manipulação de vídeo, gráficos, áudio, e outros dados no âmbito da área da multimédia.

A revisão que se apresenta seguir sobre o *standard* MPEG-4 nas referências [Shi e Sun 2000; Pereira e Ebrahimi, 2002].

O MPEG-4 possui características novas e atractivas que os *standards* anteriores não podiam fornecer. Algumas dessas características estão focadas no melhoramento da eficiência da codificação; outras são utilizadas para fornecer robustez na transmissão e interacção com o utilizador final. No entanto, a característica mais importante deste *standard* prende-se com o facto de este ser o primeiro a permitir a codificação baseada em conteúdos de objectos audiovisuais. Para os fornecedores e autores de conteúdos, o *standard* MPEG-4 pode proporcionar uma maior reutilização, flexibilidade, e manuseamento do conteúdo que é produzido. Para os fornecedores de rede, o MPEG-4 fornecerá informação transparente, a qual pode ser interpretada e traduzida nas mensagens de sinalização apropriadas para cada rede. Para os utilizadores finais, o MPEG-4 pode fornecer maior funcionalidade de modo a dar ao utilizador maior capacidade de interagir com o conteúdo. Para conseguir atingir estes objectivos o MPEG-4 possui as seguintes características:

- Os conteúdos tais como o áudio, vídeo ou dados são representados na forma de objectos audiovisuais primitivos (AVOs – *Audio Visual Objects*). Estes AVOs podem ser cenas naturais ou sons, que são gravados por uma câmara de vídeo ou gerados sinteticamente por computadores;
- Os AVOs podem ser compostos de forma a criar um composto de AVOs ou cenas;
- Os dados associados com os AVOs podem ser multiplexados e sincronizados de modo a que estes possam ser transportados, com determinados requisitos de qualidade, através dos canais da rede.

Uma vez que o alvo do *standard* MPEG-4 é maioritariamente aplicações multimédia, existem diversos requisitos para assegurar que as diversas características e funcionalidades são oferecidas. Essas características incluem a permissão de interacção, elevada compressão, acessibilidade universal, e portabilidade do conteúdo áudio e vídeo. Do documento dos requisitos de vídeo do MPEG-4, as principais funcionalidades podem ser resumidas em três aspectos: interacção baseada em conteúdo, compressão eficiente baseada em conteúdo, e acesso universal.

Para além de facultar a codificação eficiente de sequências de vídeo convencionais, o MPEG-4 vídeo possui as seguintes características de interacção baseada em conteúdo: Manipulação Baseada em Conteúdo e Edição de *Bitstream*, Codificação Híbrida Sintética ou Natural, e Acesso Temporal Aleatório Melhorado.

O MPEG-4 suporta a manipulação baseada em conteúdo e a codificação do *bitstream* sem necessitar de *transcoding* (conversão digital para digital de um esquema de codificação para outro). No MPEG-1 e no MPEG-2, não existe sintaxe nem semântica para suportar a

verdadeira manipulação e edição no domínio da compressão. O MPEG-4 faculta a sintaxe e as técnicas para suportar a manipulação baseada em conteúdo e edição do *bitstream*. O nível de acesso, edição, e manipulação pode ser efectuado ao nível do objecto em conexão com as características de escalabilidade baseada em conteúdo.

O MPEG-4 suporta a combinação de cenas ou objectos sintéticos com cenas ou objectos naturais. Esta característica serve para compor dados sintéticos com vídeo normal, permitindo interacção. As técnicas do MPEG-4 que permitem suportar esta característica incluem codificação *Sprite* (método para integrar imagens ou gráficos a 2D numa cena maior), codificação eficiente de superfícies a 2D ou 3D, e codificação *wavelet* para texturas fixas.

O MPEG-4 faculta um método eficiente para aceder de forma aleatória, dentro de um determinado limite de tempo, e com partes de resolução fina, isto é, *frames* de vídeo ou objectos de imagem arbitrariamente formados de uma sequência audiovisual. Isto inclui acesso aleatório convencional a bit *rate* muito baixo. Esta característica é igualmente importante para manipulação e edição baseada em conteúdo de *bitstreams*.

Um objectivo inicial do MPEG-4 era fornecer uma ferramenta de codificação altamente eficiente com elevada compressão a bit *rates* muito baixos. Este objectivo foi estendido para uma gama de bit *rates* desde os 10 Kbps até os 5 Mbps, que abrange os formatos de vídeo *Quarter Standard Image Format* (QSIF) e ITU-R BT.601.

O *standard* de vídeo MPEG-4 fornece subjectivamente melhor qualidade visual quando comparado, utilizando as mesmas bit *rates* que os outros *standards*, incluindo o MPEG-1/2 e o H.263. O MPEG-4 vídeo contém novas ferramentas, que optimizam o código nas diferentes gamas de bit *rate*. Alguns resultados experimentais demonstram que este possui um desempenho superior ao MPEG-2 e ao H.263 para baixos bit *rates*. Para além disso, a codificação baseada em conteúdo atinge desempenhos idênticos aos da codificação baseada em *frames*.

O MPEG-4 proporciona a capacidade de codificar eficientemente múltiplas vistas de uma cena. Para aplicações de vídeo estereoscópicas, o MPEG-4 faculta a possibilidade de explorar redundância na visualização múltipla de pontos da mesma cena, permitindo soluções de codificação conjuntas que possibilitam compatibilidade com o vídeo normal assim como vídeos que não possuem restrições de compatibilidade.

Uma outra importante característica do MPEG-4 é a característica do acesso universal. Este *standard* fornece capacidades de robustez ao erro muito fortes, isto para permitir o acesso de aplicações através de variadas redes, com fios ou sem fios, e o armazenamento de

dados multimédia. Para aplicações de baixo bit *rate*, sobre condições de erro severas, o *standard* fornece robustez ao erro suficiente.

O MPEG-4 fornece a capacidade de atingir escalabilidade com ténue granularidade em conteúdo, qualidade (resolução espacial e temporal), e complexidade. Esta escalabilidade destina-se a obter o escalonamento da informação visual baseado em conteúdo.

3.3 H.261 E H.263

O H.261 [H.261, 1993] foi o primeiro *standard* a ser amplamente utilizado em videoconferência, foi desenvolvido pelo ITU-T para suportar videotelefone e videoconferência sobre redes RDIS de comutação de circuitos. Este tipo de redes opera com múltiplos de 64 Kbit/s e o *standard* H.261 foi projectado para oferecer, a estes bit *rates*, codificação de vídeo utilizando computação simples. Este *standard* utiliza o conhecido modelo híbrido DPCM/DCT com compensação de movimento com precisão de inteiro [Richardson, 2003].

Numa tentativa de melhorar o desempenho de compressão do H.261, o grupo de trabalho do ITU-T desenvolveu o *standard* H.263 [H.263, 1998]. Este fornece compressão superior à do H.261, suportando, a bit *rates* inferiores a 30 Kbit/s, qualidades de vídeo básicas; o *standard* foi projectado para operar sobre uma ampla gama de redes de comutação de circuitos e de comutação de pacotes. O modelo base de codificação H.263 (modelo híbrido DPCM/DCT com compensação de movimento meio-*pixel* (*half-pixel*)) foi adoptado como núcleo do MPEG-4 Visual *Simple Profile* (Perfil Simples) [Richardson, 2003]. A versão original do H.263 inclui quatro modelos de codificação opcionais e posteriormente foram adicionados uma série de modos opcionais para suportar características tais como, melhoramento da eficiência de compressão e robustez de transmissão em redes com perdas [Richardson, 2003]. Os termos “H.263+” e “H.263++” são utilizados para descrever *codecs* (Codificador(es)/Decodificador(es)) que suportam algumas ou todas as opções e modos codificação do H.263. A escolha dos modos a suportar está a cargo do fabricante do *codec*, foi devido à larga gama de combinações possíveis, e à dificuldade de obter compatibilidade entre *codecs*, que o desenvolvimento do H.264 foi motivado [Richardson, 2003].

3.4 H.264/MPEG-4 PART 10

O *standard* H.264 possui um alcance mais limitado que o MPEG-4 Visual. Este foi essencialmente desenvolvido para suportar: a codificação robusta e eficiente, e o transporte de

frames de vídeo rectangulares. O seu objectivo original era fornecer funcionalidades similares às dos *standards* anteriores tais como, o H.263+ e o MPEG-4 Visual (perfil simples), mas com um desempenho de compressão significativamente superior e suporte para transmissões fidedignas melhorado. A aplicação alvo deste *standard* é a comunicação de vídeo bidireccional (videoconferência e videotelefone), codificação de emissão, vídeo de alta qualidade e *streamming* de vídeo sobre redes de pacotes. O *standard* suporta a transmissão robusta sobre redes nas quais está implementado. Este *standard* foi projectado de modo a ser facilmente implementado na maior gama de plataformas de processamento possível [Richardson, 2003].

O *standard* H.264 é especificado com maior detalhe do que *standards* posteriores (incluindo o MPEG-4 Visual), de forma a minimizar más interpretações por parte das pessoas ligadas ao desenvolvimento de *standards*.

3.5 COMPARAÇÃO ENTRE O MPEG-4 VISUAL E O H.264

Na Tabela 3.4 estão apresentadas algumas das principais diferenças entre os dois *standards*.

Tabela 3.1 – Apresentação sumária das diferenças entre o MPEG-4 Visual e o H.264¹¹

Comparação	MPEG-4 Visual	H.264
Tipos de dados suportados	<ul style="list-style-type: none"> • <i>Frames</i> e Campos de vídeo rectangulares; • objectos de vídeo de forma arbitrária; • texturas fixas e <i>sprites</i>; • objectos de vídeo híbridos (sintéticos ou sintético-naturais); • <i>mesh</i> de objectos 2D e 3D. 	<ul style="list-style-type: none"> • <i>Frames</i> e Campos de vídeo rectangulares.
Número de Perfis	<ul style="list-style-type: none"> • 19 	<ul style="list-style-type: none"> • 3
Eficiência de compressão	<ul style="list-style-type: none"> • Média 	<ul style="list-style-type: none"> • Elevada
Suporte de vídeo <i>streamming</i>	<ul style="list-style-type: none"> • Codificação escalável 	Comutação de <i>slices</i> <ul style="list-style-type: none"> • (<i>Switching slices</i>)

¹¹ Retirado de [Richardson, 2003]

Tabela 3.1 – Apresentação sumária das diferenças entre o MPEG-4 Visual e o H.264 (Continuação)

Comparação	MPEG-4 Visual	H.264
Compensação de movimento, tamanho mínimo do bloco	8×8	4×4
Exactidão do vector de movimento	1/2 ou 1/4 de <i>pixel</i>	1/4 de <i>pixel</i>
Transformada	8×8 DCT	Aproximação 4×4 DCT
Filtro de <i>deblocking</i> incorporado	Não	Sim
Requer pagamento de licença para implementações comerciais	Sim	Provavelmente não (perfil base); sim para os perfis principal e expandido.

3.6 RADIUS CINEPAK CODEC

A descrição apresentada nesta secção baseia-se na referência [Ferguson, 2001].

O *codec* Cinepak é uma técnica de codificação relativamente antiga que ainda hoje é utilizada. A principal vantagem deste *codec* reside no facto do seu descodificador ser muito simples, e não no bit *rate* versus qualidade de desempenho. Este *codec* é basicamente um quantificador em vector com densidade de vector adaptativa. Cada *frame* é segmentado em blocos de 4×4 *pixels*, e cada bloco é codificado utilizando 1 ou 4 vectores, os tipos de codificação estão classificados como:

- V1 – um vector por bloco;
- V4 – quatro vectores por bloco.

Cada um destes tipos de codificação, V1 e V4, referência *codebooks* (dicionários, livros de código) independentes, que são denominados *codebook* V1 e *codebook* V4. Estes *codebooks* contêm no máximo 256 entradas cada um.

Um *frame* é igualmente segmentado para *strips* (tiras) de comprimento variável. Uma *strip* define uma área do *frame*, definida com dimensões menores ou iguais às do *frame*. Cada *strip* define o seu próprio par de vector *codebooks* único.

Um *frame* pode ser codificado utilizando: 8 bits por *pixel* (bpp), ou 12 bpp. No modo 12 bpp, cada vector de *codebook* contém quatro valores de oito bits para a luminância, e dois valores de 8 bits subamostrados para a crominância.

$$\left\{ \begin{matrix} y_0 & y_1 \\ y_2 & y_3 \end{matrix} \right\} \{u\} \{v\}.$$

No modo de 8 bpp, o *codebook* contém apenas os quatro valores da luminância. A conversão do espaço de cor RGB para o espaço de cor Cinepak é efectuada utilizando a seguinte multiplicação de matrizes:

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 & 2.0 \\ 1.0 & -0.5 & -1.0 \\ 1.0 & 2.0 & 0.0 \end{bmatrix} \times \begin{bmatrix} y \\ u \\ v \end{bmatrix}.$$

Fazendo a inversa da matriz (3×3) obtém-se,

$$\begin{bmatrix} y \\ u \\ v \end{bmatrix} = \begin{bmatrix} 0.2857 & 0.5714 & 0.1429 \\ -0.1429 & -0.2857 & 0.4286 \\ 0.3571 & 0.2857 & -0.0714 \end{bmatrix} \times \begin{bmatrix} r \\ g \\ b \end{bmatrix}.$$

Esta técnica é diferente das habitualmente utilizadas na conversão de espaços de cor e foi escolhida devido à sua simplicidade matemática em detrimento ao seu desempenho perceptual.

Como foi referido anteriormente, um bloco de 4×4 *pixels* pode ser codificado utilizando um vector de oito bits, denominado V1, ou quatro vectores de oito bits denominado V4. Estes vectores referenciam os *codebooks* V1 e V4 respectivamente. No caso da codificação de bloco V1, o vector *codebook* é utilizado para codificar o bloco da seguinte forma:

$$\begin{Bmatrix} y_0 & y_0 & y_1 & y_1 \\ y_0 & y_0 & y_1 & y_1 \\ y_2 & y_2 & y_3 & y_3 \\ y_2 & y_2 & y_3 & y_3 \end{Bmatrix} \begin{Bmatrix} u & u \\ u & u \end{Bmatrix} \begin{Bmatrix} v & v \\ v & v \end{Bmatrix}.$$

Na codificação de bloco V4, para entradas da tabela *codebook* é utilizado o código do bloco.

$$\begin{Bmatrix} r_0y_0 & r_0y_1 & r_1y_0 & r_1y_1 \\ r_0y_2 & r_0y_3 & r_1y_2 & r_1y_3 \\ r_2y_0 & r_2y_1 & r_3y_0 & r_3y_1 \\ r_2y_2 & r_2y_3 & r_3y_2 & r_3y_3 \end{Bmatrix} \begin{Bmatrix} r_0u & r_1u \\ r_2u & r_3u \end{Bmatrix} \begin{Bmatrix} r_0v & r_1v \\ r_2v & r_3v \end{Bmatrix}.$$

Um *frame* típico de uma sequência de vídeo Cinepak apresenta a seguinte estrutura:

<i>Frame Header</i>
<i>Strip 1 Header</i>
<i>Strip 1 Codebook</i>
<i>Strip 1 Frame Vectors</i>
<i>Strip 2 Header</i>
<i>Strip 2 Codebook</i>
<i>Strip 2 Frame Vectors</i>
<i>Strip 3 Header</i>
...

Figura 3.5 – Estrutura de um *frame* típico de uma sequência de vídeo Cinepak¹²

3.7 MICROSOFT RUN LENGTH ENCODING - RLE

A descrição apresentada nesta secção baseia-se na referência [MSI_A, 2005].

A versão do Windows 3.0 e as posteriores suportam formatos de codificação *run-length* (RLE), para comprimir *bitmaps* (BMP) que utilizam 4 ou 8 bits por *pixel*.

Quando o elemento da estrutura do cabeçalho do BMP, BITMAPINFOHEADER, denominado *biCompression* é colocado a BI_RLE8, o *Device Independent Bitmap* (DIB) é comprimido utilizando um formato de codificação *run-length* para um *bitmap* de 256 cores. Este formato utiliza dois modos: o modo *encoded* e o modo absoluto.

3.7.1 MODO ENCODED

No modo *encoded* uma unidade de informação consiste em 2 bytes. O primeiro byte especifica o número de *pixels* consecutivos a serem desenhados utilizando o índice de cor contido no segundo byte. O primeiro byte do par pode ser colocado a zero para indicar um fim da linha, de *bitmap* ou um delta (*offset*). A interpretação depende do valor do segundo byte do par, que deve estar dentro da gama 0x00 até 0x02. Na tabela seguinte (Tabela 3.2) apresentam-se os possíveis valores que podem ser utilizados no segundo byte.

Tabela 3.2 – Possíveis significados do segundo byte

Segundo byte	Significado
0	Fim da linha.
1	Fim do <i>bitmap</i> .
2	Delta. Os dois bytes a seguir contêm valores sem sinal que indicam os deslocamentos horizontal e vertical do próximo <i>pixel</i> a partir da posição actual.

¹² Adaptado de [Ferguson, 2001]

3.7.2 MODO ABSOLUTO

O modo absoluto é assinalado colocando o primeiro byte do par a zero e o segundo byte a um valor entre 0x03 e 0xFF. O segundo byte representa o número de bytes que se seguem, cada um contendo o índice de cor de um único *pixel*. Seguidamente (Tabela 3.3) apresenta-se um exemplo de um *bitmap* RLE de 8 bits (os valores de dois dígitos hexadecimais apresentados na segunda coluna representam o índice de cor para um único *pixel*).

Tabela 3.3 – Exemplo de um *bitmap* RLE de 8 bits

Dados comprimidos	Dados expandidos
03 04	04 04 04
05 06	06 06 06 06 06
00 03 45 56 67 00	46 56 67
02 78	78 78
00 02 05 01	Move 5 posições para a direita e uma posição para baixo
02 78	78 78
00 00	Fim da linha
09 1E	1E 1E 1E 1E 1E 1E 1E 1E 1E
00 01	Fim do <i>bitmap</i>

3.8 MICROSOFT VÍDEO – 1

A descrição apresentada nesta secção baseia-se na referência [MSI_B, 2005].

O *codec* Microsoft Vídeo – 1 utiliza blocos de 4x4 *pixels*, o que implica que os dados a serem comprimidos têm que ser divisíveis por 4, tanto em largura como em altura. À semelhança da decodificação de uma imagem Windows do tipo BMP, decodificar os dados de um *frame* Vídeo-1 é uma operação *bottom-to-top*.

O *codec* Vídeo-1 possui duas variantes, uma codifica dados de 8 bits, nos quais a paleta é armazenada no cabeçalho do ficheiro *Audio Video Interleave* (AVI). A segunda variante codifica 16 bits de cor. Na verdade, as cores são dados BGR555, e o bit de topo é algumas vezes utilizado para codificar informação extra. Para determinar qual das variantes é utilizada por um determinado ficheiro AVI, é necessário analisar o cabeçalho do ficheiro AVI, mais concretamente o campo *biBitCount* do cabeçalho do *bitmap* que se encontra no cabeçalho do ficheiro AVI.

Os dados Vídeo-1 são transportados em ficheiros AVI e *Advanced Streaming Format* (ASF) da Microsoft.

3.8.1 DADOS VÍDEO-1 DE 8 BITS

Na variante de 8 bits do Vídeo-1, os valores de 8 bits descodificados são índices para a tabela de cor armazenada no ficheiro AVI que transporta os dados Vídeo-1.

Para descodificar os dados Vídeo-1, lê-se o *bytestream* codificado copiando os dois primeiros bytes para *byte_a* e *byte_b*, respectivamente. O valor do *byte_b* indicará qual a próxima acção.

Se (*byte_a* = 0 & *byte_b* = 0) & (todos os blocos 4x4 de imagem foram percorridos) então a descodificação terminou.

Quando ($0x84 \leq \text{byte_b} < 0x88$) este código especifica que um determinado número de blocos deve ser saltado no *stream* de saída. Esta operação é útil para diferenciar *frames* (codificar diferenças de *frame* para *frame*) atingindo desta forma maior compressão.

3.8.1.1 CODIFICAÇÃO DE 2 CORES E 8 CORES

Nesta codificação, os próximos 2 bytes no *bytestream* codificado são *color_a* e *color_b*, respectivamente. Os *byte_a* e *byte_b* tornam-se conjuntos de *flags* que representam qual a cor, “a” ou “b”, a colocar num determinado *pixel* no bloco descodificado.

No caso da codificação de oito cores, um único bloco 4x4 pode ter no máximo oito cores. Na verdade o bloco 4x4 é subdividido em 4 de 2x2, cada um dois quais com duas cores cada. À semelhança do que acontece na codificação de duas cores, também neste caso o *byte_a* e *byte_b* se tornam num conjunto de *flags* que indicam qual a cor do *pixel* a colocar no bloco descodificado.

3.9 INTEL IYUV

A descrição apresentada nesta secção baseia-se na referência [FOURCC, 2005].

Os formatos YUV estão divididos em duas categorias, os formatos *packed* onde as amostras Y, U (Cb) e V (Cr) são conjuntamente empacotadas em *macropixels* as quais são armazenadas num único vector, e os formatos *planar* nos quais cada componente é armazenada como sendo vectores independentes, a imagem final é a fusão dos três planos separados. O IYUV está inserido na categoria *packed*.

O sufixo numérico de cada uma das amostras Y, U e V indica a posição de amostragem através da linha da imagem. Assim, V_0 indica a amostra mais à esquerda de V e Y_n indica a amostra Y do *pixel* (n+1) a partir da esquerda.

Os intervalos de subamostragem descrevem com que frequência através da linha é retirada uma amostra desse componente. No caso do intervalo vertical este indica quais as linhas de amostra retiradas. Por exemplo, o formato UYVY possui um período horizontal de subamostragem de 2 para as componentes U e V, isto indica que as componentes U e V são adquiridas a cada segundo *pixel* ao longo da linha. O seu período de subamostragem vertical é 1 o que indica que os valores de U e V são adquiridos em cada linha da imagem.

O UYVY é o mais popular dos formatos YUV 4:2:2. Este possui 16 bits efectivos por *pixel*. Na Tabela 3.4, estão apresentados os períodos de amostragem de cada uma das componentes Y, V e U.

Tabela 3.4 – Período de amostragem das componentes YUV

	Horizontal	Vertical
Período de amostragem Y	1	1
Período de amostragem U	2	1
Período de amostragem V	2	1

O IYUV é idêntico ao UYVY com a excepção de os seus dados serem entrelaçados. As linhas estão ordenadas (0, 2, 4, 6, ... em vez de 0, 1, 2,3,4,5, ...).

CAPÍTULO 4 - ALGORITMO DE RECONSTRUÇÃO

Neste capítulo é apresentado um algoritmo de reconstrução de sinal, que é utilizado para efectuar a reconstrução dos sinais extraídos da sequência de *frames*, a versão discreta do algoritmo Papoulis [Papoulis, 1975]. Este é o algoritmo utilizado na metodologia proposta neste trabalho e descrita no capítulo 5. Outras técnicas semelhantes descritas em [Pei e Yeh, 1997; Ferreira, 1999] podiam ser utilizadas.

Neste capítulo, é feita uma introdução à notação e terminologia utilizadas e seguidamente é descrito o algoritmo iterativo e indicadas quais as restrições de utilização do referido algoritmo. No final do capítulo, são analisadas as condições de convergência do algoritmo.

O algoritmo Papoulis-Gerchberg pode ser utilizado para recuperar amostras perdidas em sinais de tamanho finito, de dados de banda-limitada [Ferreira_A, 1994]. Neste caso, não é feita nenhuma suposição relativamente à distribuição das amostras perdidas, ao contrário do que acontece com a maioria dos métodos estudados, nos quais as amostras conhecidas são contíguas. É possível considerar o sinal observado como sendo uma versão amostrada do sinal original, e interpretar o resultado da reconstrução estudada como o resultado de uma amostragem.

Demonstra-se que o algoritmo de Papoulis-Gerchberg converge, se a densidade do conjunto de amostras excede um certo valor mínimo: este valor aumenta com a largura de banda dos dados [Ferreira_A, 1994]. É também salientado que o padrão de erros pode afectar a convergência do algoritmo de reconstrução. Por exemplo, para sinais de baixa frequência (*low-pass signals*) as melhores taxas de convergência são obtidas quando a distância entre as amostras perdidas é um múltiplo de um inteiro. As piores taxas de convergência ocorrem geralmente quando as amostras perdidas são contíguas [Ferreira_A, 1994].

4.1 NOTAÇÃO E TERMINOLOGIA

Considerando o espaço complexo de dimensão n , \mathbb{C}^n , com o produto interno e norma usuais, definimos *sinal* ou *sequência* de tamanho n , como sendo qualquer vector complexo x de dimensão n , com componentes, ou amostras, x_0, x_1, \dots, x_{n-1} . A matriz de *Fourier* F é a matriz unitária de dimensão $n \times n$ com componentes F_{mk} dados por

$$F_{mk} = \frac{1}{\sqrt{n}} e^{j \frac{2\pi}{n} mk}, \quad (4.1)$$

onde j representa a unidade imaginária. A transformada discreta de Fourier (DFT- *Discrete Fourier Transform*) de x , representada por \hat{x} , é por definição a sequência $\hat{x} = Fx$.

As duas operações, definidas em \mathbb{C}^n , nas quais estamos interessados, serão denominadas por *amostragem* e por *limitação de banda*.

Por definição, as *operações de amostragem* fazem o mapeamento da sequência numa outra, colocando a zero um subconjunto de amostras. Na forma matricial, isto corresponde à multiplicação de uma matriz diagonal D , contendo apenas zeros e uns. A diagonal de D será denominada de conjunto de amostras associada com a operação de amostragem D , e a própria matriz D será denominada de *matriz de amostragem*. A *densidade* de um conjunto de amostragem é por definição, s/n , sendo s o número de entradas não-nulas no conjunto de amostragem. Assume-se que $0 < s < n$, $D \neq I$, $D \neq 0$ [Ferreira_A, 1994].

Pode definir-se operador limitador de banda pela matriz B da forma $B = F^{-1}\Gamma F$, onde Γ é uma matriz de amostragem na frequência. Assim, de acordo com esta definição, a operação limitação em banda em \mathbb{C}^n é semelhante à operação de amostragem efectuada no domínio do tempo. Definiu-se largura de banda da sequência $y = Bx$ como sendo q/n , onde q é o número de entradas não-nulas na matriz Γ . Para sequências reais de baixa frequência, uma largura de banda igual a b significa que a maior frequência normalizada na sequência é $b/2$. As entradas não-nulas da diagonal de Γ [Ferreira_A, 1994] constituem a *banda de passagem* da matriz B .

4.2 OBJECTIVOS

Existem quatro modelos possíveis para o problema de extrapolação de banda-limitada, que correspondem a todas as combinações possíveis entre discreto e contínuo, tempo e frequência [Sanz e Huang, 1983]. Neste capítulo, é abordado o problema da reconstrução de sinais parcialmente conhecidos e limitados em banda (sinal observado), que implicará o conhecimento de um dado subconjunto de amostras da DFT dos dados, os harmónicos nulos, equivalente a modelo discreto-discreto. Os sinais são parcialmente conhecidos uma vez que, apenas um subconjunto das suas amostras se encontra disponível. A tarefa, consiste em, a partir das amostras disponíveis obter as amostras desconhecidas. Este problema apresenta semelhanças com o problema de amostragem, que basicamente consiste na reconstrução de

um sinal dado um subconjunto das suas amostras [Ferreira, 1996; Ferreira, 1999; Vieira e Ferreira, 1997; Reis, 2000].

Quando o padrão de erro é contíguo estamos perante um *problema de extrapolação*. Para uma determinada largura de banda podem ser recuperadas um número máximo de amostras desconhecidas. Este número aumenta com o decréscimo da largura de banda dos sinais. Isto pode ser interpretado como resultado da amostragem, garantido a possibilidade de reconstrução sem erros, de um sinal de banda-limitada, se um subconjunto suficientemente denso das suas amostras é conhecido.

O algoritmo de reconstrução utilizado é um método iterativo que se reduz à versão discreta de dimensão finita do algoritmo de extrapolação de Papoulis-Gerchberg, se as amostras desconhecidas são contíguas, e se a constante de relaxação $\mu = 1$ [Ferreira_A, 1994].

É muitas vezes conveniente conhecer os limites superior e inferior para a taxa de convergência de um algoritmo iterativo.

É também referida qual a natureza dos conjuntos de amostragem para os quais a taxa de convergência do algoritmo é a melhor ou a pior possível num determinado sentido. Verifica-se, que para o mesmo número de amostras desconhecidas, o pior padrão de erro para sinais passa-baixo será um conjunto contíguo de amostras perdidas [Ferreira_A, 1994]. Por outro lado, os conjuntos de amostragem óptimos são aqueles que surgem quando a distância entre as amostras perdidas é múltiplo de um inteiro. Para sinais que não são passa-baixo, isto não é necessariamente verdade. Os sinais passa-banda fornecem um exemplo de uma classe importante de sinais para os quais os piores conjuntos de amostragem podem não ser contíguos.

Demonstra-se em [Ferreira_A, 1994] que, para um conjunto de amostragem óptimo o raio espectral da matriz de iteração [Horn e Johnson, 1985], que é um importante indicador da taxa de convergência, é independente do número de amostras desconhecidas. De facto, um sinal com diversas amostras desconhecidas pode ser tão facilmente reconstruído como um sinal com apenas uma amostra desconhecida, isto no sentido de convergência assintótica, se as posições das amostras desconhecidas forem bem escolhidas. Neste trabalho, utiliza-se esta propriedade para que o problema da reconstrução seja sempre estável garantindo assim que a matriz de iteração é bem condicionada.

4.3 SINAIS DE BANDA-LIMITADA

Uma das condições para que possa ser aplicado o algoritmo de reconstrução aos sinais, é que estes sejam de banda-limitada. Para que esta condição seja garantida, será necessário neste caso filtrar todos os sinais, utilizando um Filtro Passa-Baixo (FPB), eliminando desta forma as altas frequências (ruído), e obtendo sinais de banda-limitada. Os sinais de banda-limitada serão a partir deste momento denominados *sinais observados*.

A limitação de banda dos sinais é obtida através de $y = Bx$ onde $B = F^{-1}\Gamma F$, onde y representa o sinal observado, e x o sinal original. Na Figura 4.1 estão apresentados graficamente os resultados das operações especificadas pelo operador B . Em primeiro lugar é efectuado o cálculo da DFT de x , no nosso caso utilizamos o algoritmo da *Fast Fourier Transform* (FFT) [Cooley e Tukey, 1965]. Depois de obtidas as componentes de x no domínio da frequência (Figura 4.1b), multiplica-se o sinal x pela matriz Γ , obtendo-se deste modo um sinal limitado na frequência (Figura 4.2a). Esta filtragem, no domínio da frequência, é equivalente a colocar a zero um determinado número de amostras conhecidas. As amostras que são colocadas a zero são as componentes centrais do espectro de frequência, uma vez que estas representam as componentes de frequência mais alta. Para obter o sinal observado no domínio do tempo calcula-se a *Inverse Fast Fourier Transform* (IFFT), do sinal limitado em frequência (Figura 4.2b).

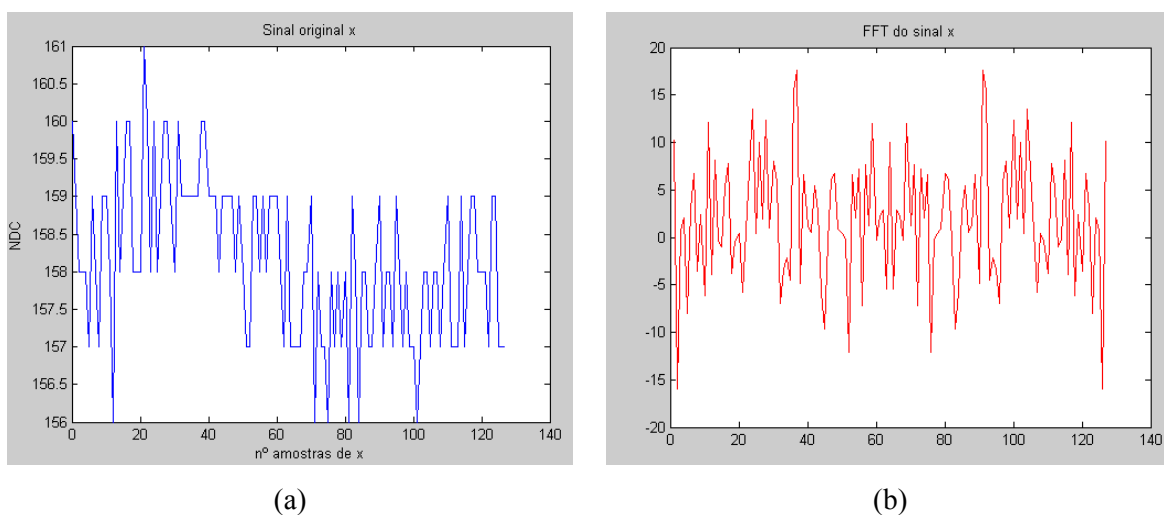


Figura 4.1 – Sinal não limitado na frequência (a) e respectiva FFT (b)

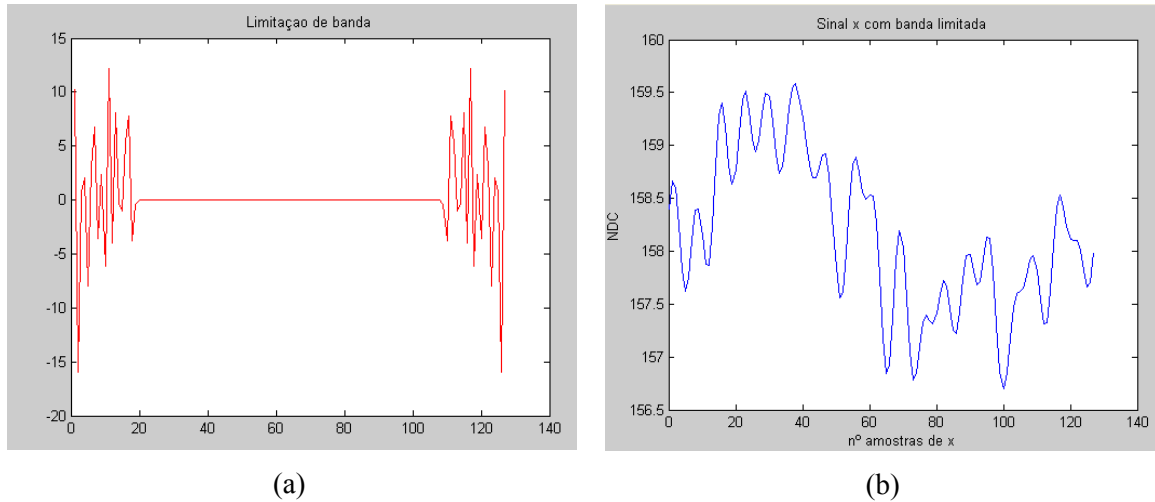


Figura 4.2 – Limitação de banda de um sinal utilizando o operador B

4.4 O ALGORITMO DE PAPOULIS-GERCHBERG

O algoritmo apresentado nesta secção é um algoritmo linear iterativo de 1ª ordem, com matrizes de iteração não-Hermiticas [Ferreira_A, 1994]. Neste algoritmo em particular, utilizam-se as operações de amostragem e limitação de banda definidas na secção 4.1, e mencionadas como distorção e restrição em [Ferreira_A, 1994]. O sinal original satisfaz a $x = Bx$ e o sinal distorcido é $y = Dx$, para as matrizes B e D adequadas. Isto sugere a seguinte equação

$$x = Bx + \mu(y - Dx) \quad (4.2)$$

$$x = \mu y + (I - \mu D)Bx \quad (4.3)$$

onde μ é uma constante fixa qualquer, que no nosso caso tomará o valor de 1, e I é a matriz identidade. A sequência das aproximações sucessivas é representada por

$$u_{n+1} = \mu y + (I - \mu D)Bu_n = T_1 u_n \quad (4.4)$$

onde

$$T_1(\cdot) = \mu y + (I - \mu D)B(\cdot) \quad (4.5)$$

O algoritmo de Papoulis-Gerchberg é baseado na filtragem seguida de amostragem. A dimensão do problema é máxima, ao contrário de outros métodos em que a dimensão do problema é do tamanho das amostras perdidas: neste caso, diz-se que a dimensão é mínima [Ferreira, 1996]. O esforço computacional envolvido está directamente relacionado com o número total de amostras do sinal e não com o número de amostras perdidas. Em [Ferreira_B, 1994; Soares, 2003] pode comprovar-se este facto.

Para sinais limitados em frequência, por exemplo, o sinal da Figura 4.3, é possível sob certas condições perder um determinado conjunto de amostras, sendo ainda possível a sua recuperação.

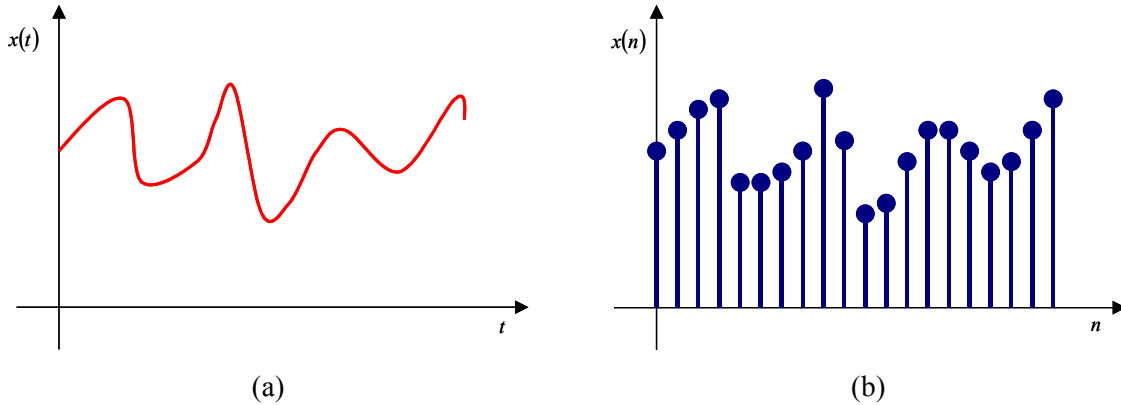


Figura 4.3 – (a) Sinal $x(t)$ contínuo limitado em frequência; (b) Sinal $x(t)$ discretizado

Admitindo um padrão de erro modelado pela função $r(n)$, obtém-se $x'(n)$ através da multiplicação de $x(n)$ por $r(n)$. As componentes nulas do sinal de $x'(n)$ (Figura 4.4b) representam as posições onde foram perdidas as amostras.

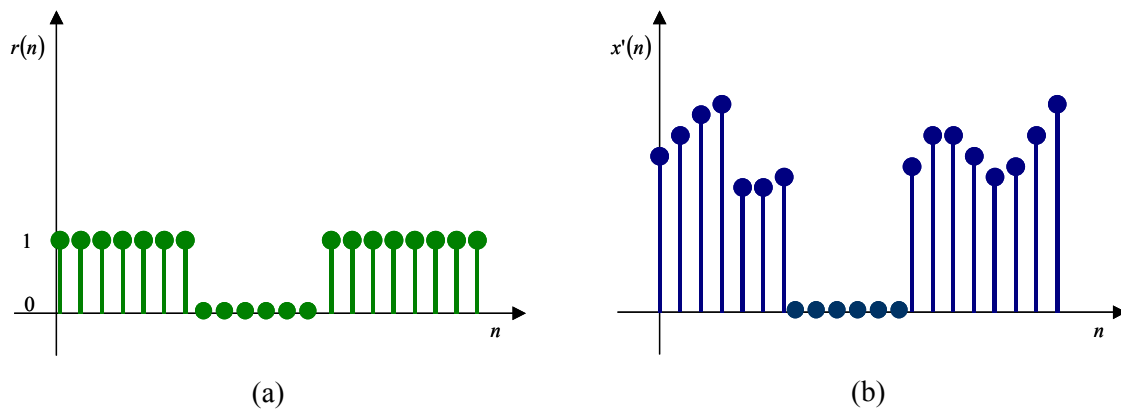


Figura 4.4 – (a) Trem de impulsos para amostrar o sinal $x(n)$; (b) Sinal $x(n)$ corrompido

O processo iterativo é descrito nas Figuras 4.5 e 4.6. A equação utilizada para implementar este algoritmo é a equação (4.3).

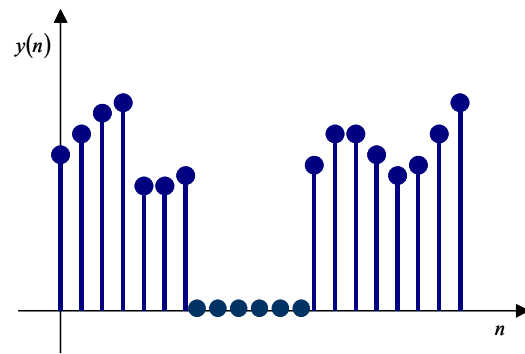
A primeira etapa consiste em calcular $x = Bx$, como já foi descrito na secção 4.3. A segunda etapa do processo iterativo consiste em multiplicar $(I - D)$ pelo resultado de Bx , onde D é a matriz de amostragem (neste caso o sinal da Figura 4.4a) e I é a matriz de identidade. Com esta operação consegue-se colocar a zero as amostras conhecidas do sinal, deixando as amostras desconhecidas com um valor diferente de zero (resultado da operação Bx). O

resultado desta operação pode ser visualizado na Figura 4.6a. Ao sinal resultante da operação anteriormente descrita é adicionado o sinal com as amostras desconhecidas $x = y + (I - D)Bx$. O resultado pode ser visualizado na Figura 4.6b.

Este processo repete-se até que o valor das amostras desconhecidas se aproxime do seu valor observado. Ao fim de k iterações estaremos afastados da solução do problema de um dado erro residual ε . Em cada iteração do algoritmo será medida a diferença entre os sinais obtidos na iteração k e na iteração $k - 1$. Quando esta diferença for inferior a um determinado valor, que será previamente estabelecido, o processo terminará. Essa diferença pode ser expressa matematicamente, por uma distância entre o sinal da iteração k e o da iteração $k - 1$. Quando esta diferença for inferior a um determinado erro ε o processo iterativo termina, uma vez que existe a garantia que o erro residual é inferior ou igual a ε .

Sinal observado com amostras desconhecidas

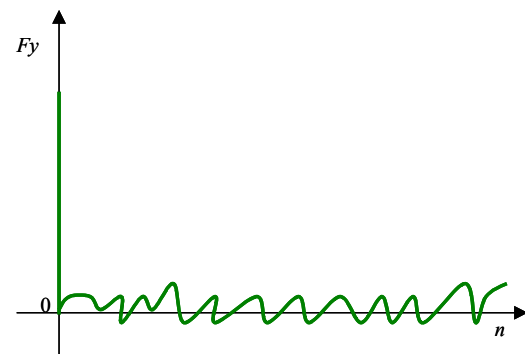
$$y = Dx$$



(a)

Transformada de Fourier de $y(n)$

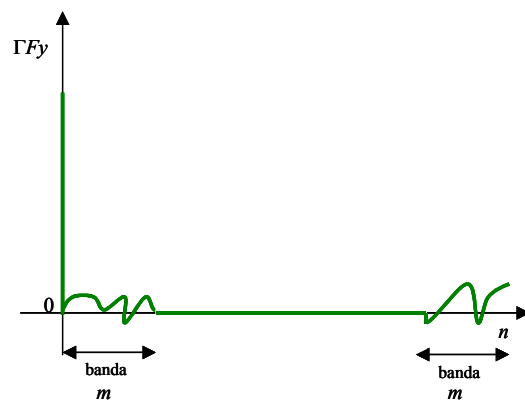
$$Fy$$



(b)

Limitação em frequência do sinal

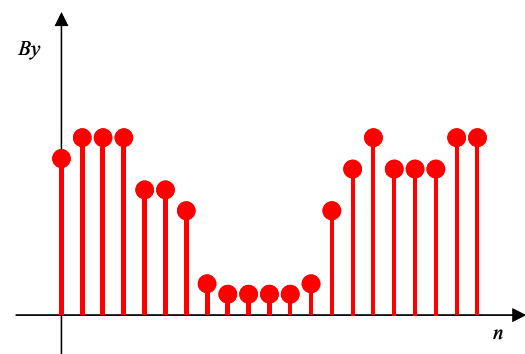
$$\Gamma Fy$$



(c)

Transformada Inversa de Fourier

$$F^{-1}\Gamma Fy = By$$



(d)

Figura 4.5 – Limitação de banda de um sinal

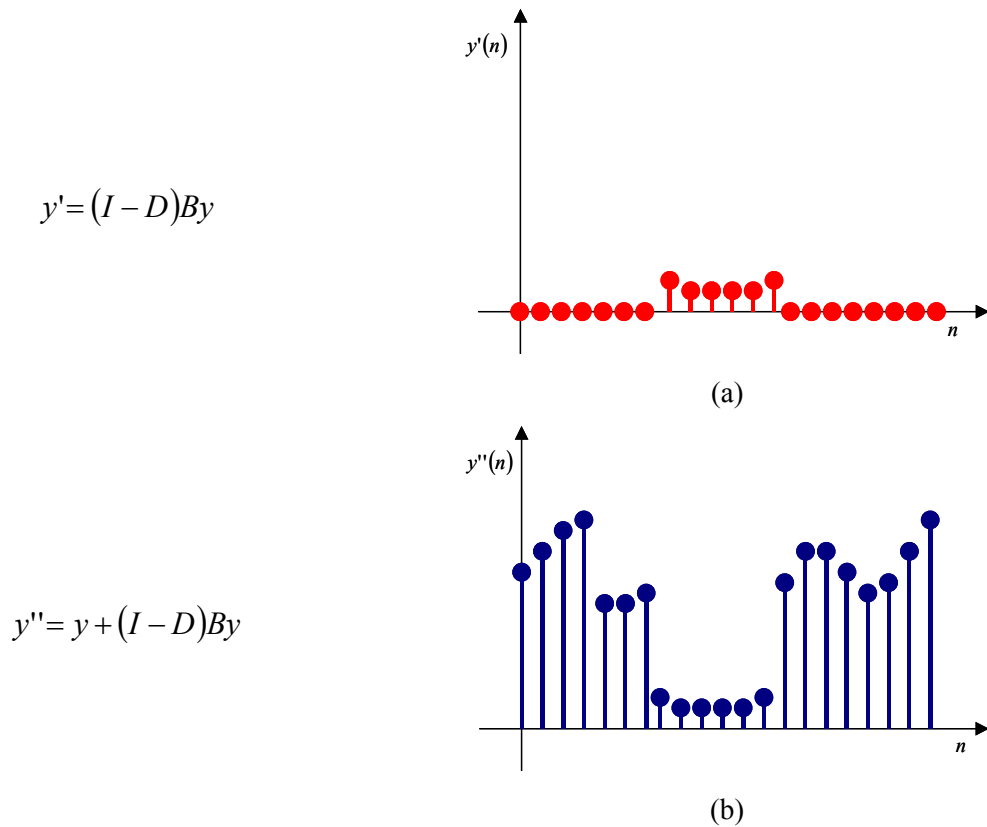


Figura 4.6 – Substituição das amostras conhecidas no sinal de banda-limitada

4.5 CONVERGÊNCIA DO ALGORITMO

Demonstra-se em [Ferreira_A, 1994] que o algoritmo converge nas seguintes condições: $0 < \mu < 2$ e $d \geq \omega$, onde $d = (n - q)$ e $\omega = (n - s)$. Lembra-se que n é o número de amostras do sinal, s é o número de entradas não-nulas da matriz D , e q é o número de entradas não-nulas da matriz Γ . No algoritmo implementado e descrito no capítulo 5, considerou-se $\mu = 1$, logo a primeira condição é sempre verificada. A segunda condição:

$$(n - q) \geq (n - s) \tag{4.6}$$

equivale a,

$$q \leq s \tag{4.7}$$

Como já foi referido anteriormente q é o número de entradas não-nulas da matriz Γ , ou seja, o número de amostras da banda de passagem do FPB, logo $q = 2m + 1$, onde m é a banda de amostras do FPB. Assim a equação (4.7) pode ser reescrita da seguinte forma:

$$s \geq 2m + 1 \tag{4.8}$$

Será através da equação (4.8) que será verificada a convergência do algoritmo de Papoulis-Gerchberg na metodologia aplicada e descrita no capítulo 5.

CAPÍTULO 4 – ALGORITMO DE RECONSTRUÇÃO

Para o caso de $\mu = 1$, é fornecido em [Ferreira_A, 1994] o limite inferior e superior para o erro de reconstrução em qualquer iteração. Os limites são os melhores possíveis e são fornecidos os vectores iniciais para os quais estes são alcançados.

CAPÍTULO 5 - METODOLOGIA APLICADA

Conforme já foi referido anteriormente, é pretendido neste trabalho conjugar técnicas de compressão, com técnicas de reconstrução de sinal. O objectivo é avaliar uma alternativa aos métodos de compressão mais tradicionais, através da inclusão de técnicas de reconstrução. Neste capítulo, é apresentada a metodologia aplicada para avaliar o comportamento da conjugação das duas técnicas.

O método consiste em aplicar um algoritmo de reconstrução de sinais a sequências sem compressão, e a sequências codificadas com diferentes métodos de compressão. Os métodos de compressão utilizados são os disponíveis no *software Paint Shop Pro 7*¹³ mais concretamente no *Animation Shop 3*, que permite codificar e descodificar os ficheiros AVI das sequências utilizando ou não compressão. As imagens (*frames*) da sequência são guardadas no formato **Bitmap** (BMP).

Admita-se uma sequência de imagens de alta resolução (225 *frames*/segundo). Se analisarmos quais as diferenças e variações entre dois *frames* consecutivos da sequência a conclusão a que chegamos é que as diferenças são, obviamente, muito pequenas e muitas vezes imperceptíveis ao olho humano (Figura 5.1).

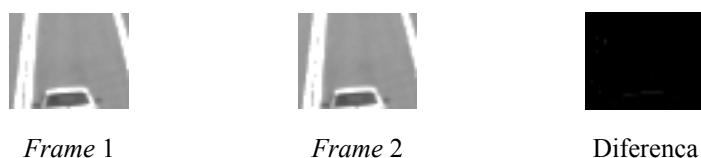


Figura 5.1 – Dois *frames* consecutivos e respectiva diferença entre ambos

Esta metodologia consiste em analisar, não os *frames* da sequência, mas sim a variação do valor de cada um dos *pixels* do *frame* ao longo de todos os *frames* da sequência. Isto acontece porque, em sequências de alta resolução, as diferenças entre *frames* consecutivos são muito pequenas, logo, é de esperar que o valor em NDC de grande maioria dos *pixels* do *frame* não sofrerá grandes variações ao longo da sequência. Tirando partido desta particularidade, pode-se estudar os sinais da sequência e verificar se é possível ou não representar esse sinal com menos amostras. Se o valor de um determinado *pixel*, ao longo da sequência, for sempre o mesmo necessitamos apenas de uma amostra para representar o seu

¹³ Paint Shop Pro 7.0, é um produto da *JASC Software* – URL: <http://www.jasc.com>

valor ao longo de toda a sequência. O objectivo é representar esses sinais com o menor número de amostras possível, garantido no entanto, uma qualidade aceitável, nos *frames* reconstruídos, para o tipo de utilização pretendida. Ou seja, guardar a sequência com o menor número de *frames* possível, mas garantindo que a reconstrução da sequência com o número de *frames* original é possível.

5.1 EXTRACÇÃO DOS SINAIS DE UMA SEQUÊNCIA

Foi desenvolvida uma aplicação em Matlab¹⁴ que permite extrair, analisar e guardar o sinal correspondente ao valor em NDC de cada um dos *pixels* dos *frames* de uma determinada sequência. No caso genérico, numa sequência com n *frames*, cuja dimensão dos *frames* é $M \times N$ *pixels*, sendo M e N o número de colunas e linhas respectivamente, cada sinal é composto por n amostras e o número de sinais que serão retirados da sequência para análise é $M \times N$. Na Figura 5.2 está descrito o processo de extracção dos sinais de uma sequência.

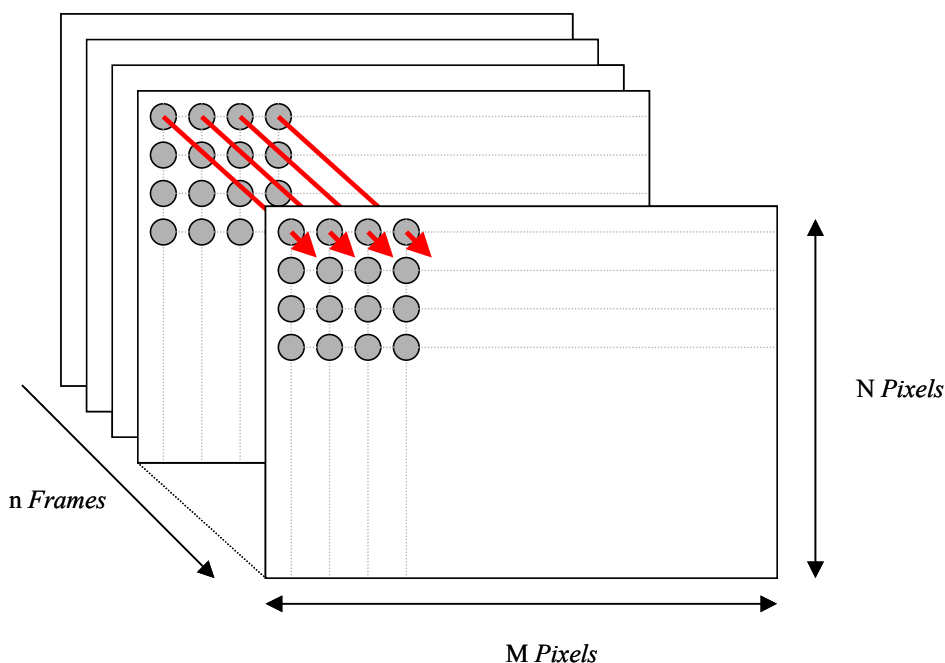


Figura 5.2 – Processo de extracção dos sinais de uma sequência

O processo consiste no seguinte: depois de lida a imagem para a matriz A , são guardadas todas as linhas de A , numa linha da matriz X . As dimensões de X são as seguintes, $M \times N$ colunas, sendo M e N a dimensão de A (imagem), e n linhas, onde n é como indicado

¹⁴ Matlab 6.5, é um produto da *MathWorks Inc.* – URL: <http://www.mathworks.com>

anteriormente, o número de *frames* da sequência. A Figura 5.3 exemplifica com maior detalhe este processo.

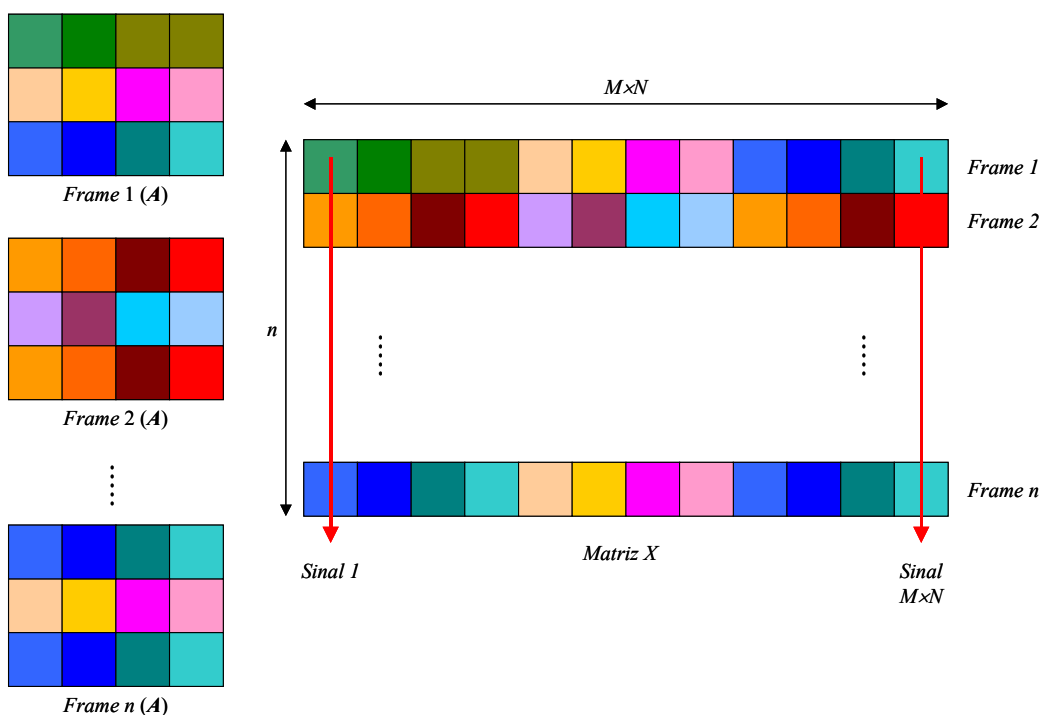


Figura 5.3 – Processo para guardar todos os *frames* da sequência numa única matriz

Cada um dos sinais extraídos representa o valor em NDC de cada um dos *pixels* dos *frames* ao longo dos *n frames* da sequência. O valor em NDC de um *pixel* ao longo de uma sequência está apresentado na Figura 5.4.

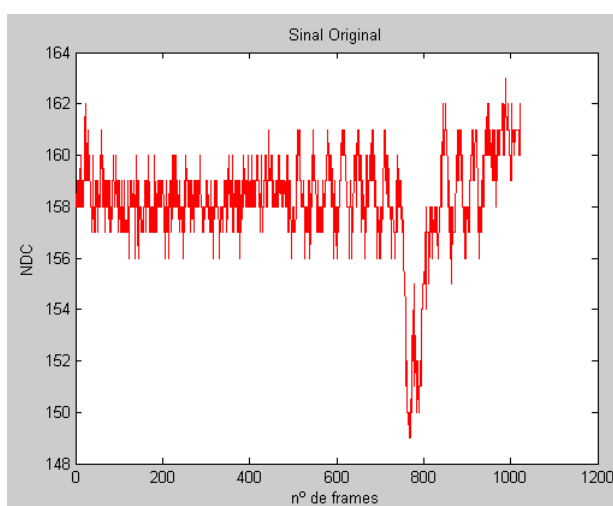


Figura 5.4 – Sinal correspondente aos níveis de cinzento de um *pixel* na sequência.

5.2 CARACTERÍSTICAS DAS SEQUÊNCIAS

Neste caso em particular, as duas sequências em estudo possuem 1024 *frames*. As características das sequências utilizadas para testar a metodologia são apresentadas na Tabela 5.1.

Tabela 5.1 – Características das sequências utilizadas

Sequência 1	Sequência 2
<ul style="list-style-type: none"> ❑ Taxa de aquisição: 225 <i>frames</i>/segundo ❑ Tamanho: 1024 <i>frames</i> ❑ Dimensão dos <i>frames</i>: 45×36 ❑ N° de sinais: 1620 ❑ Iluminação natural ❑ 8 bits por <i>pixel</i> 	<ul style="list-style-type: none"> ❑ Taxa de aquisição: 225 <i>frames</i>/segundo ❑ Tamanho: 1024 <i>frames</i> ❑ Dimensão dos <i>frames</i>: 35×31 ❑ N° de sinais: 1085 ❑ Iluminação natural ❑ 8 bits por <i>pixel</i>

No caso da sequência 1, as 45 colunas e 36 linhas perfazem um total de 1620 *pixels* por *frame*, o que corresponde à análise e extração de 1620 sinais cada um com 1024 amostras. Estes sinais representam o valor em NDC dos 1620 *pixels* ao longo do tempo (sequência). Relativamente à sequência 2, as 36 colunas e as 31 linhas constituem um total de 1085 *pixels* por *frame*, deste modo serão extraídos e analisados 1085 sinais com 1024 amostras cada. Estes sinais representam o valor em NDC dos 1085 *pixels* ao longo do tempo (sequência). Os sinais extraídos da sequência são denominados *sinais originais*, uma vez que representam o valor em NDC dos *pixels* ao longo da sequência.

5.3 FILTRAGEM DOS SINAIS EXTRAÍDOS

Depois de obtidos todos os sinais originais representativos da sequência em análise, estes são filtrados utilizando um FPB, para garantir que todos os sinais sejam de banda-limitada, e para que possa ser aplicado o algoritmo de reconstrução escolhido, o algoritmo discreto de Papoulis-Gerchberg [Ferreira_A, 1994]. Após a primeira filtragem, os sinais passam a ser denominados *sinais observados*, e possuem um número de amostras igual aos dos sinais originais, que no caso genérico será n . Na Figura 5.5 é possível observar um sinal extraído da sequência, sinal original, e o resultado da aplicação de um filtro passa-baixo a esse mesmo sinal, sinal observado. É igualmente importante referir que a largura de banda utilizada na primeira filtragem será a mesma do algoritmo de reconstrução.

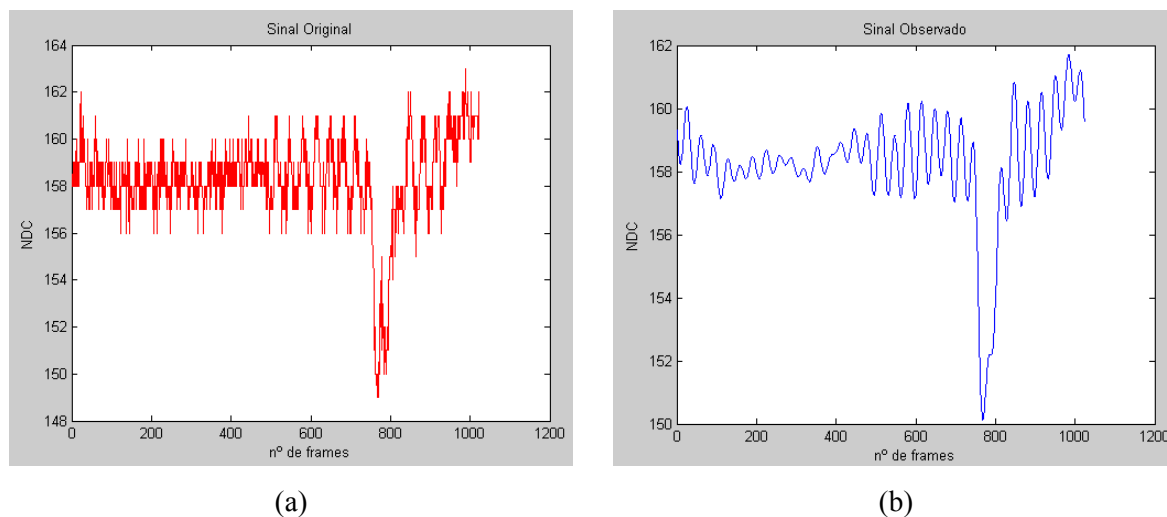


Figura 5.5 – (a) Sinal original; (b) Sinal observado

A operação descrita anteriormente pode ser entendida como uma operação de pré-processamento dos *frames* da sequência. O pré-processamento é o nome utilizado para operações em imagens ao mais baixo nível de abstracção (a entrada e a saída são a intensidade das imagens), como por exemplo realce do contraste. É importante referir que o pré-processamento não aumenta o conteúdo informativo das imagens. Se a informação da imagem é medida utilizando a entropia como definida em [Sonka *et al.*, 1999] então, o pré-processamento diminui tipicamente o conteúdo informativo da imagem. Do ponto de vista teórico, o ideal seria não efectuar nenhum tipo de pré-processamento, e a melhor forma de conseguir isso, seria efectuar uma aquisição de imagem de alta qualidade, obtendo desse modo imagens com alta definição. No entanto, o pré-processamento é muito útil em diversas situações uma vez que, ajuda a eliminar informação que não é relevante para o processamento ou tarefas de análise de imagem específicas. Assim, o objectivo do pré-processamento é o de melhorar os dados contidos na imagem, eliminando distorções indesejáveis ou dando maior ênfase a algumas características que são consideradas importantes para o processamento posterior [Sonka *et al.*, 1999].

Os métodos de pré-processamento local podem ser divididos em dois grupos de acordo com o objectivo. O primeiro, *smoothing*, cujo principal objectivo é eliminar ruído ou outras pequenas flutuações na imagem; é equivalente a eliminar as altas frequências no domínio da transformada de Fourier [Sonka *et al.*, 1999; Gonzalez e Wintz, 1987]. O segundo, baseado em operadores de gradiente, que são derivadas locais da função da imagem, permite identificar os locais da imagem nos quais se verificam maiores variações. Os operadores de gradiente possuem um efeito igual ao de eliminar as baixas frequências no domínio da transformada de Fourier [Sonka *et al.*, 1999; Gonzalez e Wintz, 1987].

O ruído é geralmente de alta frequência e ao efectuarmos esta primeira filtragem (ou *smoothing*) com algum cuidado na escolha da frequência de corte dos filtros, conseguimos eliminar algum desse ruído, sem alterar a informação contida nos *frames*.

O ruído contido nos *frames* da sequência pode ser originado por vários factores: à oscilação da câmara, provocadas por exemplo, pelo vento se a câmara não estiver bem fixa; ou no caso da aquisição ser efectuada em locais com iluminação artificial, ao cintilar das lâmpadas, uma vez que a frequência da rede eléctrica é de 50 Hz, e normalmente a frequência de aquisição é superior a esse valor, logo a intensidade luminosa entre *frames* pode variar drasticamente entre *frames* consecutivos. Existem equipamentos disponíveis no mercado que permitem ultrapassar o problema da iluminação [Infaimon, 2005; Phoenix, 2005].

5.4 DECIMAÇÃO DOS SINAIS

Após a primeira filtragem, os sinais observados são sujeitos a um processo de decimação, ou seja, são retiradas amostras. São retiradas as amostras pares (ou ímpares) o que equivale a considerar um valor de entrelaçamento igual a dois. O processo de perda de amostras não é aleatório, é feita uma primeira decimação de valor dois ao sinal observado, com n amostras, obtendo no fim um novo sinal de $n/2$ amostras. A próxima decimação será aplicada ao sinal de $n/2$ amostras e dará origem a um novo sinal com $n/4$ amostras, e assim sucessivamente. Na Figura 5.6 é apresentado a título de exemplo, o resultado do processo de decimação aplicado ao sinal observado da Figura 5.5.

Este processo origina 3 novos sinais com 512, 256 e 128 amostras, após as respectivas decimações.

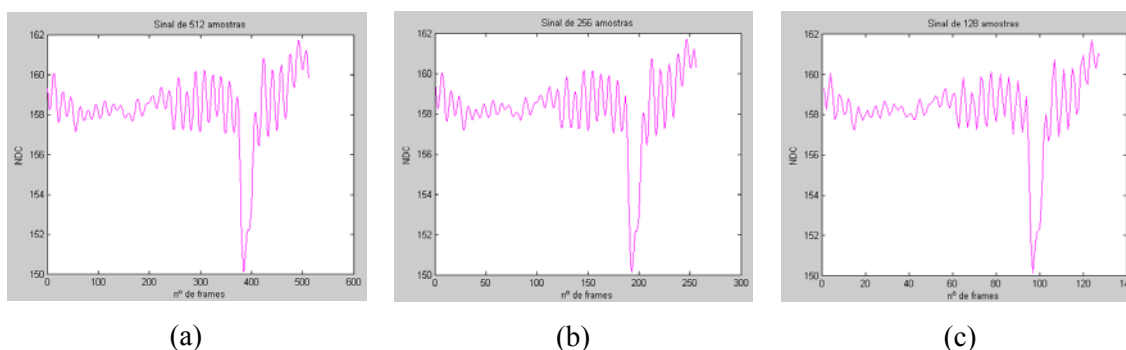


Figura 5.6 – Resultado da decimação com entrelaçamento igual a dois, sinais com, (a)512 amostras; (b) 256 amostras; (c) 128 amostras.

O processo descrito anteriormente é aplicado a todos os sinais observados da sequência, ou seja, aos $M \times N$ sinais.

Este método permite armazenar os sinais observados utilizando um menor número de *frames* e conseqüentemente um menor número de bytes, garantindo que a reconstrução do sinal seja possível para um determinado erro.

No exemplo da sequência de 1024 *frames*, é possível armazená-la utilizando um dos três sinais apresentados anteriormente, com 512, 256 ou 128 *frames*. A escolha do sinal a utilizar para representar o sinal observado depende do resultado obtido na reconstrução e do erro admissível em cada sinal. Este estudo é analisado no capítulo seguinte desta dissertação.

5.5 CONDIÇÕES DE CONVERGÊNCIA DO ALGORITMO DE RECONSTRUÇÃO

Neste caso cada um dos $M \times N$ sinais observados só poderá ser decimado um determinado número de vezes, isto para que seja possível a sua reconstrução. Para tal tem que verificar-se a condição de convergência do algoritmo de Papoulis-Gerchberg (equação 4.8) apresentada no capítulo anterior.

$$s \geq 2m + 1 \quad (5.1)$$

Relativamente à equação (5.1), m é a banda do filtro passa-baixo, e s é o número de amostras do sinal a reconstruir, como referido no capítulo anterior.

Como as sequências em análise possuem 1024 *frames*, todos sinais terão igualmente 1024 amostras. Se aos $M \times N$ sinais aplicarmos o processo de decimação descrito anteriormente, obtêm-se $M \times N$ sinais de 512, 256 e 128 amostras e para que seja possível a reconstrução a partir destes sinais tendo em conta que a condição da equação (5.1).

Na Tabela 5.2 é apresentado o valor mínimo de amostras que os sinais decimados devem possuir para que seja possível a reconstrução dos sinais observados. Os valores apresentados são para os casos em estudo, em que o número total de amostras dos sinais é 1024 (valor de n), e o entrelaçamento utilizado na decimação é dois. Os valores de l e s representam respectivamente, o número de amostras desconhecidas e o número de amostras do sinal a reconstruir.

Tabela 5.2 – Verificação da condição da condição de convergência do algoritmo de Papoulis-Gerchberg

Banda do Filtro	1ª Decimação	2ª Decimação	3ª Decimação	4ª Decimação
Passa-Baixo (m)	$l = 512$ $s = 512$	$l = 768$ $s = 256$	$l = 896$ $s = 128$	$l = 960$ $s = 64$
32	$512 \geq 65$ ✓	$256 \geq 65$ ✓	$128 \geq 65$ ✓	$64 \geq 65$ ✗
64	$512 \geq 129$ ✓	$256 \geq 129$ ✓	$128 \geq 129$ ✗	✗
128	$512 \geq 257$ ✓	$256 \geq 257$ ✗	✗	✗

✓ É garantida a convergência do algoritmo

✗ Não é garantida a convergência do algoritmo

Analisando a Tabela 5.2 é possível observar que, utilizando uma banda igual a 32 amostras no filtro passa-baixo do algoritmo de Papoulis-Gerchberg, podem perder-se no máximo 896 amostras do sinal observado. Sendo assim, aos $M \times N$ sinais observados com 1024 amostras cada, podendo aplicar no máximo três decimações, obtendo 3 conjuntos de $M \times N$ sinais, cada um com 512, 256 e 128 amostras. Se for aplicado o algoritmo de reconstrução a cada um desses conjuntos de sinais, a convergência do algoritmo é garantida para qualquer um dos casos. No entanto, se a banda de amostras for igual a 64 amostras, apenas se podem perder no máximo 768 amostras do sinal observado, o que corresponde a um máximo de duas decimações com entrelaçamento dois, que permitem obter dois conjuntos de $M \times N$ sinais, cada um com 512 e 256 amostras.

Pode concluir-se, que à medida que um sinal tem mais componentes na frequência, a sua reconstrução é, para as mesmas condições, mais difícil. Os conjuntos de sinais para os quais a reconstrução é mais difícil, utilizando este algoritmo, são mencionados no capítulo 4, secção 4.2.

Com as condições mencionadas anteriormente, a convergência do algoritmo de reconstrução é garantida. Se o valor da banda de amostras do filtro passa-baixo for igual a 128 amostras, é possível apenas decimar uma vez os $M \times N$ sinais observados, obtendo assim um novo conjunto de $M \times N$ sinais com 512 amostras: o processo está descrito na Figura 5.7.

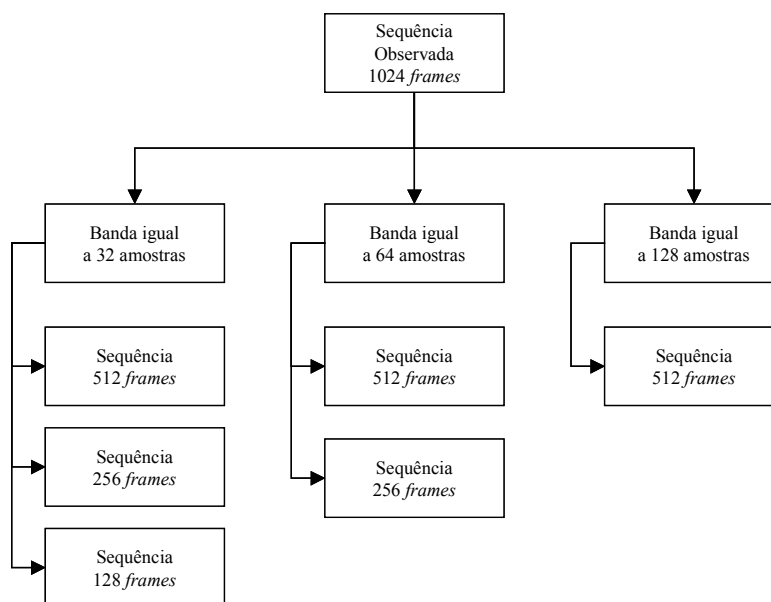


Figura 5.7 – Processo de decimação da sequência de acordo com a banda de amostras do filtro passa-baixo

5.6 ANÁLISE DO DESEMPENHO E CRITÉRIOS DE PARAGEM DO ALGORITMO DE RECONSTRUÇÃO

Depois de analisadas quais as condições para que o algoritmo convirja, foram efectuados testes para verificar confirmar este facto. Aquando do processo de decimação as amostras perdidas são colocadas a zero. Os $M \times N$ sinais decimados com as amostras desconhecidas colocadas a zero são parâmetros de entrada do algoritmo de reconstrução. Os outros parâmetros de entrada são a banda do filtro passa-baixo e o número de decimações, ou seja, o número de amostras colocadas a zero.

Como critérios de paragem do algoritmo, este pára quando o número máximo de iterações, é atingido (nos testes efectuados consideramos esse valor igual a 100) ou o erro obtido pelo cálculo da distância medida entre duas iterações consecutivas (equação 5.2).

$$E_{\text{iteração}} = \frac{\sqrt{\sum_{i=1}^n |s_r(k) - s_r(k-1)|}}{n} \quad (5.2)$$

Relativamente à equação (5.2), s_r representa o sinal obtido na iteração k , e n é o número de amostras dos sinais. Na prática calcula-se a diferença entre os sinais reconstruídos na iteração k e na anterior $(k-1)$ [Reis, 1996]. O valor utilizado nos testes para o erro máximo é 10^{-8} .

Depois de aplicado o algoritmo de reconstrução de Papoulis-Gerchberg a todos os $M \times N$ sinais, é novamente reconstruída a sequência de *frames* a partir dos sinais reconstruídos. Conforme já foi referido no capítulo anterior, este algoritmo utiliza a informação conhecida dos sinais tanto no domínio do tempo, como no da frequência, para encontrar o valor das amostras perdidas. Esses valores são determinados através de sucessivas filtrações no domínio da frequência. Em termos gerais, o algoritmo funciona da seguinte forma: é calculada a DFT das amostras temporais do sinal a reconstruir, isto para passar o sinal do domínio do tempo para o domínio da frequência, depois de obtida a representação em frequência do sinal, ou seja, após obtido o espectro de frequência do sinal, é efectuada uma filtração para eliminar as componentes de alta frequência. No espectro de frequência de um sinal, o primeiro componente da DFT é a componente DC do sinal, os componentes dos extremos são as baixas frequências, as altas frequências estão localizadas na parte central do espectro de frequências, sendo estas últimas as componentes que se pretende eliminar. Depois de eliminadas essas componentes, através da utilização de um filtro passa-baixo (a banda do filtro é a utilizada na 1ª filtração), o sinal é passado novamente para o domínio do tempo, através do cálculo da IDFT. Tendo novamente as amostras no domínio do tempo, é possível substituir nesse sinal as amostras conhecidas do sinal observado, e as desconhecidas já terão um valor diferente de zero. O processo repete-se até que o sinal reconstruído se aproxime do sinal observado a menos de um erro, de acordo com o algoritmo apresentado no capítulo 4. Na Figura 5.8, e a título de exemplo, está apresentado o resultado da reconstrução de um sinal de 1024 amostras a partir dos sinais de 512 e 128 amostras. A banda utilizada na filtração destas reconstruções foi igual a 32.

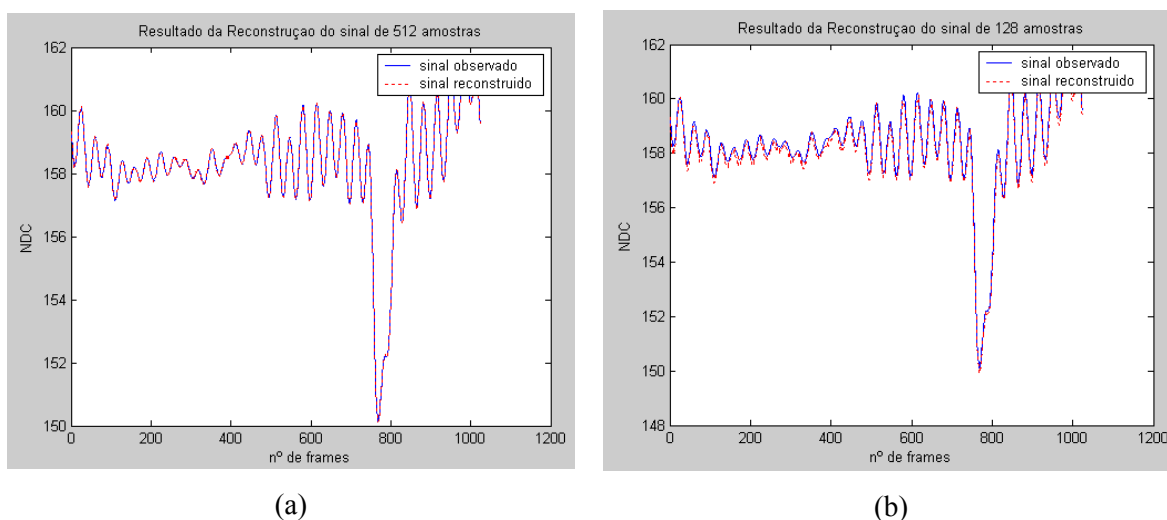


Figura 5.8 – Resultados da reconstrução de um sinal de: (a) 512 amostras; (b) 128 amostras

Na Figura 5.9. estão indicados todos os passos do algoritmo de reconstrução implementado, que são efectuados na reconstrução dos $M \times N$ sinais para a sequência 1 e sequência 2. O processo refere-se à reconstrução de um sinal, este processo é repetido de acordo com o número de sinais observados da sequência de *frames*. Em primeiro lugar é lido o sinal a reconstruir, e qual a banda de amostras do FPB, que é igual à utilizada na limitação de banda dos sinais, a seguir são estabelecidos os valores iniciais necessários (por exemplo: erro máximo e número máximo de iterações), posteriormente é iniciado o processo de reconstrução do sinal.

Sempre que o número de iterações for inferior ao máximo de iterações estipulado ou o erro da iteração for superior ao erro máximo, é executado o seguinte processo: é efectuada a operação de limitação de banda ao sinal, definida pelo operador B (secção 4.3 do capítulo 4), posteriormente são substituídas no sinal limitado em banda as amostras conhecidas do sinal a reconstruir, seguidamente é calculado o erro da iteração e incrementado o número da iteração. Quando o erro residual for inferior ao erro máximo ou se foi atingido o número máximo de iterações, o sinal está reconstruído. O passo seguinte é avaliar o desempenho do algoritmo de reconstrução.

Para avaliar e quantificar a qualidade da reconstrução da sequência, nomeadamente dos sinais, calcularam-se e analisaram-se três parâmetros, o **Erro** medido segundo a equação 5.2, o **RMSE** (*Root Mean-Squared Error*) segundo a equação 5.5 e o **PSNR** (*Peak Signal-to-Noise Ratio*) segundo a equação 5.3 de cada um dos sinais reconstruídos. O cálculo do erro final obtido na reconstrução de cada um dos sinais é efectuado de forma análoga ao cálculo do erro residual entre iteração (equação 5.2), a única diferença é que o erro final é determinado através da diferença entre o sinal observado e o sinal reconstruído.

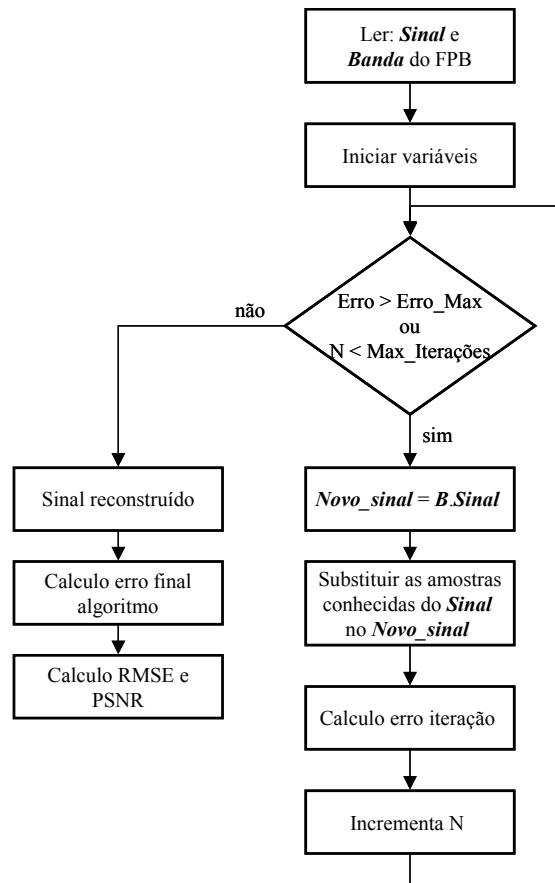


Figura 5.9 – Processo iterativo do algoritmo de reconstrução de sinal

Em relação ao cálculo do PSNR e RMSE, neste caso em particular utilizaram-se imagens em níveis de cinzento, com 8 bits por *pixel*, o PSNR em decibéis (dB) é calculado através da equação (5.3)

$$PSNR = 20 \log_{10} \frac{255}{RMSE} \text{ (dB)}. \quad (5.3)$$

O RMSE é definido como sendo

$$RMSE = \sqrt{\frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M [f(i, j) - \tilde{f}(i, j)]^2}, \quad (5.4)$$

onde M e N são respectivamente o número de colunas e número de linhas da imagem, f é a imagem original e \tilde{f} a imagem reconstruída, ambas com a mesma dimensão [Hilton *et al.*, 1994; MSA, 2004]. Neste caso em particular como o que estamos a reconstruir são sinais unidimensionais, o RMSE será definido como

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N [f(i) - \tilde{f}(i)]^2}, \quad (5.5)$$

em que N é o número de amostras do sinal f é o sinal original e \tilde{f} o sinal reconstruído, os dois sinais têm a mesma dimensão. Para cada reconstrução de sinal são calculados os respectivos RMSE (5.5) e PSNR (5.3) do sinal reconstruído. Para posterior análise estatística determinam-se os valores, máximo, médio e mínimo destes parâmetros. Determina-se igualmente qual a percentagem de sinais reconstruídos que possuem um PSNR acima do valor médio de PSNR obtido na reconstrução.

Resumindo, a aplicação desenvolvida permite efectuar a extracção de todos os $M \times N$ sinais originais, permite efectuar a limitação em frequência dos sinais, obtendo os denominados sinais observados, e conforme a banda de amostras escolhida é feita a decimação dos sinais observados. Esses novos sinais dão origem a novas sequências com menor número de *frames* que podem ser guardadas pelo utilizador. Caso se pretenda reconstruir a sequência observada, a partir de uma nova sequência com menor número de *frames*, apenas se tem que identificar qual. Depois de identificada qual a sequência a reconstruir, são colocados zeros entre as amostras conhecidas, dos sinais a reconstruir. O número de zeros colocados depende do número de decimações efectuadas anteriormente. No caso da sequência a reconstruir possuir 128 *frames*, os $M \times N$ sinais possuem 128 amostras conhecidas, para obter o número total de amostras dos sinais observados, 1024, é necessário colocar 896 zeros. Introduzir esses 896 zeros corresponde a colocar 7 zeros entre amostras sucessivas conhecidas dos sinais a reconstruir. Depois de identificados quais os sinais a reconstruir, é aplicado o algoritmo de Papoulis-Gerchberg. A análise do desempenho da reconstrução é feita através dos valores máximos, médios e mínimos dos parâmetros mencionados anteriormente, e duma análise estatística dos resultados obtidos. A aplicação permite igualmente a visualização gráfica desses resultados (RMSE e PSNR), permitindo ao utilizador verificar quais as zonas dos *frames* nas quais se verificou melhor qualidade do sinal reconstruído.

5.7 METODOLOGIA UTILIZADA NA GERAÇÃO DOS FICHEIROS

AVI

Seguidamente, enumeram-se os métodos de compressão e os parâmetros utilizados na criação dos ficheiros AVI utilizando o *software Animation Shop 3*.

Os métodos para comprimir e descomprimir as sequências foram os seguintes:

- RCC – *Radius Cinepak Codec*;
- IYUV – *Intel IYUV*;

- MV1 – *Microsoft Video 1.0*;
- RLE – *Run Length Encoding*.

Relativamente à geração dos ficheiros AVI os parâmetros são os definidos no *software* utilizado:

- *Color depth* – 8 bits por *pixel*;
- Se utilizada compressão é escolhido um dos métodos anteriormente mencionados;
- Número de fps é 100 (esta é a taxa máxima permitida pelo *software* utilizado).

Na Figura 5.10 está representado o resultado da compressão e descompressão dos *frames* da sequência 1, utilizando os métodos referidos anteriormente. O *frame* apresentado é o 400 da sequência 1.

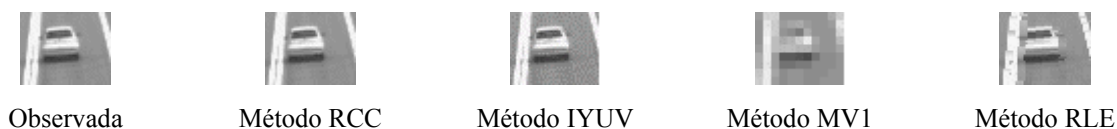


Figura 5.10 – *Frame* comprimido e descomprimido com diferentes métodos de compressão

Analisando as imagens da Figura 5.10 pode verificar-se que perceptivelmente os dois últimos métodos de compressão, MV1 e RLE, introduzem perdas nos *frames* da sequência, alterando desta forma a qualidade dos mesmos. Estas perdas traduzem-se naturalmente na diminuição do tamanho em bytes e consequentemente numa diminuição da necessidade de espaço em disco para os armazenar. Relativamente aos métodos RCC e IYUV não existem perdas. No entanto, e para este exemplo como se poderá verificar na Tabela 5.3, o ficheiro AVI gerado a partir destes métodos é maior.

A partir da sequência observada de 1024 *frames* e das versões decimadas, foram criadas 4 novas sequências para cada método de compressão, através do processo apresentado na Figura 5.11.

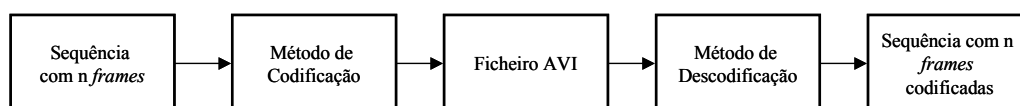


Figura 5.11 – Codificação/descodificação de sequências utilizando o formato AVI.

A sequência de 1024 *frames* obtida para cada um dos métodos vai servir de referência (Sequência RCC, Sequência IYUV, Sequência MV1 e Sequência RLE, Figura 5.12) para avaliar a capacidade de reconstrução do algoritmo.

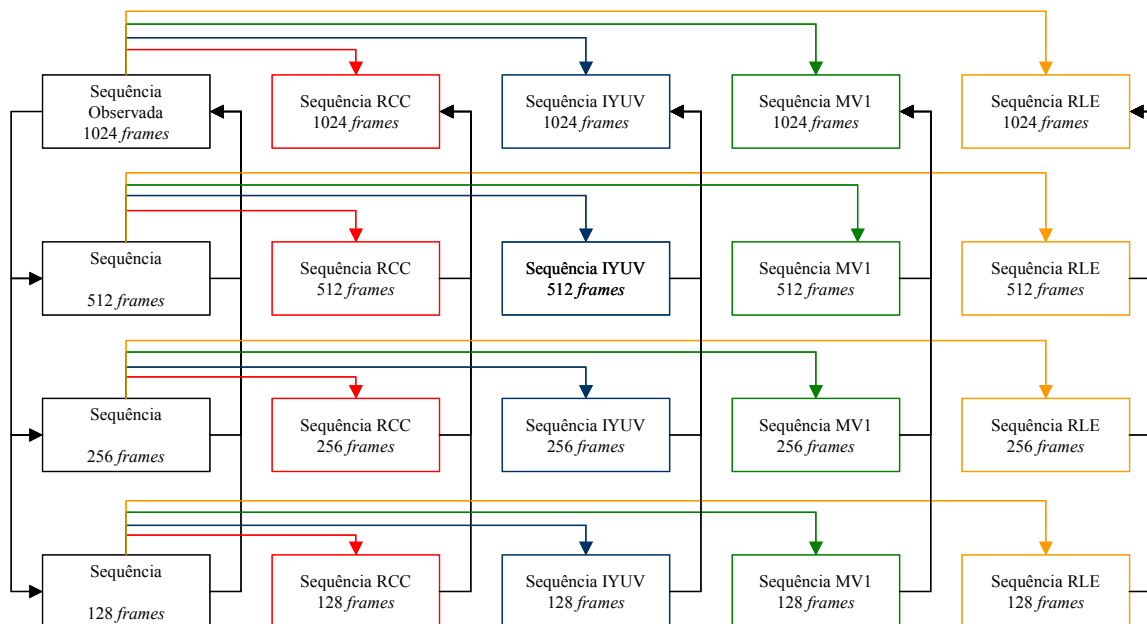


Figura 5.12 – Metodologia de criação dos ficheiros AVI e das novas sequências utilizando métodos de compressão

Na Tabela 5.3 apresenta-se o tamanho em bytes de todos os ficheiros AVI. A partir da análise desta tabela rapidamente se pode concluir que o espaço ocupado é muito menor para as sequências mais pequenas, como seria espectável. No próximo capítulo, apresentam-se os testes efectuados que provam que é possível a partir das sequências com menor número de *frames* reconstruir a sequência de 1024 *frames* respectiva, a menos de um erro. Esta metodologia traduz-se numa menor necessidade de espaço de armazenamento em bytes, para uma mesma sequência de *frames*.

Desta forma, propomos a alteração ao diagrama de blocos apresentado na Figura 5.11 para o processo de codificação/descodificação, Figura 5.13, como uma possível aplicação deste método.

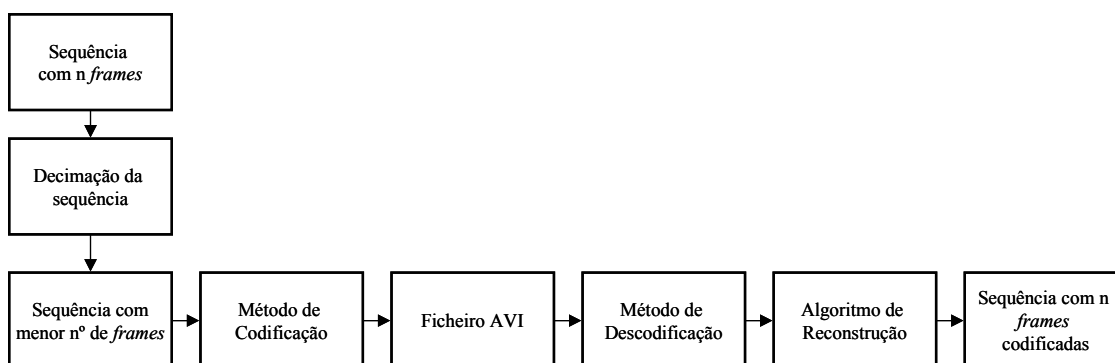


Figura 5.13 – Proposta para aplicação, da metodologia apresentada, em sequências codificadas/descodificadas utilizando o formato AVI.

Com esta proposta pretende contribuir-se para a redução do tamanho em bytes de sequências de alta resolução temporal como é o caso das duas sequências estudadas neste trabalho, por exemplo para o método MV1 a partir da sequência com 33Kb é possível obter 229Kb, Tabela 5.3.

Tabela 5.3 – Tabela comparativa dos ficheiros AVI criados utilizando ou não métodos de codificação

Compressão Nº Frames	Sem Compressão	Cinepak Codec by Radius	Intel IYUV	Microsoft Video 1	Run Length Encoding
1024	1757 kB	2056 kB	2403 kB	229 kB	1021 kB
512	881 kB	1029 kB	1203 kB	117 kB	513 kB
256	443 kB	516 kB	603 kB	61 kB	259 kB
128	224 kB	259 kB	303 kB	33 kB	132 kB

No capítulo 6, são apresentados os resultados para a sequência 1 obtidos com o algoritmo Papoulis-Gerchberg dentro dos limites de convergência do método, descrito na secção 4.5. Detalham-se também os resultados semelhantes para as Sequência RCC, Sequência IYUV, Sequência MV1 e Sequência RLE, obtidas a partir da sequência 1. Os resultados obtidos para sequência 2 podem ser observados no Anexo.

CAPÍTULO 6 - RESULTADOS

Neste capítulo, são apresentados os resultados obtidos na reconstrução da sequência 1 (sequência na qual se pode observar um veículo a circular no sentido Sul-Norte). O algoritmo proposto, foi também testado em sinais previamente comprimidos com e sem perdas conforme apresentado na secção 5.7. Os dados relativos ao desempenho do algoritmo de reconstrução utilizado, descrito no capítulo 4, estão apresentados em tabelas, nas quais é indicado o valor máximo, médio e mínimo para os seguintes parâmetros: erro final do algoritmo, RMSE e PSNR.

São igualmente apresentados gráficos a duas e três dimensões, 2D e 3D respectivamente, nos quais são expostos os resultados obtidos (RMSE e PSNR) na reconstrução de cada um dos sinais, representativos dos vários *pixels* dos *frames* da sequência.

Inicialmente são apresentados os resultados obtidos na reconstrução das sequências criadas a partir da sequência 1, e posteriormente, os resultados obtidos na reconstrução das sequências criadas a partir da sequência 2 (ver Anexo). Os resultados apresentados estão organizados de acordo com a banda de amostras utilizada no filtro passa-baixo. Os testes foram efectuados para bandas iguais a 32, 64 e 128 amostras, de acordo com a condição de convergência do algoritmo de Papoulis-Gerchberg apresentada na secção 4.5 e especificada na secção 5.5.

É igualmente feita uma análise dos ficheiros AVI criados com diversos algoritmos de codificação disponíveis no *Animation Shop 3.0* e apresentada uma tabela comparativa.

Posteriormente apresentam-se os resultados da reconstrução de sinais extraídos das sequências codificadas obtidas a partir dos ficheiros AVI criados com os métodos de codificação referidos na secção 5.7.

Uma vez que a metodologia proposta tem como alvo os sinais de alta resolução temporal, convém referir que ambas as sequências (1 e 2) são de alta resolução temporal, 225 *frames* por segundo (Tabela 5.1).

6.1 SEQUÊNCIA 1 – RESULTADOS PARA BANDA DE 32 AMOSTRAS

Esta sequência possui um total de 1620 (45×36) sinais cada um com 1024 amostras. Na Figura 6.1 podem observar-se 54 *frames* da sequência 1 observada. Apresentam-se os *frames*

do 90 ao 143 apenas para posterior comparação visual com os resultados obtidos na secção 6.1. Refira-se que de acordo com a Tabela 5.3 o ficheiro AVI desta sequência ocupa 1757kB.



Figura 6.1 – Alguns *frames* da sequência 1 observada, banda de 32 amostras no FPB

Na reconstrução do sinal observado (1024 amostras) a partir dos sinais de 512, 256 e 128 amostras, foram obtidos os resultados apresentados na Tabela 6.1, nesta tabela apresentam-se apenas os valores: máximo, médio e mínimo dos parâmetros analisados.

Os resultados seguintes, secção 6.1.1 à secção 6.1.3, foram obtidos utilizando uma banda de amostras do FPB igual a 32, isto para que a convergência do algoritmo esteja garantida de acordo com a Tabela 5.2.

Tabela 6.1 – Sequência 1 – Resultados obtidos na reconstrução dos sinais de 128, 256 e 512 *frames* sem compressão (banda de 32 amostras).

Nº de <i>Frames</i> do Sinal a Reconstruir	Banda de Amostras do FPB	Nº de Zeros entre <i>Frames</i>	Erro Final do Algoritmo de Reconstrução		RMSE		PSNR (dB)	
			Max	Med	Max	Med	Max	Med
128	32	7	Max	0.0334	Max	1.3417	Max	116.56
			Med	0.0134	Med	0.3065	Med	67.10
			Min	5.8820e-4	Min	3.7875e-4	Min	45.58
256	32	3	Max	0.0291	Max	1.1180	Max	198.65
			Med	0.0117	Med	0.2553	Med	73.51
			Min	5.0185e-6	Min	2.9779e-8	Min	47.16
512	32	1	Max	0.0218	Max	0.7607	Max	207.71
			Med	0.0088	Med	0.1737	Med	77.19
			Min	2.6921e-6	Min	1.0496e-8	Min	50.51

Uma vez que nos resultados apresentados na Tabela 6.1, não se podem identificar quais as zonas dos *frames* nos quais se obteve melhor desempenho na reconstrução dos sinais; para que isso seja possível apresentam-se posteriormente os resultados de forma gráfica. Nesses gráficos apresentam-se os valores de RMSE e PSNR obtidos na reconstrução dos sinais representativos de cada *pixel* dos *frames* da sequência.

6.1.1 SEQUÊNCIA 1 – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 128 *FRAMES*

Nas Figuras 6.2 e 6.3, estão apresentados os valores do RMSE e PSNR obtidos na reconstrução de cada um dos 1620 sinais de 128 amostras, para 1024 amostras, correspondentes aos sinais de cada um dos *pixels* dos *frames*. Os resultados apresentados estão normalizados.

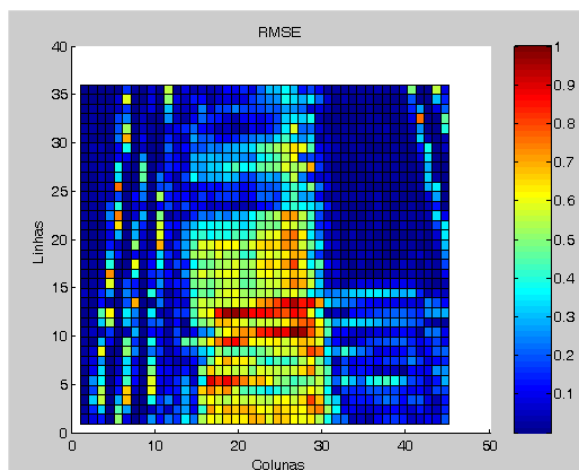
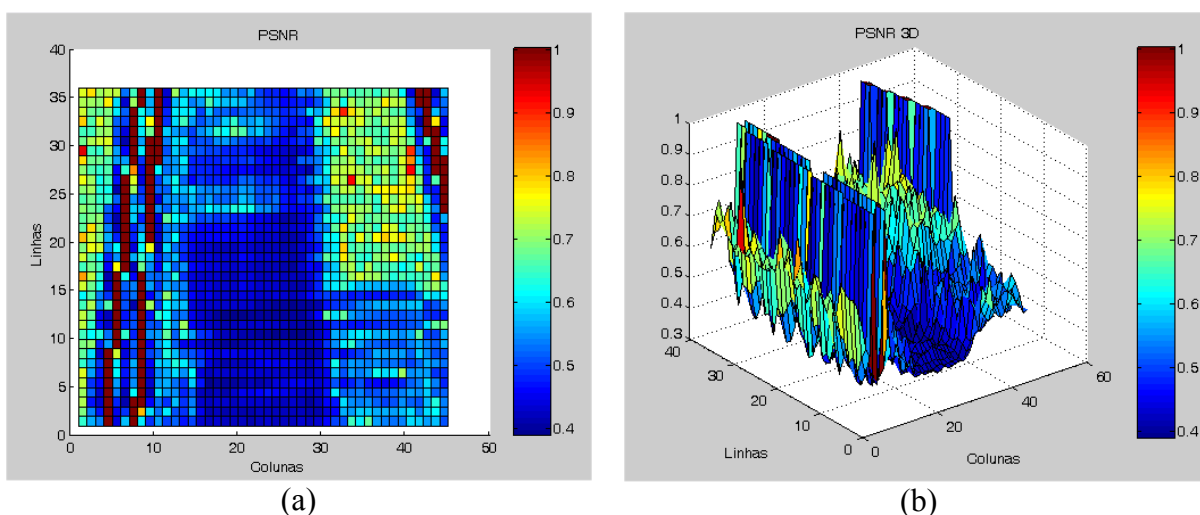


Figura 6.2 – Sequência 1, RMSE, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 128 *frames* (banda de 32 amostras)



(a)

(b)

Figura 6.3 – Sequência 1, PSNR, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 128 *frames* (banda de 32 amostras)

Da análise das Figuras 6.2 e 6.3, concluí-se que na reconstrução dos 1620 sinais observados de 1024 amostras a partir dos sinais de 128 amostras, as zonas dos *frames* da sequência em que se conseguiu obter uma melhor reconstrução, foram as zonas laterais dos *frames*. Os melhores resultados ocorreram na zona lateral direita dos *frames*, zonas castanhas na Figura 6.3. No entanto, na zona lateral esquerda dos *frames* também se obtiveram bons resultados na reconstrução. Isto porque da análise visual da sequência (ver exemplo nas Figuras 6.1 e 6.4 que contêm alguns *frames* da sequência observada e da sequência reconstruída) pode constatar-se que os sinais representativos dos *pixels* das zonas laterais dos *frames*, não possuem grande variação ao longo da sequência, uma vez que o veículo (objecto em movimento nesta sequência) descreve uma trajectória de Sul-Norte na parte central dos *frames*; o que é expectável uma vez que as regiões laterais correspondem a zonas em que a

variação dos sinais é menor. Nessas zonas, os sinais reconstruídos apresentam um PSNR superior a 67.10 dB, e conseqüentemente um RMSE mais pequeno. Após análise estatística dos resultados, verificou-se que dos 1620 sinais reconstruídos, 39.26% possuem uma relação sinal/ruído igual ou superior ao PSNR médio, que é 67.10 dB. Por outro lado, 60.74% dos sinais apresentam uma relação sinal/ruído inferior a 67.10 dB. Estes sinais são maioritariamente pertencentes à zona central dos *frames* da sequência, que são as zonas com um RMSE mais elevado. Os valores, máximo e mínimo de PSNR, obtidos nesta reconstrução são respectivamente, 116.56 dB e 45.58 dB.

Na Figura 6.4 são apresentados 54 *frames* da sequência de 128 *frames* reconstruída que podem ser comparados com os *frames* da Figura 6.1. O ficheiro AVI desta sequência ocupa 224kB (Tabela 5.3).



Figura 6.4 – Alguns *frames* da sequência 1 reconstruída a partir da sequência de 128 *frames*, banda de 32 amostras no FPB

6.1.2 SEQUÊNCIA 1 – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 256 FRAMES

Nas Figuras 6.5 e 6.6, estão apresentados os valores do RMSE e PSNR obtidos na reconstrução de cada um dos 1620 sinais de 256 amostras, para 1024 amostras, correspondentes aos sinais de cada um dos *pixels* dos *frames*. Os resultados apresentados estão normalizados.

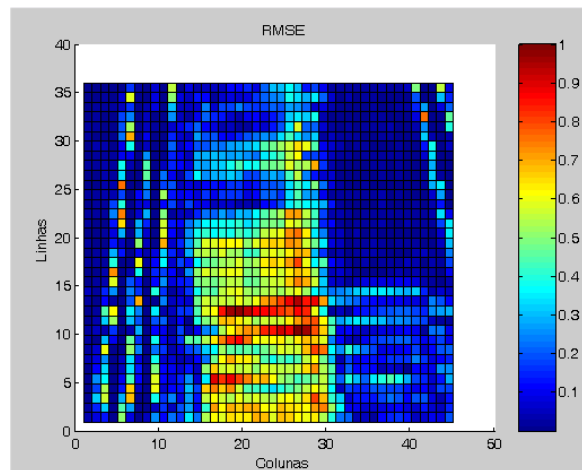


Figura 6.5 – Sequência 1, RMSE, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 256 *frames* (banda de 32 amostras)

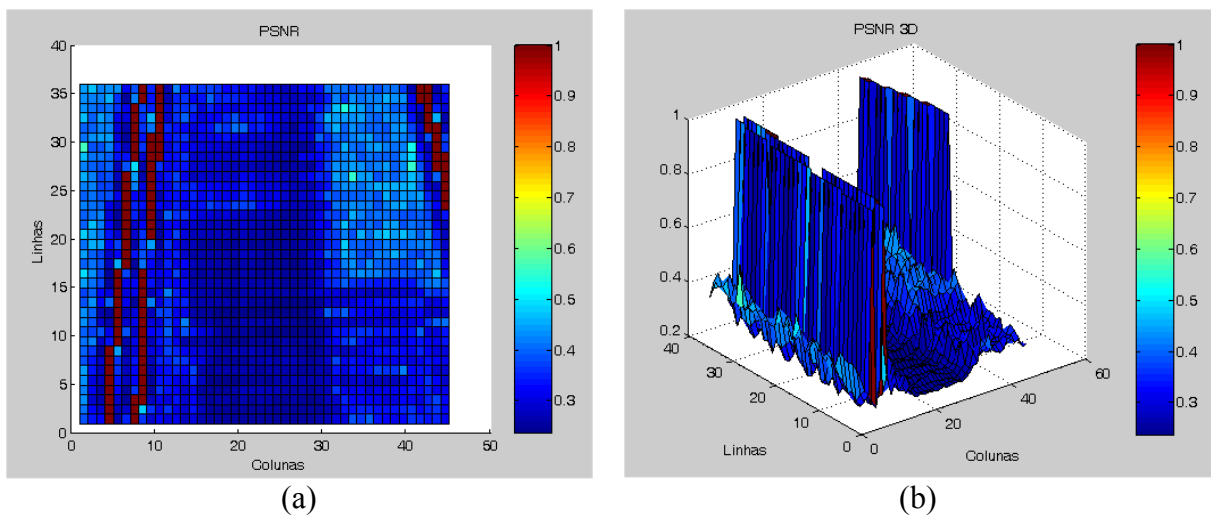


Figura 6.6 – Sequência 1, PSNR, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 256 *frames* (banda de 32 amostras)

Analisando os resultados apresentados nas Figuras 6.5 e 6.6, verifica-se que as zonas onde se obteve melhor reconstrução dos sinais, são igualmente as zonas laterais dos *frames*. A zona central dos *frames* continua, como no caso anterior, a ser a zona com menor relação sinal/ruído e maior RMSE.

CAPÍTULO 6 – RESULTADOS

O valor máximo, médio e mínimo de PSNR desta reconstrução são superiores aos valores obtidos na reconstrução dos sinais de 128 amostras, pois o número de amostras desconhecidas neste caso, é inferior. O valor máximo de PSNR é agora 198.65 dB, o mínimo passou a ser 47.16 dB. Em relação ao valor obtido na reconstrução anterior este aumentou aproximadamente 2 dB. Estatisticamente, dos 1620 sinais, 31.17% possuem um PSNR igual ou superior a 73.51 dB e 68.83% possuem um PSNR entre 47.16 dB e 73.51 dB.

Na Figura 6.7 são apresentados 54 *frames* da sequência de 256 *frames* reconstruída, que podem ser comparados com os *frames* da Figura 6.1. O ficheiro AVI desta sequência ocupa 443kB (Tabela 5.3).



Figura 6.7 – Alguns *frames* da sequência 1 reconstruída a partir da sequência de 256 *frames*, banda de 32 amostras no FPB

6.1.3 SEQUÊNCIA 1 – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 512 FRAMES

Nas Figuras 6.8 e 6.9, estão apresentados os valores do RMSE e PSNR obtidos na reconstrução de cada um dos 1620 sinais de 512 amostras, para 1024 amostras, correspondentes aos sinais de cada um dos *pixels* dos *frames*. Os resultados apresentados estão normalizados.

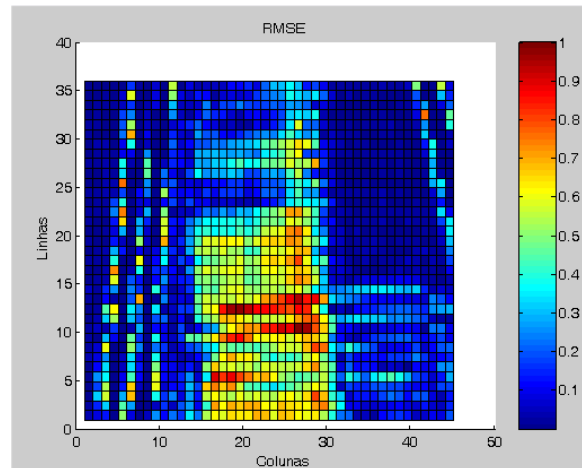


Figura 6.8 – Sequência 1, RMSE, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 512 *frames* (banda de 32 amostras)

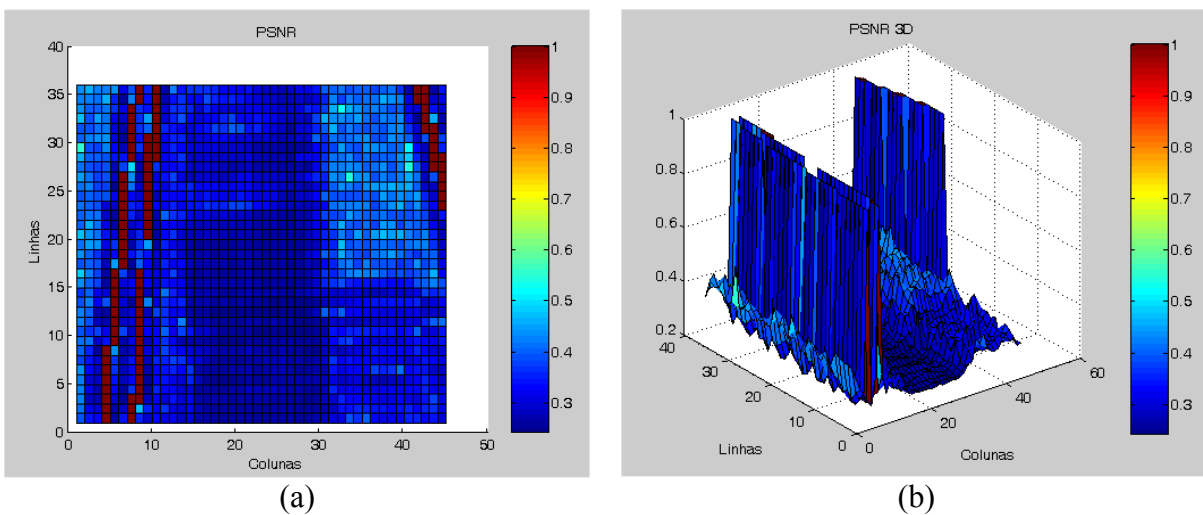


Figura 6.9 – Sequência 1, PSNR, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 512 *frames* (banda de 32 amostras)

Da análise da Figura 6.8, verifica-se que as zonas com maior RMSE são, naturalmente, as zonas centrais dos *frames* da sequência, os *pixels* desta zona apresentam uma relação sinal/ruído mais baixa, Figura 6.9.

Na reconstrução da sequência com 512 *frames*, 30.62% dos 1620 sinais possuem um PSNR maior ou igual ao valor médio, 77.19 dB, e menor ou igual ao PSNR máximo, 207.71 dB. Os sinais com um PSNR igual ao valor máximo representam sinais de *pixels* das zonas laterais dos *frames*, zonas a castanho na Figura 6.9. Verifica-se ainda, que uma maior percentagem de sinais (69,38%) apresenta um PSNR maior ou igual a 50.51 dB e inferior ao PSNR médio. Estes sinais representam maioritariamente *pixels* da zona central dos *frames* onde o erro obtido no algoritmo reconstrução é maior.

Em comparação com os resultados obtidos nas reconstruções anteriores, a partir das 128 e 256 *frames*, esta é a que apresenta valores mais elevados na relação sinal/ruído, atingindo mesmo valores superiores a 200 dB em alguns casos. Os resultados do PSNR são mais uniformes do que os dois anteriores, ou seja, conseguimos idêntico desempenho na reconstrução na maioria dos *pixels*, e as variações registadas identificam os melhores resultados, ou seja, a reconstrução dos *pixels* associados às zonas laterais dos *frames*.

Concluí-se com estas três experiências que as zonas onde o desempenho do algoritmo de reconstrução é melhor são as zonas onde a variação dos NDC dos *pixels* é menor. Concluimos também que à medida que o número de amostras desconhecidas diminui, se por um lado o valor da relação sinal/ruído aumenta, por outro a convergência do método iterativo é mais rápida. Nas secções 6.2 e 6.3, avaliam-se estes resultados utilizando valores diferentes para LB (64 e 128) do FPB de modo a caracterizar o comportamento do algoritmo nestes casos, Tabela 5.2.

Na Figura 6.10 são apresentados 54 *frames* da sequência de 512 *frames* reconstruída. O ficheiro AVI desta sequência ocupa 881kB (Tabela 5.3).

A observação dos *frames* reconstruídos a partir das sequências decimadas de 128, 256 e 512 *frames* leva a concluir que os resultados da reconstrução são perceptivelmente indistinguíveis da sequência observada (Figura 6.1). Desta forma, a representação da sequência observada pode ser realizada através de qualquer uma destas versões decimadas, que necessitam um espaço de armazenamento em bytes de aproximadamente 12,5%, 25% e 50% relativamente à sequência observada (Tabela 5.3)

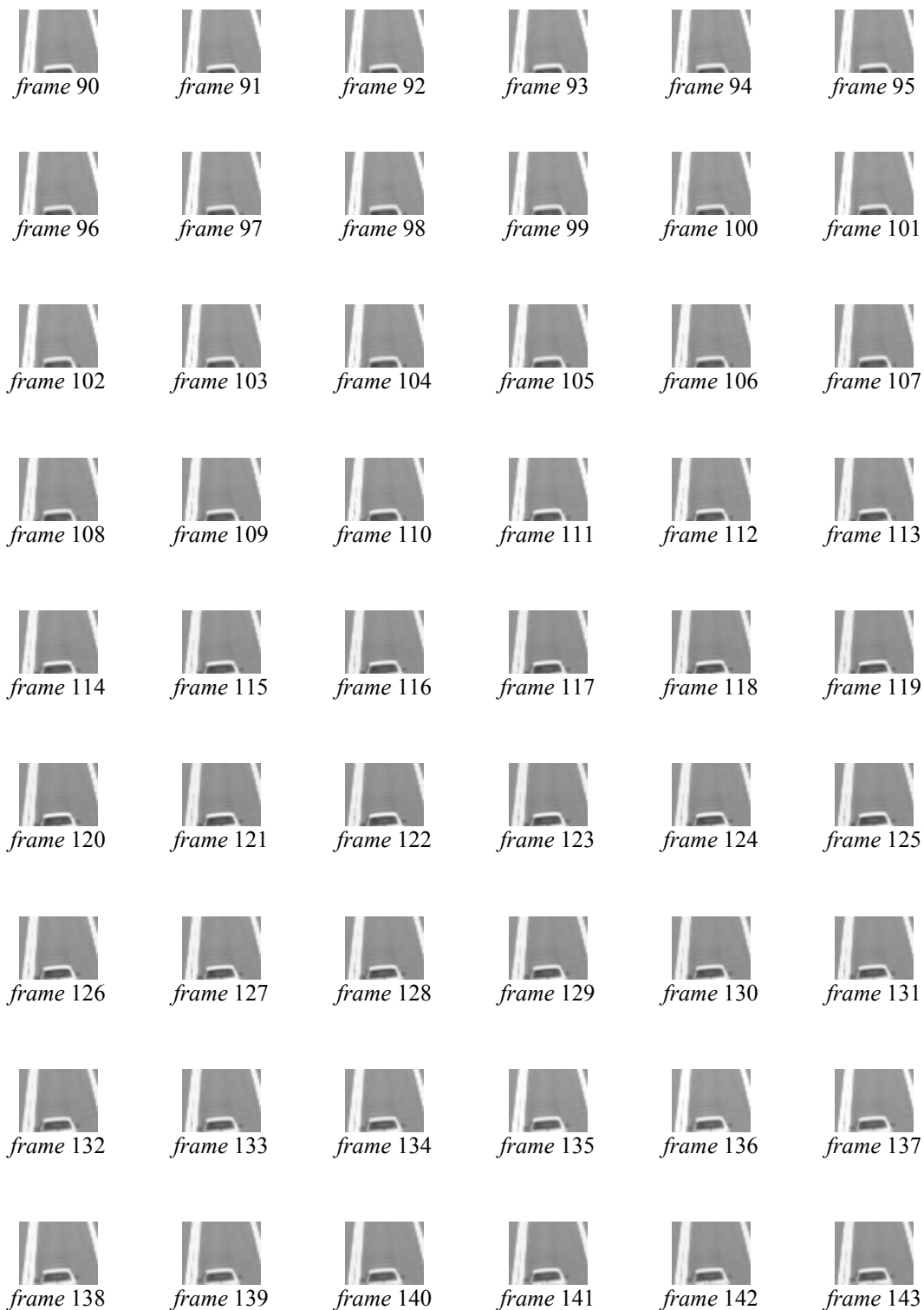


Figura 6.10 – Alguns *frames* da sequência 1 reconstruída a partir da sequência de 512 *frames*, banda de 32 amostras no FPB

6.2 SEQUÊNCIA 1 – RESULTADOS PARA BANDA DE 64 AMOSTRAS

Na reconstrução do sinal observado (1024 amostras) a partir dos sinais de 512 e 256 amostras, foram obtidos os resultados apresentados na Tabela 6.2, nesta tabela apresentam-se apenas os valores: máximo, médio e mínimo dos parâmetros analisados.

Os resultados seguintes, secção 6.2.1 e secção 6.2.2, foram obtidos utilizando uma banda de amostras do FPB igual a 64, isto para que a convergência do algoritmo esteja garantida de acordo com a Tabela 5.2.

O facto de não se apresentarem os resultados para reconstrução a partir da sequência com 128 *frames* está directamente relacionado com o processo de decimação aplicado de modo a garantir a convergência como descrito na Figura 5.7.

Tabela 6.2 – Sequência 1 – Resultados obtidos na reconstrução dos sinais de 256 e 512 *frames* sem compressão (banda de 64 amostras).

Nº de <i>Frames</i> do Sinal a Reconstruir	Banda de Amostras do FPB	Nº de Zeros entre <i>Frames</i>	Erro Final do Algoritmo de Reconstrução		RMSE		PSNR (dB)	
256	64	3	Max	0.0286	Max	1.0209	Max	198.65
			Med	0.0081	Med	0.1177	Med	78.78
			Min	5.0185e ⁻⁶	Min	2.9779e ⁻⁸	Min	47.95
512	64	1	Max	0.0210	Max	0.6946	Max	207.71
			Med	0.0060	Med	0.0801	Med	82.46
			Min	2.6921e ⁻⁶	Min	1.0496e ⁻⁸	Min	51.29

Nos resultados apresentados na Tabela 6.2, não é possível identificar quais as zonas dos *frames* nos quais se obteve melhor desempenho na reconstrução dos sinais; para que isso seja possível posteriormente apresentam-se os resultados de forma gráfica. Nesses gráficos apresentam-se os valores de RMSE e PSNR obtidos na reconstrução dos sinais representativos de cada *pixel* dos *frames* da sequência.

6.2.1 SEQUÊNCIA 1 – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 256 *FRAMES*

Nas Figuras 6.11 e 6.12, estão apresentados os valores do RMSE e PSNR da reconstrução de cada um dos 1620 sinais de 256 amostras, para 1024 amostras,

correspondentes a cada um dos *pixels* dos *frames*. Refira-se que tal como no caso anterior os resultados apresentados estão normalizados.

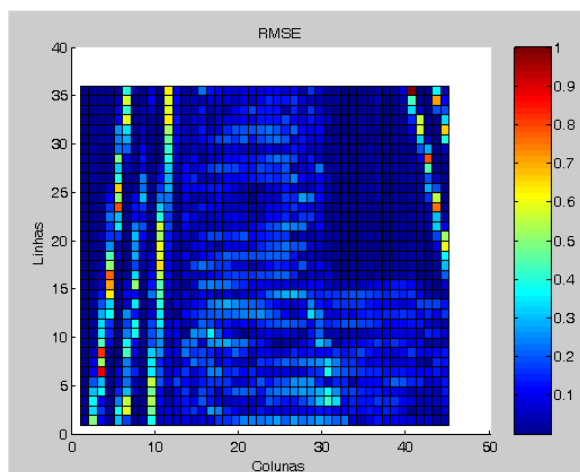


Figura 6.11 – Sequência 1, RMSE, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 256 *frames* (banda de 64 amostras)

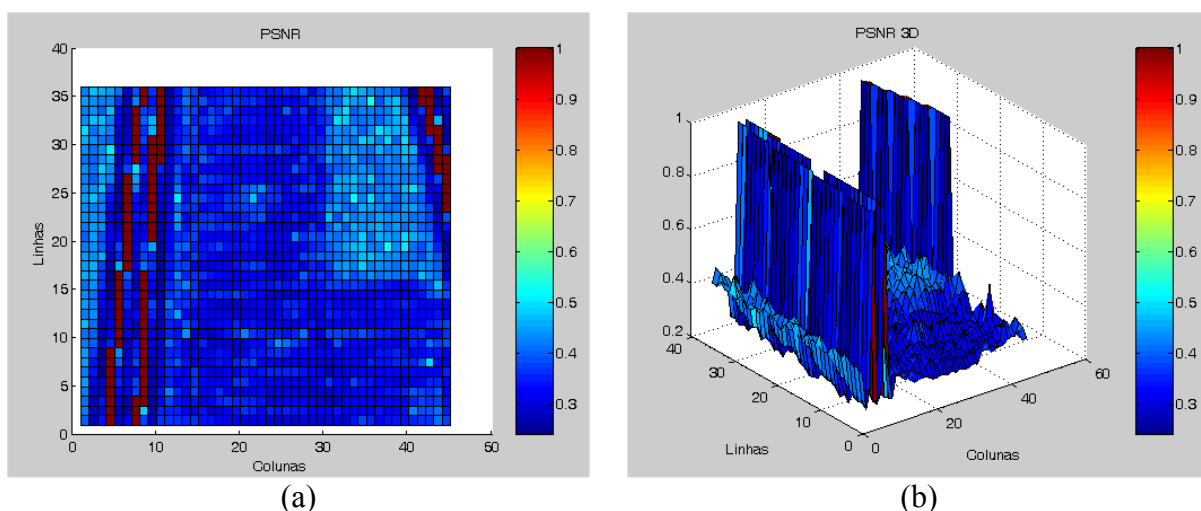


Figura 6.12 – Sequência 1, PSNR, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 256 *frames* (banda de 64 amostras)

Em concordância com os resultados obtidos na secção 6.1.2 para uma LB igual a 32, verifica-se que as zonas onde se obteve melhor reconstrução nos sinais são as zonas laterais dos *frames*, Figura 6.11 e Figura 6.12. A zona central dos *frames* continua a ser a que tem menor relação sinal/ruído e maior RMSE.

Estatisticamente, dos 1620 sinais, 29.26% possuem um PSNR igual ou superior a 78.78 dB (PSNR médio) e 70.74% possuem um PSNR entre 47.95 dB (PSNR mínimo) e 78.78 dB. O PSNR máximo neste caso possui como seria de esperar um valor igual ao correspondente

para o caso da LB de 32 ou seja, 198.65 dB, já que no algoritmo, o valor do erro máximo admissível é sempre o mesmo, 10^{-8} .

6.2.2 SEQUÊNCIA 1 – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 512 FRAMES

Nas Figuras 6.13 e 6.14, estão apresentados os valores do RMSE e PSNR da reconstrução de cada um dos 1620 sinais de 512 amostras, para 1024 amostras, correspondentes a cada um dos *pixels* dos *frames*. Os resultados apresentados estão normalizados.

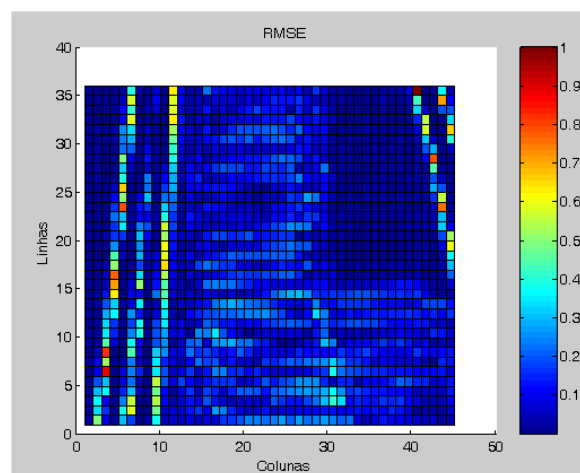


Figura 6.13 – Sequência 1, RMSE, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 512 *frames* (banda de 64 amostras)

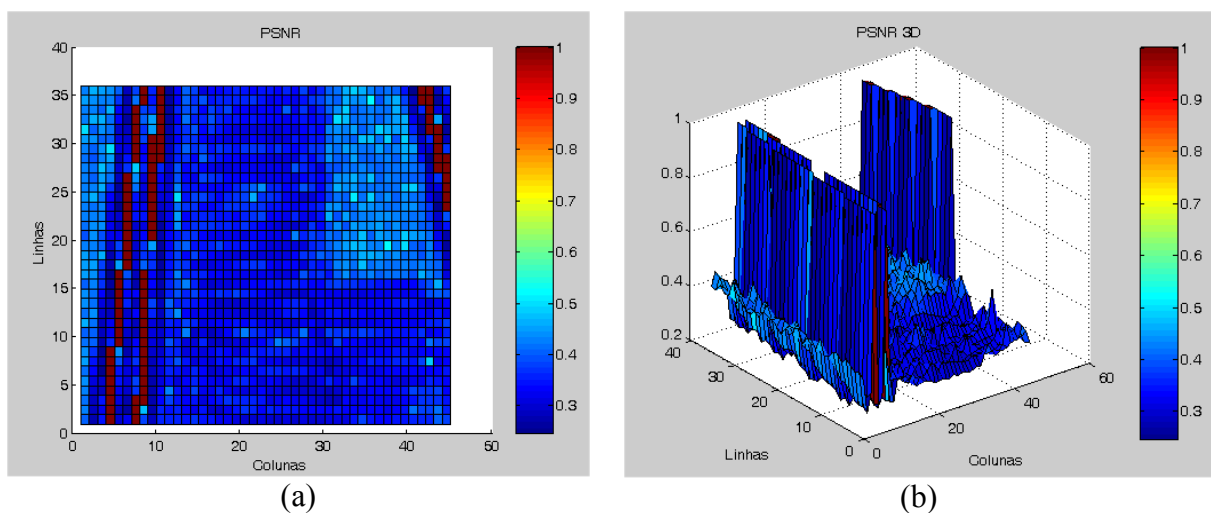


Figura 6.14 – Sequência 1, PSNR, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 512 *frames* (banda de 64 amostras)

Também neste caso se verifica que as zonas onde se obteve melhor reconstrução nos sinais são as zonas laterais dos *frames*, Figura 6.13 e Figura 6.14. Neste caso, 29.01% dos 1620 sinais possuem um PSNR maior ou igual ao valor médio, 82.46 dB e menor ou igual ao PSNR máximo, 207.71 dB. Os restantes 70.99%, são sinais com uma relação sinal/ruído entre 51.29 dB e 82.46 dB.

Como neste caso o número de amostras desconhecidas é inferior, os valores da relação sinal/ruído são mais elevados quando comparados com o caso anterior, atingindo mesmo valores superiores a 200 dB em alguns casos.

6.3 SEQUÊNCIA 1 – RESULTADOS PARA BANDA DE 128 AMOSTRAS

Na reconstrução do sinal observado (1024 amostras) a partir dos sinais de 512 amostras, foram obtidos os resultados apresentados na Tabela 6.3. Nesta tabela apresentam-se apenas os valores máximo, médio e mínimo dos parâmetros analisados.

Os resultados seguintes, secção 6.3.1, foram obtidos utilizando uma banda de amostras do FPB igual a 128, isto para que a convergência do algoritmo esteja garantida de acordo com a Tabela 5.2.

O facto de não se apresentarem os resultados para reconstrução a partir das sequências com 256 e 128 *frames* está directamente relacionado com o processo de decimação aplicado de modo a garantir a convergência como descrito na Figura 5.7.

Tabela 6.3 – Sequência 1 – Resultados obtidos na reconstrução dos sinais de 512 *frames* sem compressão (banda de 128 amostras)

Nº de <i>Frames</i> do Sinal a Reconstruir	Banda de Amostras do FPB	Nº de Zeros entre <i>Frames</i>	Erro Final do Algoritmo de Reconstrução		RMSE		PSNR (dB)	
512	64	1	Max	0.0127	Max	0.2665	Max	207.71
			Med	0.0035	Med	0.0259	Med	90.72
			Min	2.6961e ⁻⁶	Min	1.0496e ⁻⁸	Min	59.62

6.3.1 SEQUÊNCIA 1 – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 512 FRAMES

Nas Figuras 6.15 e 6.16, estão apresentados os valores do RMSE e PSNR para a reconstrução de cada um dos 1620 sinais de 512 amostras, para 1024 amostras, correspondentes a cada um dos *pixels* dos *frames*. Os resultados apresentados estão normalizados.

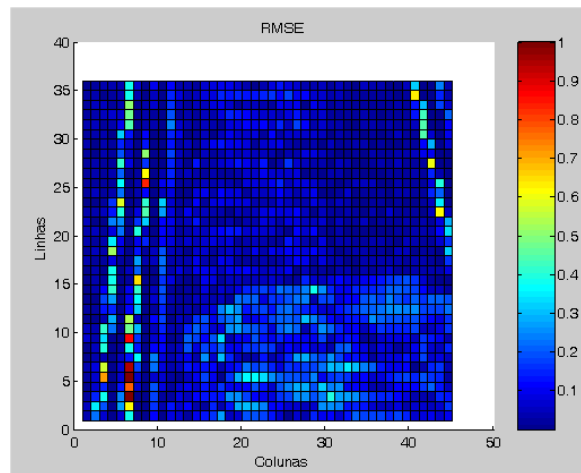


Figura 6.15 – Sequência 1, RMSE, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 512 *frames* (banda de 128 amostras)

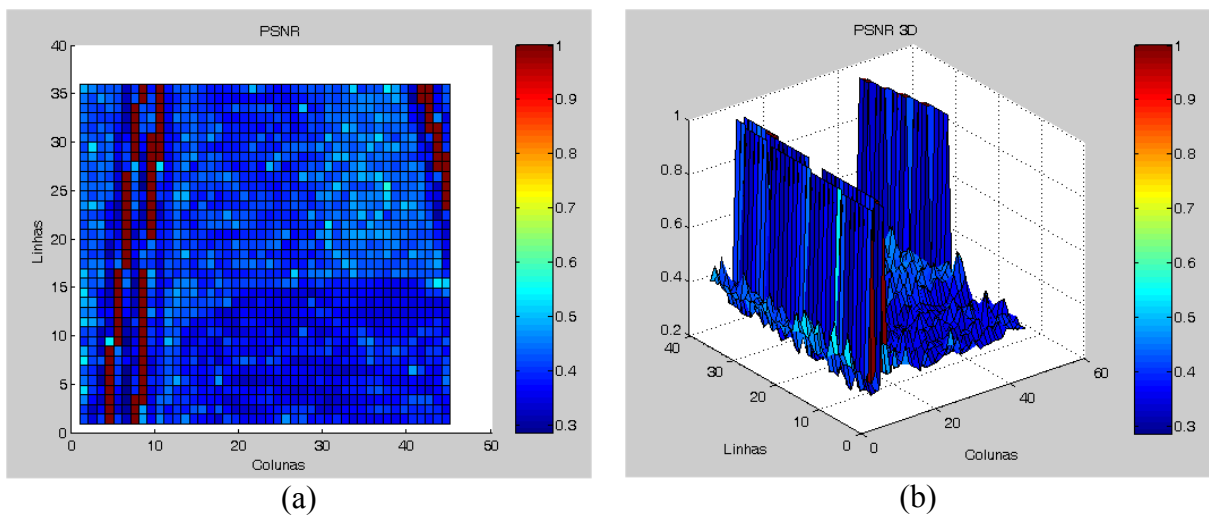


Figura 6.16 – Sequência 1, PSNR, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 512 *frames* (banda de 128 amostras)

Os resultados estatísticos obtidos nesta reconstrução indicam que 24.38% dos sinais possuem um PSNR maior ou igual ao PSNR médio, 90.72 dB, e que 75.62% dos sinais possuem um PSNR maior ou igual a 59.62 dB e inferior ao PSNR médio. Também neste caso as zonas com menor relação sinal/ruído são as zonas centrais dos *frames*.

À semelhança dos resultados apresentados anteriormente, são nas zonas laterais dos *frames* que se encontram os *pixels* cuja reconstrução dos sinais possui a relação sinal/ruído mais elevada, com um PSNR de 207.71 dB. Verifica-se que este valor se mantém igual mesmo para valores de LB diferentes como foi referido no ponto anterior (Tabela 6.1, Tabela 6.2 e Tabela 6.3).

Torna-se evidente que os melhores resultados são obtidos na reconstrução dos 1620 sinais de 512 amostras para 1620 sinais de 1024 amostras, isto para qualquer valor de LB de amostras utilizado uma vez que, é a reconstrução com resultados mais uniformes.

O desempenho do algoritmo relaciona-se com o valor da LB do FPB. Como todo o resto é constante, à medida que o número de componentes não nulas dos sinais aumenta a reconstrução é pior [Ferreira_A, 1994].

Verifica-se um aumento do valor do PSNR médio à medida que a LB aumenta. No entanto, a percentagem de sinais com PSNR superior ao médio diminui apesar do valor mínimo ter aumentado.

6.4 FICHEIROS AVI

Na Tabela 6.4 estão novamente apresentados os tamanhos em bytes dos ficheiros AVI gerados a partir da sequência 1 (Tabela 5.3), a metodologia de geração dos ficheiros AVI está descrita na secção 5.7 e apresentada nas Figuras 5.12 e 5.13. A banda do FPB utilizada é de 32 amostras, isto para que a convergência do algoritmo na reconstrução de sequência de referência, a partir das sequências de 512, 256 e 128 *frames* seja garantida, Tabela 5.2.

Apresenta-se novamente esta tabela para que se possam relacionar os resultados obtidos na reconstrução das sequências que servem de referência (Sequência RCC, Sequência IYUV, Sequência MV1 e Sequência RLE, Figura 5.12), codificadas com os métodos apresentados na secção 5.7, com o espaço necessário para armazenar a totalidade dos *frames* de uma determinada sequência codificada.

Tabela 6.4 – Tabela comparativa dos ficheiros AVI criados utilizando ou não métodos de codificação

Compressão Nº <i>Frames</i>	Sem Compressão	<i>Cinpack</i> <i>Codec by</i> <i>Radius</i>	<i>Intel IYUV</i>	<i>Microsoft</i> <i>Video 1</i>	<i>Run Length</i> <i>Encoding</i>
1024	1757 kB	2056 kB	2403 kB	229 kB	1021 kB
512	881 kB	1029 kB	1203 kB	117 kB	513 kB
256	443 kB	516 kB	603 kB	61 kB	259 kB
128	224 kB	259 kB	303 kB	33 kB	132 kB

Como já foi referido na secção 5.7, a metodologia proposta para codificação/descodificação utilizando ficheiros AVI (Figura 5.13), apenas foi testada na sequência 1.

Na secção 6.4.1 até à secção 6.4.4, apresentam-se os resultados da reconstrução das sequências de referência a partir das sequências de 512, 256 e 128 *frames*, codificadas/descodificadas. Os resultados apresentados nas figuras da secção 6.4.1 à secção 6.4.4 estão normalizados.

6.4.1 SEQUÊNCIA 1 CODIFICADA COM O MÉTODO RCC – RESULTADOS PARA BANDA DE 32 AMOSTRAS

Esta sequência de referência, criada com o método apresentado na Figura 5.11, possui 1620 (45×36) sinais cada um com 1024 amostras e foi codificada/descodificada com o método RCC.

Na reconstrução do sinal de referência (1024 amostras) a partir dos sinais, codificados com o RCC, de 512, 256 e 128 amostras, foram obtidos os resultados apresentados na Tabela 6.6, nesta tabela apresentam-se apenas os valores: máximo, médio e mínimo dos parâmetros analisados.

Tabela 6.5 – Sequência 1 codificada com o RCC, Resultados da reconstrução dos sinais de 128, 256 e 512 *frames* (banda de 32 amostras)

Nº de <i>Frames</i> do Sinal a Reconstruir	Banda de Amostras do FPB	Nº de Zeros entre <i>Frames</i>	Erro Final do Algoritmo de Reconstrução		RMSE		PSNR (dB)	
128	32	7	Max	0.0097	Max	0.1120	Max	101.42
			Med	0.0049	Med	0.0312	Med	79.62
			Min	0.0013	Min	0.0022	Min	67.14
256	32	3	Max	0.0084	Max	0.0933	Max	103.01
			Med	0.0043	Med	0.0260	Med	81.21
			Min	0.0012	Min	0.0018	Min	68.73
512	32	1	Max	0.0063	Max	0.0635	Max	106.35
			Med	0.0032	Med	0.0177	Med	84.55
			Min	8.7248e ⁻⁴	Min	0.0012	Min	72.07

Verifica-se que em qualquer das versões se consegue reconstruir o sinal de referência. Os resultados obtidos indicam que a utilização da sequência de 128 *frames* para efeitos de representação é viável. Pode ver-se também que a reconstrução obtida com a sequência de 512 *frames* apresenta melhores resultados, não sendo no entanto significativamente superiores aos obtidos com a sequência de 128 *frames*, como pode verificar-se nos valores apresentados para o PNSR na Tabela 6.5.

Os resultados aqui discutidos podem ser observados com maior detalhe na secção 6.4.1.1 à secção 6.4.1.3, estes foram obtidos utilizando uma banda de amostras do FPB igual a 32, para que a convergência do algoritmo esteja garantida.

6.4.1.1 SEQUÊNCIA 1 CODIFICADA COM O RCC – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 128 FRAMES

Nas Figuras 6.17 e 6.18, estão apresentados os valores do RMSE e PSNR obtidos na reconstrução de cada um dos 1620 sinais de 128 amostras, para 1024 amostras, correspondentes aos sinais de cada um dos *pixels* dos *frames*, que foram codificados com o método RCC.

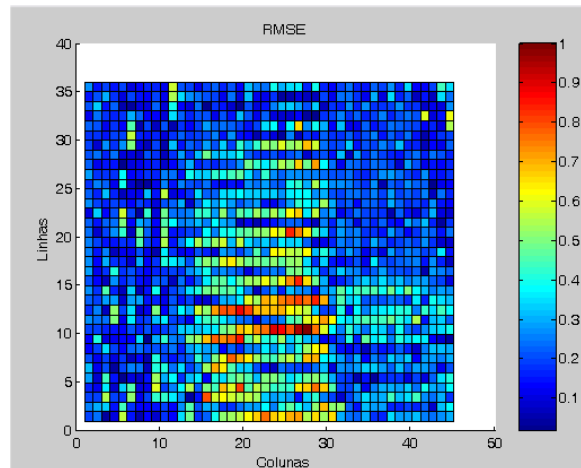


Figura 6.17 – Sequência 1 codificada com o RCC, RMSE, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 128 *frames* (banda de 32 amostras)

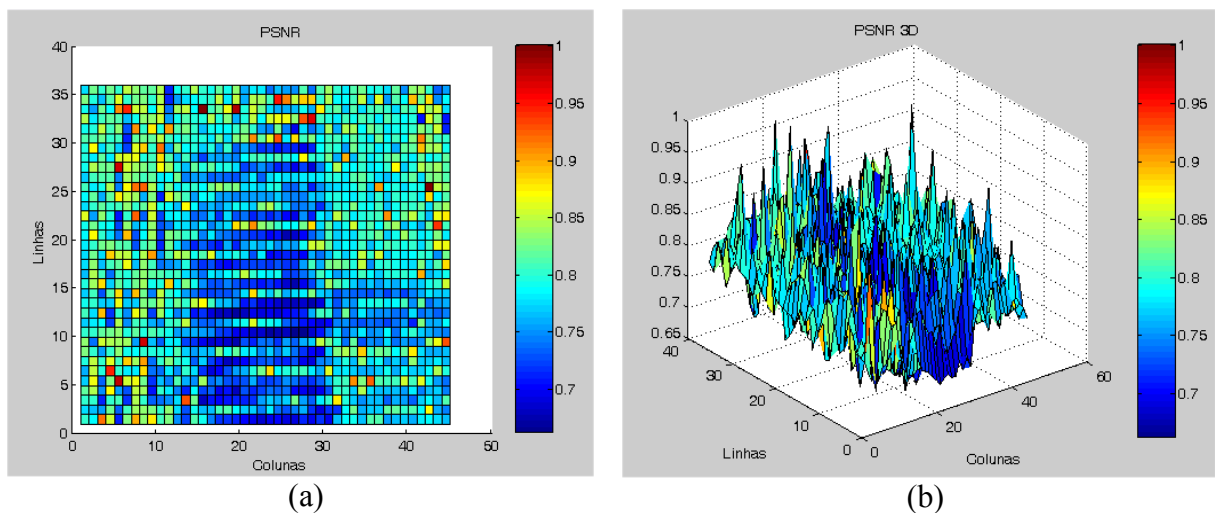


Figura 6.18 – Sequência 1 codificada com o RCC, PSNR, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 128 *frames* (banda de 32 amostras)

Da análise das Figuras 6.17 e 6.18, conclui-se que na reconstrução dos 1620 sinais observados de 1024 amostras a partir dos sinais de 128 amostras, as zonas dos *frames* da sequência, em que se conseguiu obter uma melhor reconstrução foram as zonas laterais dos *frames*. Nessas zonas, os sinais reconstruídos apresentam um PSNR superior a 79.62 dB

(PSNR médio), e conseqüentemente um RMSE mais baixo. Após análise estatística dos resultados, verificou-se que dos 1620 sinais reconstruídos, 50.12% possuem um PSNR igual ou superior ao PSNR médio. Por outro lado, 49.88% dos sinais apresentam um PSNR superior ao PSNR mínimo (67.14 dB) e inferior ao PSNR médio. Estes sinais são maioritariamente pertencentes à zona central dos *frames* da seqüência, que são as zonas com um RMSE mais elevado. Os valores, máximo e mínimo de PSNR, obtidos nesta reconstrução são respectivamente, 101.42 dB e 67.14 dB.

6.4.1.2 SEQUÊNCIA 1 CODIFICADA COM O RCC – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 256 *FRAMES*

Nas Figuras 6.19 e 6.20, estão apresentados os valores do RMSE e PSNR obtidos na reconstrução de cada um dos 1620 sinais de 256 amostras, para 1024 amostras, correspondentes aos sinais de cada um dos *pixels* dos *frames*, que foram codificados com o método RCC.

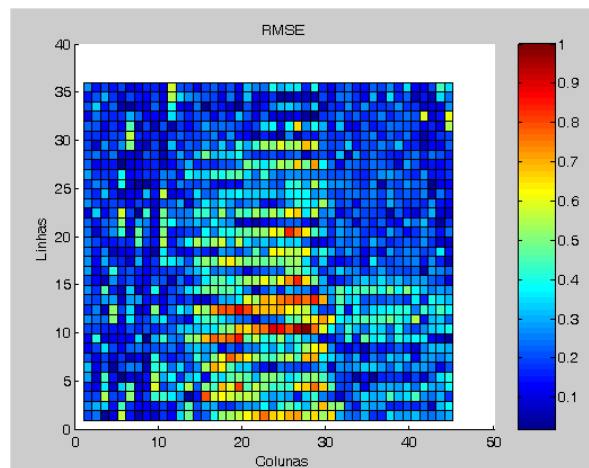


Figura 6.19 – Sequência 1 codificada com o RCC, RMSE, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da seqüência de 256 *frames* (banda de 32 amostras)

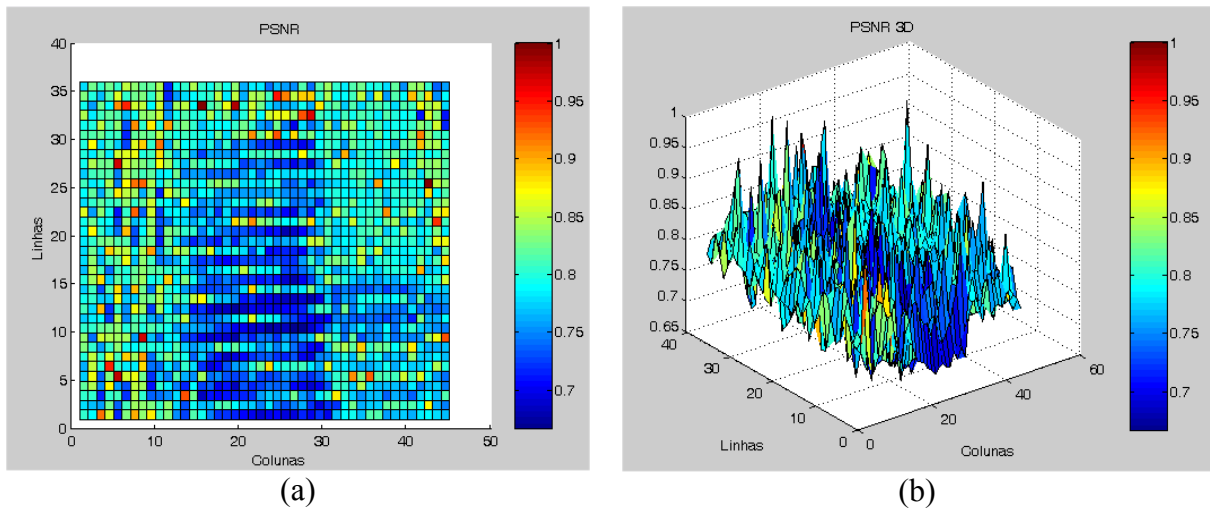


Figura 6.20 – Sequência 1 codificada com o RCC, PSNR, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 256 *frames* (banda de 32 amostras)

Analisando os resultados apresentados nas Figuras 6.19 e 6.20, é possível verificar que as zonas onde se obteve melhor reconstrução dos sinais são as zonas laterais dos *frames*. A zona central dos *frames* continua a ser a zona com menor PSNR e maior RMSE. Estatisticamente, dos 1620 sinais, 50.12% possuem um PSNR igual ou superior a 81.21 dB (valor médio) e 49.88% possuem um PSNR entre 68.73 dB (valor mínimo) e o valor médio. De referir que o valor máximo, médio e mínimo de PSNR desta reconstrução são superiores aos valores obtidos na reconstrução dos sinais de 128 amostras. O valor máximo de PSNR é agora 103.01 dB. Em relação aos valores obtidos na reconstrução anterior, o PSNR (máximo, médio e mínimo) aumentou aproximadamente 2 dB.

6.4.1.3 SEQUÊNCIA 1 CODIFICADA COM O RCC – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 512 *FRAMES*

Nas Figuras 6.21 e 6.22, estão apresentados os valores do RMSE e PSNR obtidos na reconstrução de cada um dos 1620 sinais de 512 amostras, para 1024 amostras, correspondentes aos sinais de cada um dos *pixels* dos *frames*, que foram codificados com o método RCC.

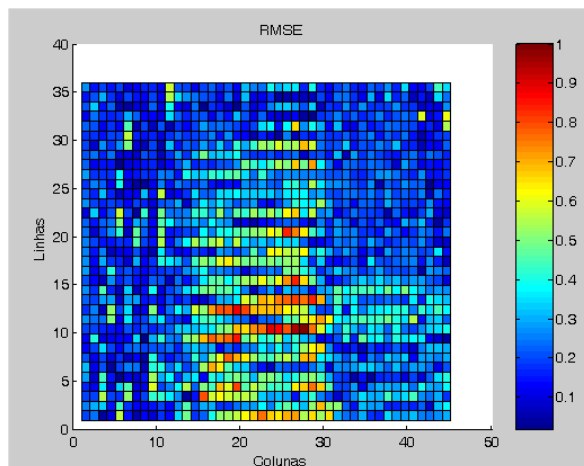


Figura 6.21 – Sequência 1 codificada com o RCC, *RMSE*, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 512 *frames* (banda de 32 amostras)

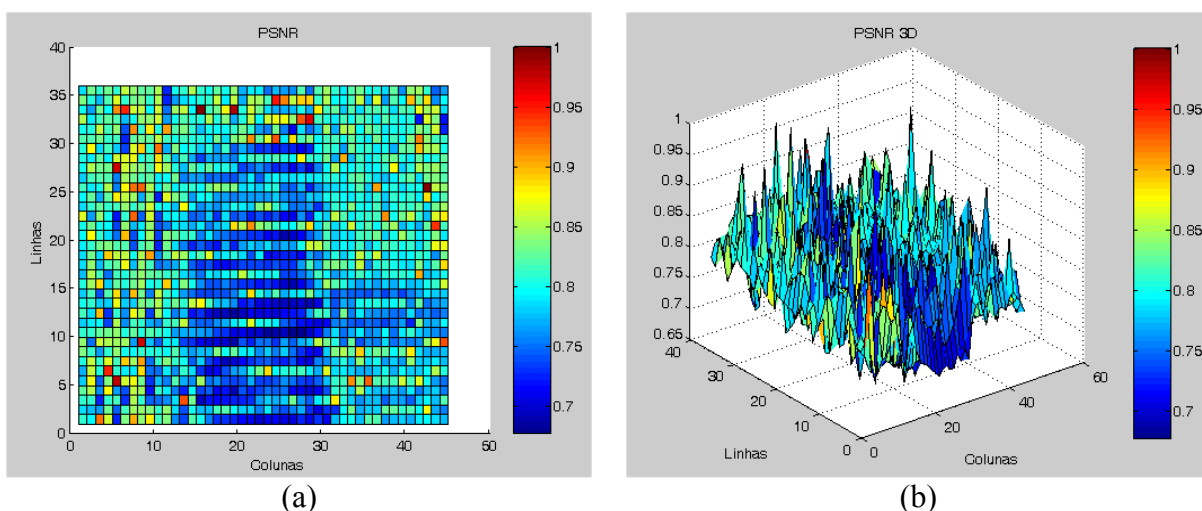


Figura 6.22 – Sequência 1 codificada com o RCC, *PSNR*, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 512 *frames* (banda de 32 amostras)

Da análise da Figuras 6.21, verifica-se que as zonas com maior *RMSE* são igualmente as zonas centrais dos *frames* da sequência, os *pixels* desta zona apresentam um *PSNR* mais baixo, Figura 6.22. Na reconstrução da sequência com 512 *frames*, 50.12% dos 1620 sinais possuem um *PSNR* maior ou igual ao valor médio, 84.55 dB e 49.88% apresenta um valor maior ou igual ao *PSNR* mínimo e inferior ao *PSNR* médio. Os sinais com um *PSNR* aproximadamente igual ao valor máximo, representam maioritariamente sinais de *pixels* das zonas laterais dos *frames* (zonas a vermelho e laranja na Figura 6.22), no entanto, existem sinais na zona central dos *frames* com *PSNR* elevado.

Em comparação com os resultados obtidos nas reconstruções anteriores, esta é a que apresenta valores mais elevados de *PSNR*; os valores: máximo, médio e mínimo aumentaram aproximadamente 2 dB, em relação à reconstrução dos 1620 sinais de 256 amostras.

Verifica-se que a percentagem de sinais com valor maior ou igual aos PSNR médios é igual em cada reconstrução, assim como a percentagem com PSNR inferior aos valores médios.

É igualmente importante referir que todas as sequências reconstruídas (128, 256 e 512 *frames*) são perceptivelmente idênticas à sequência de referência (sequência observada codificada com o método RCC).

6.4.2 SEQUÊNCIA 1 CODIFICADA COM O MÉTODO IYUV – RESULTADOS PARA BANDA DE 32 AMOSTRAS

Esta sequência de referência, possui 1620 (45×36) sinais cada um com 1024 amostras e foi codificada/descodificada com o método IYUV.

Na reconstrução do sinal de referência (1024 amostras) a partir dos sinais, codificados com o IYUV, de 512, 256 e 128 amostras, foram obtidos os resultados apresentados na Tabela 6.6, nesta tabela apresentam-se apenas os valores: máximo, médio e mínimo dos parâmetros analisados.

Tabela 6.6 – Sequência 1 codificada com o IYUV, Resultados da reconstrução dos sinais de 128, 256 e 512 *frames* (banda de 32 amostras)

Nº de <i>Frames</i> do Sinal a Reconstruir	Banda de Amostras do FPB	Nº de Zeros entre <i>Frames</i>	Erro Final do Algoritmo de Reconstrução		RMSE		PSNR (dB)	
128	32	7	Max	0.0144	Max	0.2488	Max	114.40
			Med	0.0088	Med	0.0989	Med	69.07
			Min	6.3017e ⁻⁴	Min	4.8566e ⁻⁴	Min	60.21
256	32	3	Max	0.0125	Max	0.2073	Max	116.00
			Med	0.0077	Med	0.0824	Med	70.66
			Min	5.5143e ⁻⁴	Min	4.0404e ⁻⁴	Min	61.79
512	32	1	Max	0.0093	Max	0.1410	Max	119.35
			Med	0.0058	Med	0.0561	Med	74.00
			Min	4.1774e ⁻⁴	Min	2.7492e ⁻⁴	Min	65.14

Os resultados obtidos neste caso são muito idênticos aos obtidos com o método apresentado anteriormente, dado que a aplicação destes métodos não introduzem degradação perceptível na sequência. Optou-se por apresentar também os resultados para este método para corroborar os resultados obtidos com o método anterior.

Os resultados podem ser observados com maior detalhe na secção 6.4.2.1 à secção 6.4.2.3, estes foram também obtidos utilizando uma banda de amostras do FPB igual a 32, para que a convergência do algoritmo esteja garantida.

6.4.2.1 SEQUÊNCIA 1 CODIFICADA COM O IYUV – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 128 FRAMES

Nas Figuras 6.23 e 6.24, estão apresentados os valores do RMSE e PSNR obtidos na reconstrução de cada um dos 1620 sinais de 128 amostras, para 1024 amostras, correspondentes aos sinais de cada um dos *pixels* dos *frames*, que foram codificados com o método IYUV.

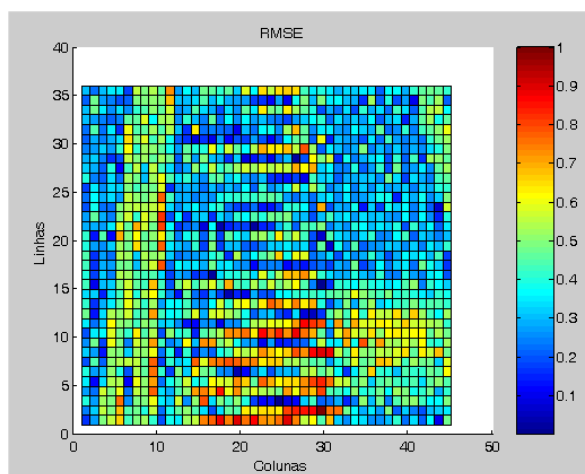


Figura 6.23 – Sequência 1 codificada com o IYUV, RMSE, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 128 *frames* (banda de 32 amostras)

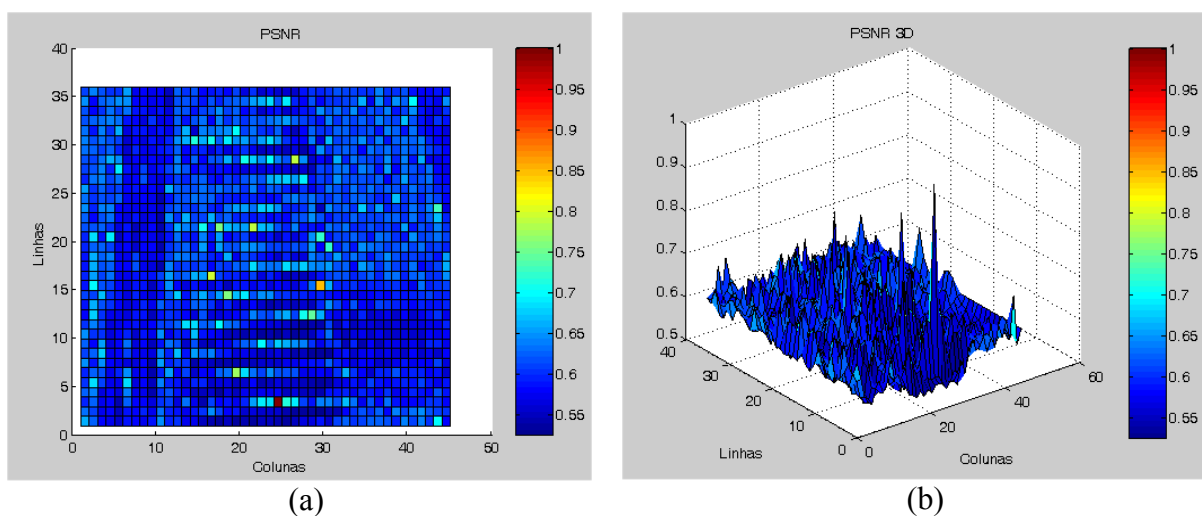


Figura 6.24 – Sequência 1 codificada com o IYUV, PSNR, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 128 *frames* (banda de 32 amostras)

Da análise das Figuras 6.23 e 6.24, concluí-se que na reconstrução dos 1620 sinais observados, de 1024 amostras a partir dos sinais de 128 amostras, as zonas dos *frames* da sequência em que se conseguiu obter uma melhor reconstrução foram as zonas laterais dos *frames*. Nessas zonas, os sinais reconstruídos apresentam um PSNR superior a 69.07 dB (valor médio), e conseqüentemente um RMSE mais baixo. Após análise estatística dos resultados, verificou-se que dos 1620 sinais reconstruídos, 45.99% possuem um PSNR igual ou superior ao PSNR médio. Por outro lado, 54.01% dos sinais apresentam um PSNR superior ao PSNR mínimo e inferior ao PSNR médio. Estes sinais são maioritariamente pertencentes à zona central dos *frames* da sequência, que são as zonas com um RMSE mais elevado. Os valores: máximo e mínimo de PSNR obtidos nesta reconstrução, são respectivamente, 114.40 dB e 60.21 dB.

6.4.2.2 SEQUÊNCIA 1 CODIFICADA COM O IYUV – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 256 FRAMES

Nas Figuras 6.25 e 6.26, estão apresentados os valores do RMSE e PSNR obtidos na reconstrução de cada um dos 1620 sinais de 256 amostras, para 1024 amostras, correspondentes aos sinais de cada um dos *pixels* dos *frames*, que foram codificados com o método IYUV.

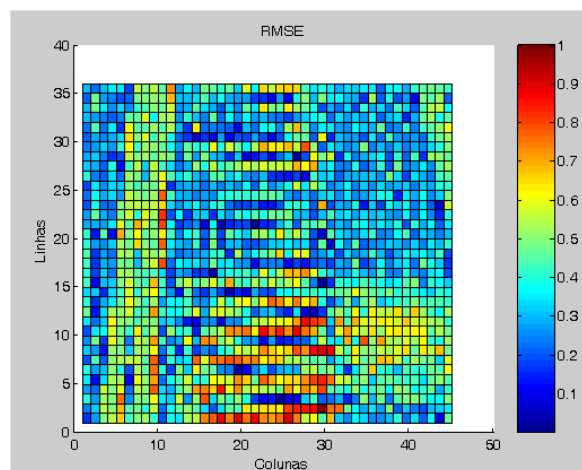


Figura 6.25 – Sequência 1 codificada com o IYUV, RMSE, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 256 *frames* (banda de 32 amostras)

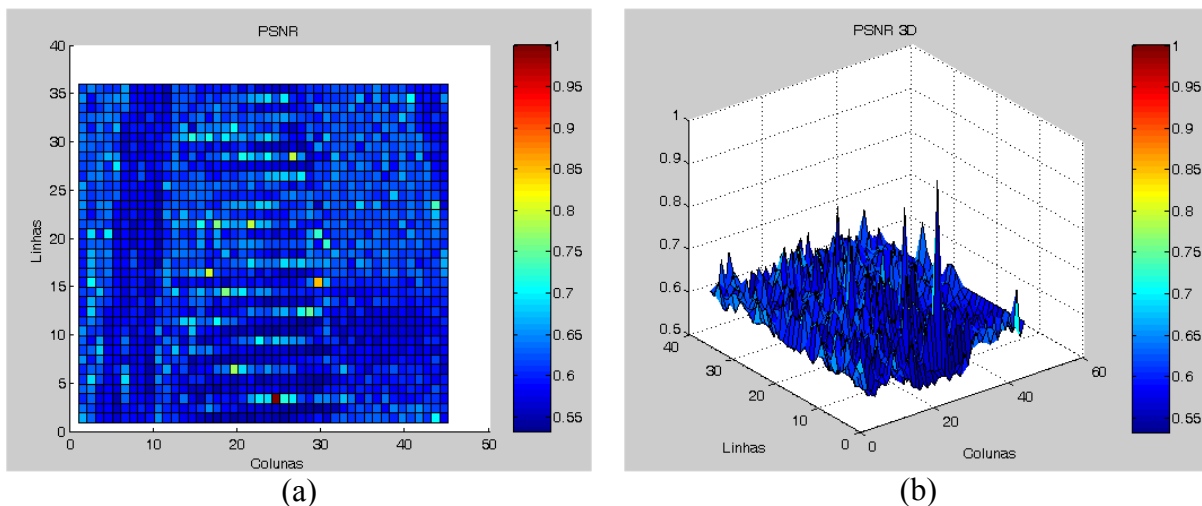


Figura 6.26 – Sequência 1 codificada com o IYUV, PSNR, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 256 *frames* (banda de 32 amostras)

Analisando os resultados apresentados nas Figuras 6.25 e 6.26, verifica-se que as zonas onde se obteve melhor reconstrução dos sinais são as zonas laterais dos *frames*. A zona central dos *frames* continua a ser a zona com menor relação sinal/ruído e maior RMSE. Estatisticamente, dos 1620 sinais, 45.99% possuem um PSNR igual ou superior a 70.66 dB (valor médio) e 54.01% possuem um PSNR entre 61.79 dB (valor mínimo) e o valor médio. Refira-se também que o valor máximo, médio e mínimo de PSNR desta reconstrução são superiores aos valores obtidos na reconstrução dos sinais de 128 amostras. O valor máximo de PSNR é agora 116.00 dB. Em relação aos valores obtidos na reconstrução anterior o PSNR (máximo, médio e mínimo) aumentou aproximadamente 2 dB.

6.4.2.3 SEQUÊNCIA 1 CODIFICADA COM O IYUV – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 512 *FRAMES*

Nas Figuras 6.27 e 6.28, estão apresentados os valores do RMSE e PSNR obtidos na reconstrução de cada um dos 1620 sinais de 512 amostras, para 1024 amostras, correspondentes aos sinais de cada um dos *pixels* dos *frames*, que foram codificados com o método IYUV.

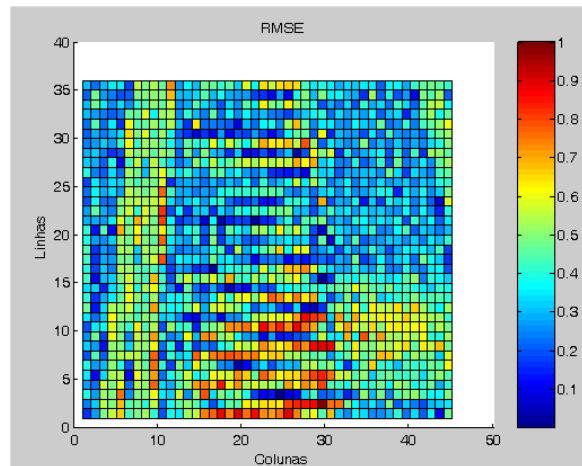


Figura 6.27 – Sequência 1 codificada com o IYUV, *RMSE*, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 512 *frames* (banda de 32 amostras)

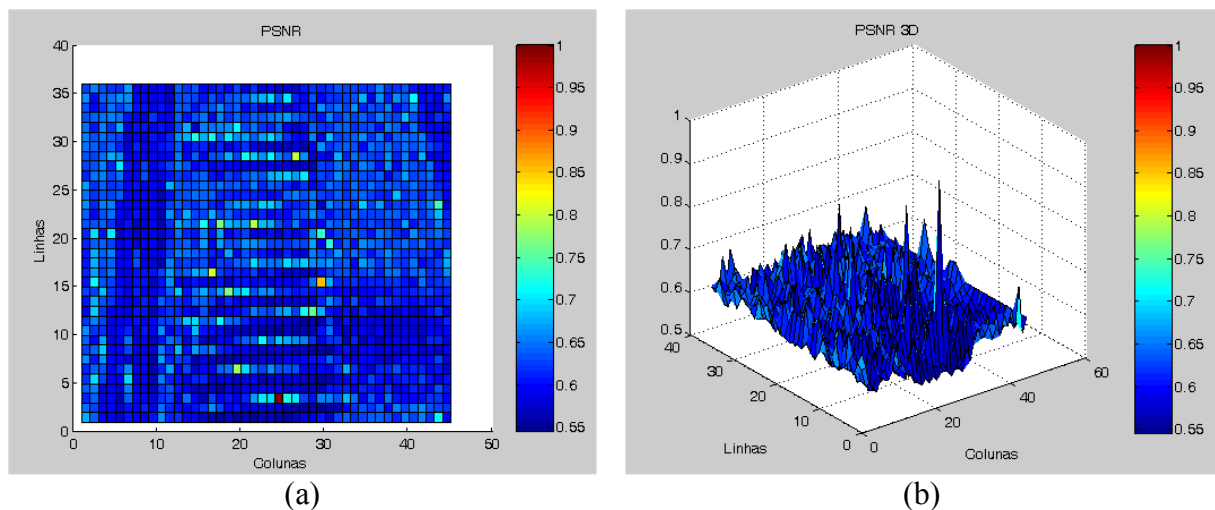


Figura 6.28 – Sequência 1 codificada com o IYUV, *PSNR*, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 512 *frames* (banda de 32 amostras)

Da análise da Figura 6.27, verifica-se que as zonas com maior *RMSE* são igualmente as zonas centrais dos *frames* da sequência, os *pixels* desta zona apresentam uma relação sinal/ruído mais baixa, Figura 6.28. Na reconstrução da sequência com 512 *frames*, 45.99% dos 1620 sinais possuem um *PSNR* maior ou igual ao valor médio, 74.00 dB e 54.01% apresenta um valor maior ou igual ao *PSNR* mínimo, 65.14 dB, e inferior ao valor médio. Analisando o resultado desta reconstrução conclui-se que a maior percentagem dos sinais apresenta um *PSNR* inferior ao *PSNR* médio. Estes sinais representam maioritariamente *pixels* da zona central dos *frames*.

Em comparação com os resultados obtidos nas reconstruções anteriores, esta é a que apresenta valores mais elevados na relação sinal/ruído.

É igualmente importante notar que todas as sequências reconstruídas (128, 256 e 512 *frames*) são perceptivelmente idênticas à sequência de referência (sequência observada codificada com o método IYUV).

6.4.3 SEQUÊNCIA 1 CODIFICADA COM O MÉTODO MV1 – RESULTADOS PARA BANDA DE 32 AMOSTRAS

Esta sequência de referência, criada com o método apresentado na Figura 5.11, possui 1620 (45×36) sinais cada um com 1024 amostras e foi codificada/descodificada com o método MV1.

Na reconstrução do sinal de referência (1024 amostras) a partir dos sinais, codificados com o MV1, de 512, 256 e 128 amostras, foram obtidos os resultados apresentados na Tabela 6.7, nesta tabela apresentam-se apenas os valores: máximo, médio e mínimo dos parâmetros analisados.

Tabela 6.7 – Sequência 1 codificada com o MV1, Resultados da reconstrução dos sinais de 128, 256 e 512 *frames* (banda de 32 amostras)

Nº de <i>Frames</i> do Sinal a Reconstruir	Banda de Amostras do FPB	Nº de Zeros entre <i>Frames</i>	Erro Final do Algoritmo de Reconstrução		RMSE		PSNR (dB)	
128	32	7	Max	0.0092	Max	0.1043	Max	92.52
			Med	0.0064	Med	0.0510	Med	74.36
			Min	0.0022	Min	0.0060	Min	67.77
256	32	3	Max	0.0081	Max	0.0869	Max	94.10
			Med	0.0056	Med	0.0425	Med	75.94
			Min	0.0020	Min	0.0050	Min	69.35
512	32	1	Max	0.0060	Max	0.0591	Max	97.44
			Med	0.0042	Med	0.0289	Med	79.29
			Min	0.0015	Min	0.0034	Min	72.69

Refira-se que este método introduz uma elevada degradação perceptível; o objectivo neste caso, é testar o desempenho do algoritmo neste tipo de sinais.

Verificou-se, no entanto, que em qualquer das versões se consegue também reconstruir o sinal de referência. Os resultados obtidos indicam que a utilização da sequência de 128 *frames* para efeitos de representação é também viável nestes casos. Pode ver-se também que a reconstrução obtida com a sequência de 512 *frames* apresenta melhores resultados, não sendo

no entanto significativamente superiores aos obtidos com a sequência de 128 *frames*, como pode verificar-se nos valores apresentados para o PNSR na Tabela 6.7.

Deste modo constatou-se que o desempenho do algoritmo é muito semelhante em sinais sujeitos a métodos de compressão com diferentes níveis de degradação perceptível.

Estes resultados podem ser observados com maior detalhe na secção 6.4.3.1 à secção 6.4.3.3, estes foram obtidos utilizando uma banda de amostras do FPB igual a 32, para que a convergência do algoritmo esteja garantida.

6.4.3.1 SEQUÊNCIA 1 CODIFICADA COM O MV1 – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 128 *FRAMES*

Nas Figuras 6.29 e 6.30, estão apresentados os valores do RMSE e PSNR obtidos na reconstrução de cada um dos 1620 sinais de 128 amostras, para 1024 amostras, correspondentes aos sinais de cada um dos *pixels* dos *frames*, que foram codificados com o método MV1.

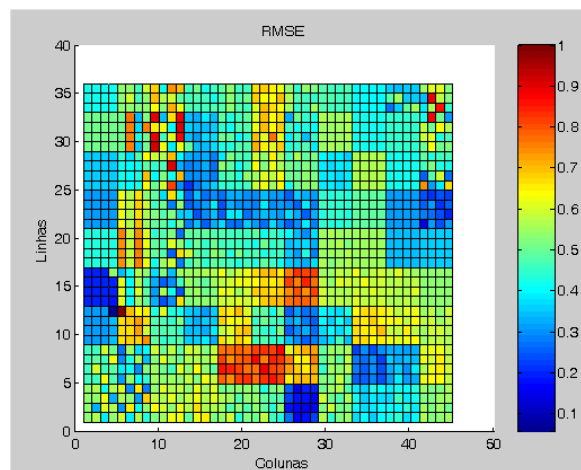


Figura 6.29 – Sequência 1 codificada com o MV1, RMSE, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 128 *frames* (banda de 32 amostras)

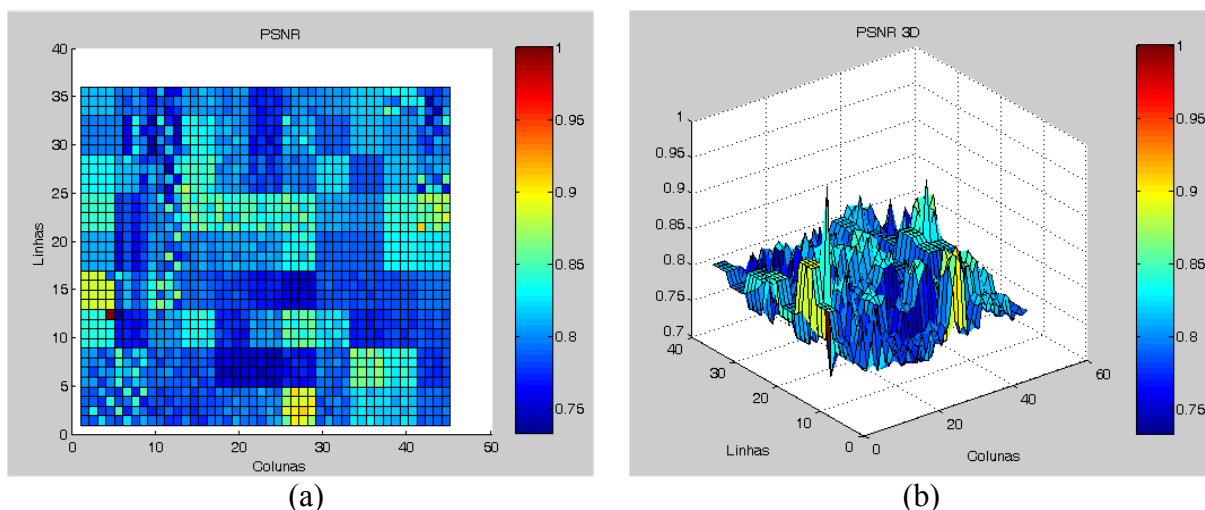


Figura 6.30 – Sequência 1 codificada com o MV1, PSNR, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 128 *frames* (banda de 32 amostras)

Da análise das Figuras 6.29 e 6.30, conclui-se que na reconstrução dos 1620 sinais observados, de 1024 amostras, a partir dos sinais de 128 amostras, as zonas dos *frames* da sequência em que se conseguiu obter uma melhor reconstrução foram maioritariamente as zonas laterais dos *frames*. Nessas zonas, os sinais reconstruídos apresentam um RMSE mais baixo, pelo contrário, as zonas onde o RMSE é mais alto (zonas a vermelho na Figura 6.29) estão maioritariamente na zona central dos *frames*. Estatisticamente 44.63% dos sinais apresentam um PSNR superior a 74.36 dB (valor médio), e conseqüentemente um RMSE mais baixo. Por outro lado, 55.37% dos sinais apresentam uma relação sinal/ruído inferior 74.36 dB. Estes sinais são maioritariamente pertencentes à zona central dos *frames* da sequência, que são as zonas com um RMSE mais elevado. Os valores, máximo e mínimo de PSNR, obtidos nesta reconstrução são respectivamente, 92.52 dB e 67.77 dB.

6.4.3.2 SEQUÊNCIA 1 CODIFICADA COM O MV1 – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 256 *FRAMES*

Nas Figuras 6.31 e 6.32, estão apresentados os valores do RMSE e PSNR obtidos na reconstrução de cada um dos 1620 sinais de 256 amostras, para 1024 amostras, correspondentes aos sinais de cada um dos *pixels* dos *frames*, que foram codificados com o método MV1.

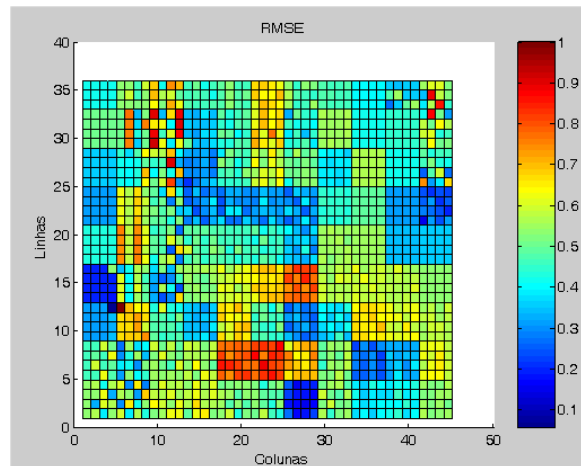


Figura 6.31 – Sequência 1 codificada com o MV1, RMSE, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 256 *frames* (banda de 32 amostras)

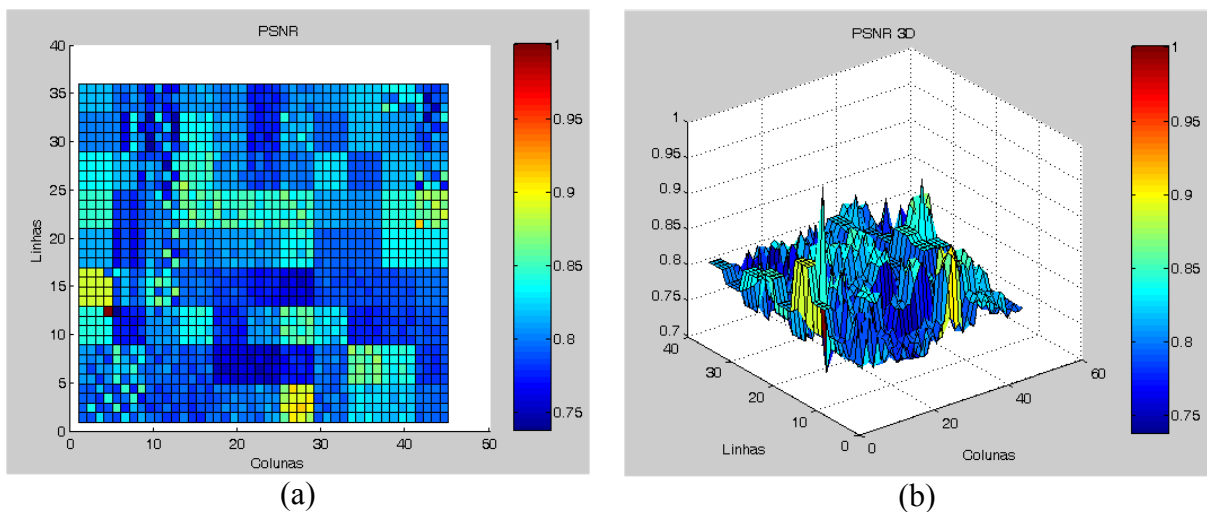


Figura 6.32 – Sequência 1 codificada com o MV1, PSNR, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 256 *frames* (banda de 32 amostras)

Analisando os resultados apresentados nas Figuras 6.31 e 6.32, verifica-se, à semelhança do que aconteceu na reconstrução anterior, que as zonas onde se obteve melhor reconstrução dos sinais, são as zonas laterais dos *frames*. A zona central dos *frames* continua a ser a zona onde se encontram a maioria dos sinais com maior RMSE (zonas as vermelho na Figura 6.31). São estes os sinais que possuem um PSNR mais baixo. Estatisticamente, dos 1620 sinais, 44.63% possuem um PSNR igual ou superior a 75.94 dB (valor médio) e 55.37% possuem um PSNR entre 69.35 dB (valor mínimo) e o valor médio. De referir que o valor máximo, médio e mínimo de PSNR desta reconstrução são superiores aos valores obtidos na reconstrução dos sinais de 128 amostras. O valor máximo de PSNR é agora 94.10 dB. Em relação aos valores obtidos na reconstrução anterior o PSNR (máximo, médio e mínimo) aumentou aproximadamente 2 dB.

6.4.3.3 SEQUÊNCIA 1 CODIFICADA COM O MV1 – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 512 *FRAMES*

Nas Figuras 6.33 e 6.34, estão apresentados os valores do RMSE e PSNR obtidos na reconstrução de cada um dos 1620 sinais de 512 amostras, para 1024 amostras, correspondentes aos sinais de cada um dos *pixels* dos *frames*, que foram codificados com o método MV1.

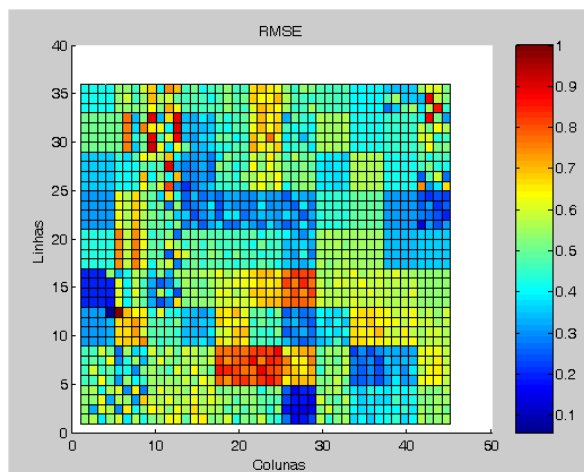


Figura 6.33 – Sequência 1 codificada com o MV1, *RMSE*, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 512 *frames* (banda de 32 amostras)

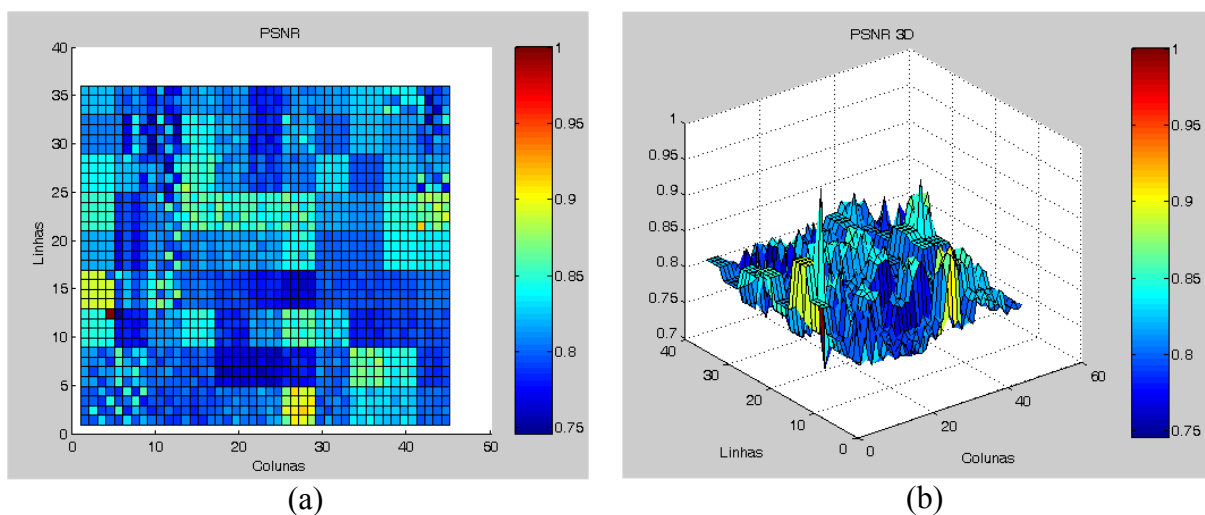


Figura 6.34 – Sequência 1 codificada com o MV1, *PSNR*, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 512 *frames* (banda de 32 amostras)

Analisando a Figura 6.33, verifica-se que as zonas com maior RMSE são igualmente as zonas centrais dos *frames* da sequência, os *pixels* desta zona apresentam igualmente um PSNR mais baixo (Figura 6.34). Na reconstrução da sequência com 512 *frames*, 44.63% dos 1620 sinais possuem um PSNR maior ou igual ao valor médio, 79.29 dB e 55.37% apresenta

um valor maior ou igual ao PSNR mínimo, 72.69 dB, e inferior ao valor médio, estes sinais representam maioritariamente *pixels* da zona central dos *frames*.

Em comparação com os resultados obtidos nas reconstruções anteriores, esta é a que apresenta valores mais elevados de PSNR, os valores máximo, médio e mínimo aumentaram aproximadamente 2 dB, em relação à reconstrução dos 1620 sinais de 256 amostras.

É igualmente importante referir que todas as sequências reconstruídas (128, 256 e 512 *frames*) são perceptivelmente idênticas à sequência de referência (sequência observada codificada com o método MV1).

6.4.4 SEQUÊNCIA 1 CODIFICADA COM O MÉTODO RLE – RESULTADOS PARA BANDA DE 32 AMOSTRAS

Esta sequência de referência, criada com o método apresentado na Figura 5.11, possui 1620 (45×36) sinais cada um com 1024 amostras e foi codificada/descodificada com o método RLE.

Na reconstrução do sinal de referência (1024 amostras) a partir dos sinais, codificados com o RLE, de 512, 256 e 128 amostras, foram obtidos os resultados apresentados na Tabela 6.8, nesta tabela apresentam-se apenas os valores: máximo, médio e mínimo dos parâmetros analisados.

Tabela 6.8 – Sequência 1 codificada com o RLE, Resultados da reconstrução dos sinais de 128, 256 e 512 *frames* (banda de 32 amostras)

Nº de <i>Frames</i> do Sinal a Reconstruir	Banda de Amostras do FPB	Nº de Zeros entre <i>Frames</i>	Erro Final do Algoritmo de Reconstrução		RMSE		PSNR (dB)	
			Max	Min	Max	Min	Max	Min
128	32	7	Max	0.0134	Max	0.2222	Max	118.59
			Med	0.0050	Med	0.0364	Med	80.02
			Min	4.9474e ⁻⁴	Min	2.9980e ⁻⁴	Min	61.19
256	32	3	Max	0.0118	Max	0.1852	Max	120.19
			Med	0.0044	Med	0.0304	Med	81.61
			Min	4.3450e ⁻⁴	Min	2.4953e ⁻⁴	Min	62.78
512	32	1	Max	0.0088	Max	0.1260	Max	123.53
			Med	0.0033	Med	0.0207	Med	84.95
			Min	3.2555e ⁻⁴	Min	1.6978e ⁻⁴	Min	66.12

Os resultados obtidos neste caso são muito idênticos aos obtidos com o método apresentado anteriormente, dado que a aplicação deste método também introduz degradação perceptível na sequência. Optou-se, também, por apresentar os resultados para este método para corroborar os resultados obtidos com o método anterior.

Os resultados podem ser observados com maior detalhe na secção 6.4.4.1 à secção 6.4.4.3, estes foram também obtidos utilizando uma banda de amostras do FPB igual a 32, para que a convergência do algoritmo esteja garantida.

6.4.4.1 SEQUÊNCIA 1 CODIFICADA COM O RLE – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 128 *FRAMES*

Nas Figuras 6.35 e 6.36, estão apresentados os valores do RMSE e PSNR obtidos na reconstrução de cada um dos 1620 sinais de 128 amostras, para 1024 amostras, correspondentes aos sinais de cada um dos *pixels* dos *frames*, que foram codificados com o método RLE.

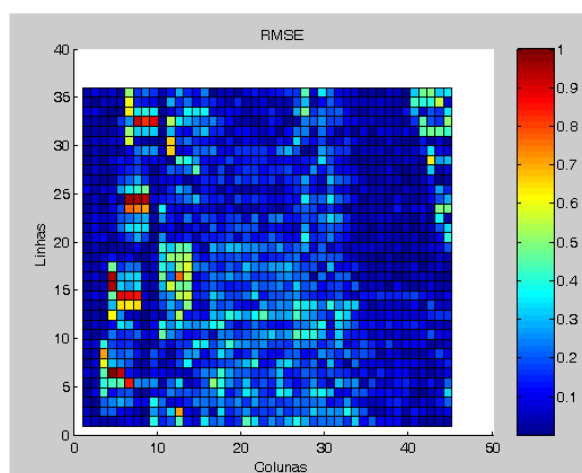


Figura 6.35 – Sequência 1 codificada com o RLE, RMSE, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 128 *frames* (banda de 32 amostras)

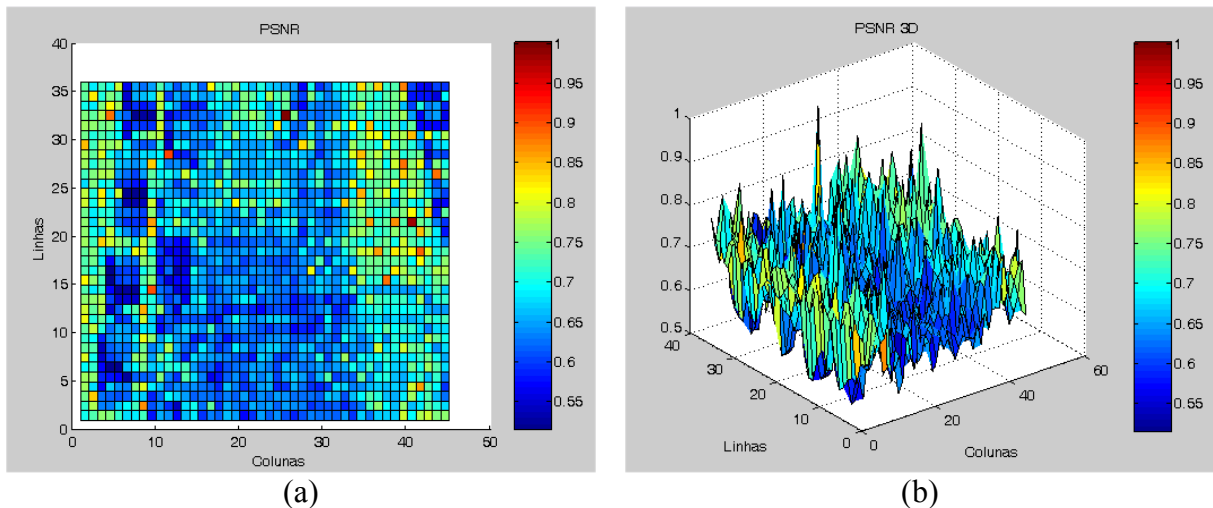


Figura 6.36 – Sequência 1 codificada com o RLE, PSNR, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 128 *frames* (banda de 32 amostras)

Da análise das Figuras 6.35 e 6.36, conclui-se que na reconstrução dos 1620 sinais observados de 1024 amostras a partir dos sinais de 128 amostras, as zonas dos *frames* da sequência em que se conseguiu obter uma melhor reconstrução foram as zonas laterais dos *frames*. No entanto, analisando a Figura 6.35 observam-se na zona lateral direita das *frames* sinais com um RMSE muito alto (zonas a vermelho). Relativamente à Figura 6.46, verifica-se que os sinais reconstruídos que apresentam um PSNR superior a 80.02 dB (valor médio), e consequentemente um RMSE mais baixo, se encontram maioritariamente nas zonas laterais dos *frames*. Após análise estatística dos resultados, verificou-se que dos 1620 sinais reconstruídos, 44.81% possuem um PSNR igual ou superior ao PSNR médio, Por outro lado, 55.19% dos sinais apresentam um PSNR inferior ao PSNR médio e superior ao mínimo. Os valores, máximo e mínimo de PSNR, obtidos nesta reconstrução são respectivamente, 118.59 dB e 61.19 dB.

6.4.4.2 SEQUÊNCIA 1 CODIFICADA COM O RLE – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 256 *FRAMES*

Nas Figuras 6.37 e 6.38, estão apresentados os valores do RMSE e PSNR obtidos na reconstrução de cada um dos 1620 sinais de 256 amostras, para 1024 amostras, correspondentes aos sinais de cada um dos *pixels* dos *frames*, que foram codificados com o método RLE.

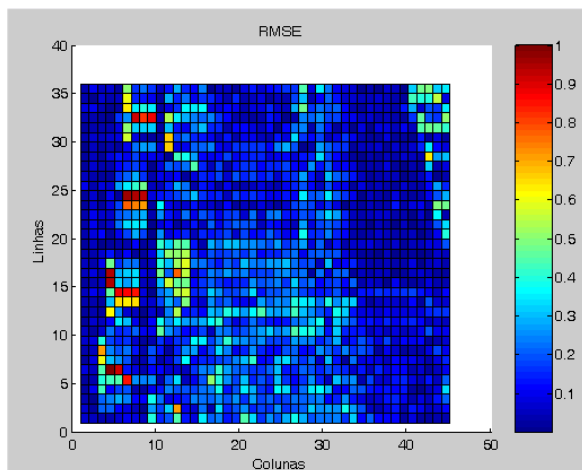


Figura 6.37 – Sequência 1 codificada com o RLE, RMSE, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 256 *frames* (banda de 32 amostras)

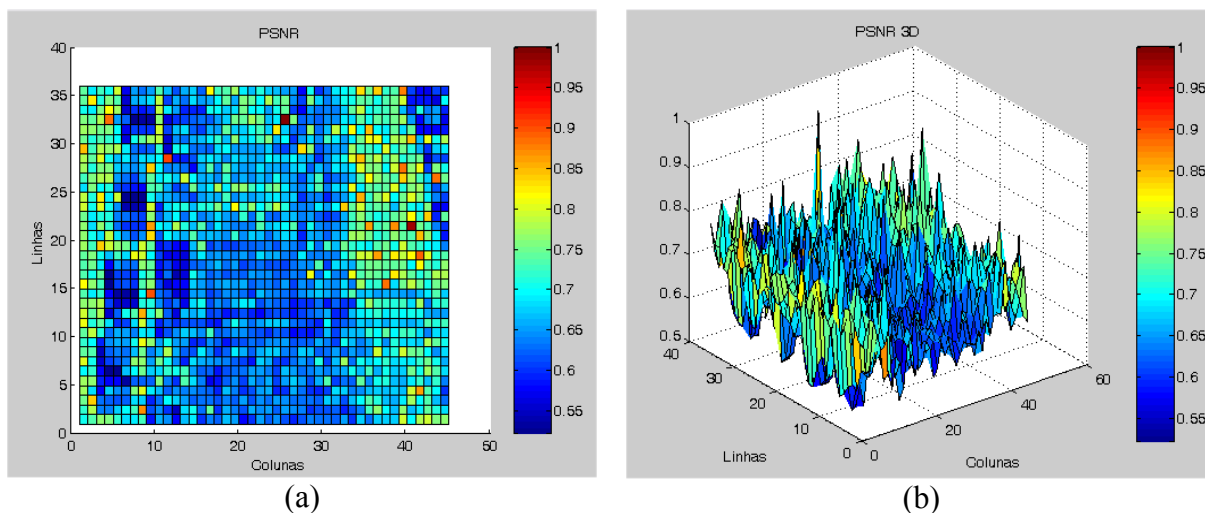


Figura 6.38 – Sequência 1 codificada com o RLE, PSNR, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 256 *frames* (banda de 32 amostras)

Analisando os resultados apresentados nas Figuras 6.37 e 6.38, verifica-se que as zonas onde se obteve melhor reconstrução dos sinais são as zonas laterais dos *frames*. A zona central dos *frames* continua a ser a zona com menor PSNR e maior RMSE. Estatisticamente, dos 1620 sinais, 44.81% possuem um PSNR igual ou superior a 81.61 dB (valor médio) e 55.19% possuem um PSNR entre 62.78 dB (valor mínimo) e o valor médio. De referir que o valor máximo, médio e mínimo de PSNR desta reconstrução são superiores aos valores obtidos na reconstrução dos sinais de 128 amostras. O valor máximo de PSNR é agora 120.19 dB. Em relação aos valores obtidos na reconstrução anterior o PSNR (máximo, médio e mínimo) aumentou aproximadamente 2 dB.

6.4.4.3 SEQUÊNCIA 1 CODIFICADA COM O RLE – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 512 FRAMES

Nas Figuras 6.39 e 6.40, estão apresentados os valores do RMSE e PSNR obtidos na reconstrução de cada um dos 1620 sinais de 512 amostras, para 1024 amostras, correspondentes aos sinais de cada um dos *pixels* dos *frames*, que foram codificados com o método RLE.

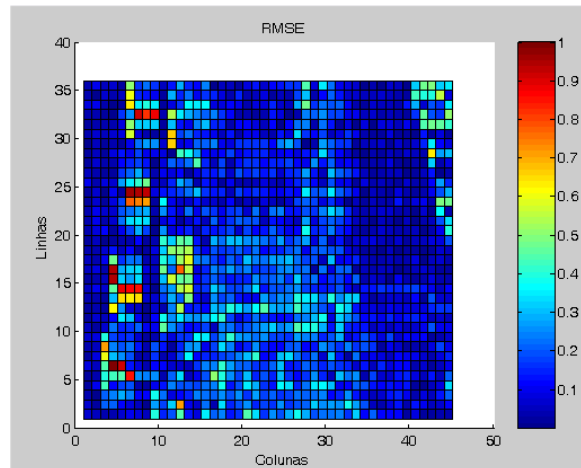


Figura 6.39 – Sequência 1 codificada com o RLE, RMSE, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 512 *frames* (banda de 32 amostras)

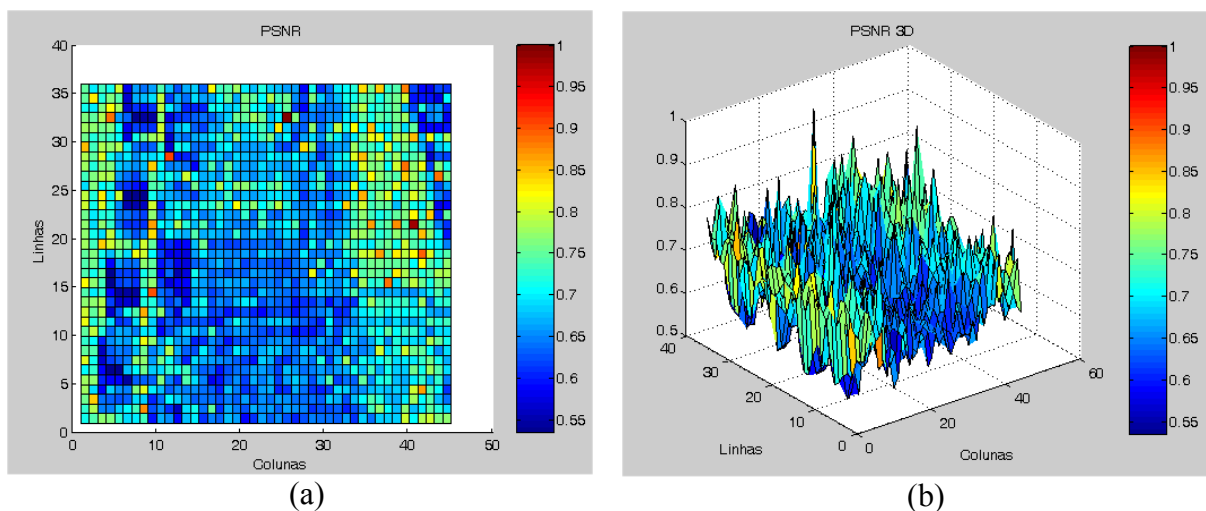


Figura 6.40 – Sequência 1 codificada com o RLE, PSNR, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 512 *frames* (banda de 32 amostras)

Analisando a Figura 6.39, verifica-se que as zonas com maior RMSE são igualmente as zonas centrais dos *frames* da sequência, os *pixels* desta zona apresentam um PSNR mais baixo, Figura 6.40. Na reconstrução dos 1620 sinais da sequência com 512 *frames*, 44.81% dos 1620 sinais possuem um PSNR maior ou igual ao valor médio, 84.95 dB e 55.19%

apresenta um valor de PSNR entre o PSNR médio e o PSNR mínimo (66.12 dB). Os sinais com um PSNR aproximadamente igual ao valor máximo representam maioritariamente sinais de *pixels* das zonas laterais dos *frames* (zonas a verde a amarelo na Figura 6.40), embora possamos encontrar sinais com o PSNR próximo do máximo na zona superior central dos *frames*.

Comparando com os resultados obtidos nas reconstruções anteriores, esta é a que apresenta valores mais elevados de PSNR; os valores: máximo, médio e mínimo aumentaram aproximadamente 2 dB, em relação à reconstrução dos 1620 sinais de 256 amostras.

Verifica-se que a percentagem de sinais com valor maior ou igual ao PSNR médio é igual em cada reconstrução, assim como a percentagem com PSNR inferior aos valores médios.

É igualmente importante notar que todas as sequências reconstruídas (128, 256 e 512 *frames*) são perceptivelmente idênticas à sequência de referência (sequência observada codificada com o método RLE).

CAPÍTULO 7 - CONCLUSÕES E TRABALHO FUTURO

Os resultados obtidos revelaram ser encorajadores. Para a maioria dos testes efectuados, conseguiu-se reconstruir a sequência observada a partir dos sinais de 128, 256 e 512 amostras. Os melhores resultados foram obtidos na reconstrução da sequência observada a partir de sinais com 512 amostras, o que é natural uma vez que para este caso, o número de amostras desconhecidas é menor. Os resultados para outras reconstruções são igualmente aceitáveis, uma vez que observando os AVI das sequências reconstruídas não se verificam grandes perdas, apenas um pequeno arrastamento em algumas zonas dos *frames* da sequência, que se torna mais visível na reconstrução dos sinais de 128 amostras. Essas áreas representam zonas dos *frames* com maior movimento, no caso da sequência 1, a zona central, que é por onde o veículo passa.

Na reconstrução de sinais nos quais se induziu maior percentagem de amostras desconhecidas, os resultados da reconstrução apresentam maior erro e menor relação sinal/ruído. Os piores resultados na reconstrução dos sinais de 128 amostras foram verificados nas zonas dos *frames* onde existe maior movimento, neste caso as zonas centrais dos *frames*. Na reconstrução dos sinais com 512 amostras, a reconstrução dos sinais possui um erro mais baixo e uma relação sinal/ruído mais elevada. Estas reconstruções (para os diferentes valores de banda 32, 64 e 128) apresentam resultados mais uniformes, ou seja, a grande percentagem dos sinais representativos do valor dos *pixels* da *frames* ao longo da sequência foram reconstruídos de forma idêntica, sempre com um PSNR acima dos 50 dB.

Relativamente às diferentes bandas utilizadas na filtragem dos sinais extraídos das sequências, esta é um factor limitativo na percentagem de amostras que se poderão perder de modo a que o algoritmo de reconstrução convirja. Ou seja, quanto maior for a banda utilizada, menor será o número de vezes que cada sinal poderá ser decimado, e assim o número de amostras perdidas. Após análise dos resultados verificou-se que, independentemente da banda utilizada, o valor máximo de PSNR e mínimo de RMSE são iguais para os três casos da reconstrução dos sinais de 512 amostras para as 1024 amostras, utilizando os três valores de banda. O aumento da banda implica um ligeiro aumento nos valores mínimo e médio de PSNR.

Verificou-se que, quanto maior for a janela de amostras perdidas, pior vai ser o desempenho do algoritmo de reconstrução de Papoulis-Gerchberg, uma vez que este possui um desempenho menos eficiente quando as amostras perdidas são contíguas [Ferreira_A, 1994]. No caso estudado o processo de decimação utilizado aumenta a janela de amostras contíguas desconhecidas. Seria interessante como trabalho futuro perder as amostras dos sinais de forma aleatória, verificando o desempenho do algoritmo de reconstrução nestas condições.

Numa sequência de vídeo, existem *pixels* cujo valor não sofre grande alteração ao longo da sequência. Isto é verdade para taxas de *frame* de 25 e 30 *frames* por segundo, e é igualmente verdade para taxas superiores, ou seja, em sequências de alta resolução temporal (taxas de *frame* superiores a 50 *frames* por segundo), que é o caso em estudo. Se o valor de um *pixel* for o mesmo ao longo da sequência, apenas precisamos de uma amostra para representar esse *pixel*. Por outro lado, se o valor de um *pixel* varia ao longo da sequência, o seu valor ao longo dos vários *frames* da sequência possui diferentes valores de NDC, então o número de amostras necessárias para representar esse sinal é mais elevado. Tendo em mente esta linha de raciocínio, surgiu a ideia, para desenvolver em trabalhos futuros, de perder o maior número de amostras possível nos sinais da sequência cujo valor sofre menos alterações, e perder menos amostras nos sinais cujo valor do *pixel* sofre mais alterações. Na Figura 7.1 observam-se três tipos de sinal: o sinal (a), a azul, para o qual necessitamos de um número de amostras elevado para o poder representar; o sinal (c), a verde, que representa um sinal contínuo, no qual necessitamos apenas uma amostra para o poder representar; e o sinal (b), a vermelho, que pode subdividir-se em duas partes: uma com maior variação (que necessita de maior número de amostras para ser representado), e outra mais contínua (que necessita de um menor número de amostras para ser representado).

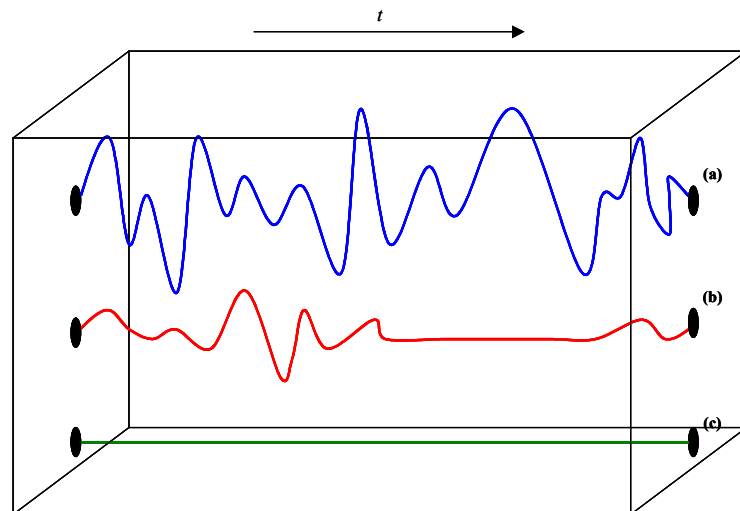


Figura 7.1 – Três tipos de sinais possíveis

O que se pretende é aplicar esta metodologia aos sinais extraídos da sequência. Na prática consiste em analisar todos os sinais da sequência e identificar as zonas com menor número de alterações nos *pixels*, e posteriormente representar essas zonas com menor número de amostras. As zonas de maior movimento, e alteração dos valores dos *pixels*, ao longo da sequência serão representadas com maior número de amostras. Esses valores mínimos de representação deverão garantir a possibilidade de reconstruir a sequência original com a taxa de sinal original. Assim guardar-se-ia a sequência por blocos, e cada bloco possuiria apenas o número de *frames* necessário para a representação dessa secção da sequência. Para reconstruir a sequência original, a reconstrução adequada seria aplicada aos diferentes blocos ou secções da sequência.

Supondo uma sequência que após extração e análise dos seus sinais, se verifica a existência dos tipos de sinais anteriormente referidos (Figura 7.1), e que se encontram nas zonas apresentadas na Figura 7.2.

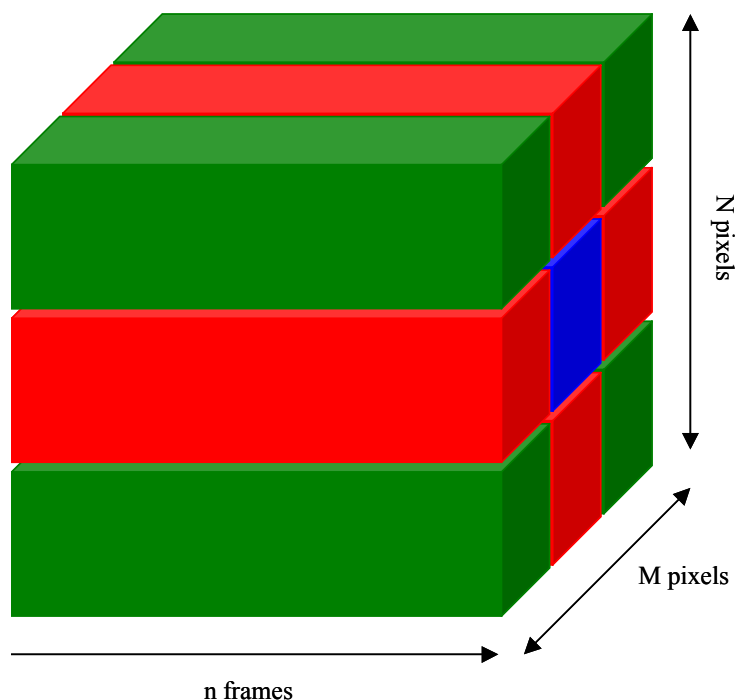


Figura 7.2 – Sequência dividida em zonas

Na Figura 7.2, as zonas a verde representam sinais que necessitam de um menor número de amostras para serem representados, logo será necessário um menor número de *frames* para estas secções. As zonas a vermelho ainda poderão ser subdivididas, identificando as zonas de maior e menor variação, permitindo deste modo diminuir o número de amostras necessárias para representar estas secções. No que diz respeito à zona a azul, esta necessitará de um número de *frames* mais elevado, uma vez que é nesta secção que se encontram os sinais com maior alteração de valores.

Por exemplo, para o sinal da Figura 7.3, supondo que este representa o valor em NDC de um *pixel* ao longo de uma sequência. Analisando o sinal, verifica-se que existem zonas que poderão ser representadas com uma amostra apenas, pois a variação dessa parte do sinal, no respectivo intervalo de tempo, é muito pequena.

Definindo essa variação máxima com sendo ε , retira-se o primeiro valor do *pixel* no instante 0, mas o segundo valor do *pixel* só será amostrado quando o seu valor sofrer uma variação (para + ou para -) superior ou a ε .

Sendo assim, utilizando o intervalo de amostragem definido pelas linhas a tracejado seriam necessárias 23 amostras para representar este sinal, com a metodologia proposta anteriormente, seriam necessárias apenas 8 amostras para representar o mesmo sinal.

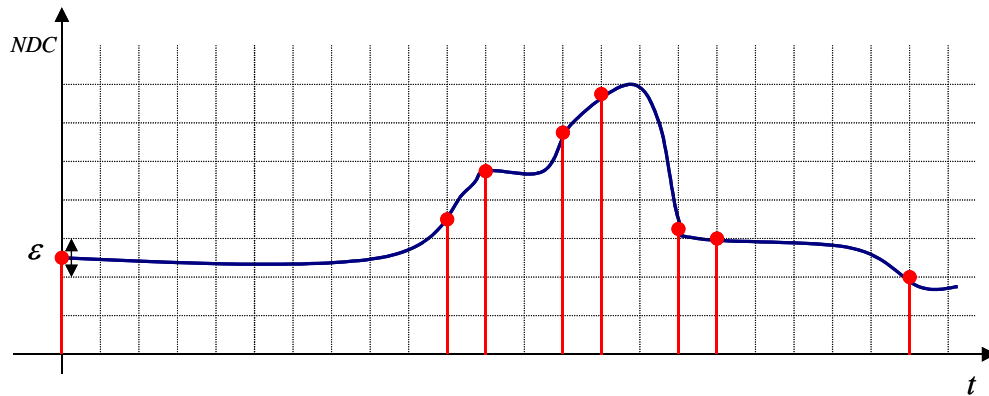


Figura 7.3 – Metodologia proposta para a amostragem dos sinais

A transição entre o caso tratado nesta dissertação e estes casos mais gerais fica facilitada pelo facto do algoritmo de reconstrução usado já permitir tratar distribuições irregulares de amostras perdidas.

Está também a ser desenvolvido sob nossa orientação um projecto para a elaboração de um interface gráfico para a aplicação desenvolvida. Este permitirá a realização dos testes e visualização de resultados de forma mais amigável.

ACRÓNIMOS

Nesta dissertação são utilizadas abreviaturas e designações comuns apenas apresentadas aquando da sua primeira utilização.

2D	2 Dimensões
3D	3 Dimensões
ASF	<i>Advanced Streaming Format</i>
AVC	<i>Advanced Video Coding</i>
AVI	<i>Audio Video Interleave</i>
AVOs	<i>Audio Visual Objects</i>
BMP	<i>Bitmap</i>
CCD	<i>Charge Coupled Device</i>
CCITT	<i>Comité Consultatif International Télégraphique et Téléphonique</i>
CD	<i>Compact Disk</i>
CIE	<i>Commission International de l'Éclairage</i>
CIF	<i>Common Intermediate Format</i>
CODEC	Codificador/Descodificador
CPU	<i>Central Processing Unit</i>
dB	Decibel
DCT	<i>Discrete Cosine Transform</i>
DFT	<i>Discrete Fourier Transform</i>
DIB	<i>Device Independent Bitmap</i>
DPCM	<i>Differential Pulse Code Modulation</i>
DSCQS	<i>Double Stimulus Continuous Quality Scale</i>
DVD	<i>Digital Versatile Disk</i>
DWT	<i>Discrete Wavelet Transform</i>
FFT	<i>Fast Fourier Transform</i>
FPB	Filtro Passa-Baixo
FPS	Frames por segundo
HDTV	<i>High Definition Television</i>
IEC	<i>International Electro-Technical Commission</i>

ACRÓNIMOS

IFFT	<i>Inverse Fast Fourier Transform</i>
ITU	<i>International Telecommunication Union</i>
ITU-R	<i>International Telecommunication Union – Radio Standardization Sector</i>
ITU-T	<i>International Telecommunication Union – Telecommunications Standardization Sector</i>
ISO	<i>International Standards Organization</i>
IYUV	<i>Intel IYUV</i>
JBIG	<i>Joint Bi-Level Image Processing Group</i>
JPEG	<i>Joint Photographic Experts Group</i>
JPEG-LS	<i>JPEG – Lossless</i>
LB	<i>Largura de Banda</i>
MPEG	<i>Moving Picture Experts Group</i>
MSE	<i>Mean Square Error</i>
MV1	<i>Microsoft Video 1.0</i>
NDC	<i>Níveis de Cinzento</i>
NTSC	<i>National Television System(s) Committee</i>
PAL	<i>Phase Alternation by Line</i>
PSNR	<i>Peak Signal-to-Noise Ratio</i>
QCIF	<i>Quarter Common Intermediate Format</i>
QSIF	<i>Quarter Standard Image Format</i>
RCC	<i>Radius Cinepak Codec</i>
RGB	<i>Red, Green, Blue</i>
RGC	<i>Reflected Gray Code</i>
RLE	<i>Run Length Encoding</i>
RMSE	<i>Root Mean-Squared Error</i>
ROI	<i>Regions of Interest</i>
ROM	<i>Read Only Memory</i>
SECAM	<i>Séquentiel Couleur avec Mémoire</i>
SPD	<i>Spectral Power Distribution</i>
TIFF	<i>Tag(ged) Image File Format</i>
VCEG	<i>Video Coding Experts Group</i>
VHS	<i>Video Home System</i>
VQEG	<i>Video Quality Experts Group</i>

REFERÊNCIAS BIBLIOGRÁFICAS

- [Boch *et al.*, 1997] L. Boch, S. Fragola, R. Lancini, P. Sunna and M. Visca: “*Motion Detection on Video Compressed Sequences as a Tool to Correlate Objective Measure and Subjective Score*”, 13th International Conference on Digital Signal Processing, DSP 97, 1997.
- [Castleman, 1996] Kenneth R. Castleman: “*Digital Image Processing*”, Prentice Hall Inc., 1996 ISBN: 0-13-211467-4
- [Christopoulos *et al.*, 2000] Charilaos Christopoulos, Athanassios Skodras and Touradj Ebrahimi: “*The JPEG2000 Still Image Coding System: An Overview*”, IEEE Transactions on Consumer Electronics, Vol. 46, No. 4, pp 1103-1127, November 2000.
- [Cooley e Tukey, 1965] Cooley, J. W., John. W. Tukey: “*An algorithm for the machine calculation of complex Fourier series*”, Math. Computation, 19:297-301, 1965.
- [Cruz e Ebrahimi, 2000] Diego Santa Cruz and Touradj Ebrahimi: “*A Study of JPEG2000 Still Image Coding Versus Other Standards*”, X European Signal Processing Conference, Tampere, Finland, September, 2000.
- [Data Compression, 2005] Data-Compression.com, URL: <http://www.data-compression.com/index.shtml>, consultado em Junho 2005.
- [Ferguson, 2001] Monash University, Dr. Tim Ferguson Page homepage, URL: <http://www.csse.monash.edu.au/~timf/>, consultado em Setembro, 2005.
- [Ferreira_A, 1994] Ferreira, P. J. S. G.: “*Interpolation and the Discrete Papoulis-Gershberg Algorithm*”, IEEE Transactions on Signal Processing, Vol. 42, No. 10, 1994.
- [Ferreira_B, 1994] Ferreira, P. J. S. G.: “*Non Iterative and Fast Iterative Methods for Interpolation and Extrapolation*”, IEEE Transactions on Signal Processing, 42(11):3278-3282, 1994.
- [Ferreira, 1996] Ferreira, P. J. S. G.: “*Interpolation in the Time and Frequency Domains*”, IEEE Signal Processing Letters, 3(6):176-178, 1996.

REFERÊNCIAS BIBLIOGRÁFICAS

- [Ferreira, 1999] Ferreira, P. J. S. G.: “*Mathematics for multimedia signal processing II – discrete finite frames and signal reconstruction. Signal Processing for Multimedia*”, pages 35-54. IOS Press, J. S. Byrnes, Ed., 1999
- [FOURCC, 2005] FOURCC.org, URL: <http://www.fourcc.org>, consultado em Outubro de 2005.
- [Gall, 1991] Didier Le Gall: “*MPEG: A Video Compression Standard for Multimedia Applications*”, ACM, Vol. 34, No.4, April 1991.
- [Gonzalez e Wintz, 1987] Rafael C. Gonzalez, Paul Wintz: “*Digital Image Processing*”, Addison-Wesley Publishing Company, Inc., 1987.
ISBN: 0-201-11026-1
- [H.261, 1993] ITU-T Recommendation H.261, Video CODEC for audiovisual services at px64 Kbit/s, 1993.
- [H.263, 1998] ITU-T Recommendation H.263, Video coding for low bit rate communication, Version 2, 1998.
- [H.264, 2003] ISO/IEC 14496-10 and ITU-T Rec. H.264, Advanced Video Coding, 2003.
- [Hankersson *et al.*, 1998] Darrel Hankersson, Greg A. Harris, Peter D. Johnson, Jr., “*Introduction to Information Theory and Data Compression*”, CRC Press, 1998.
ISBN: 0-8493-3985-5
- [Hilton *et al.*, 1994] Michael L. Hilton, Björn D. Jawerth, Ayan Sengupta, “*Compressing still and moving images with wavelets*”, Multimedia Systems, v.2 n.5, p.218-227, Dec. 1994.
- [Horn e Johnson, 1985] Horn, R. A., C. R. Johnson: “*Matrix Analysis*”, Cambridge University Press, New York, 1985.
- [Infaimon, 2005] Infaimon, URL: <http://www.infaimon.com/catalogo/industria/index.htm>, consultado em Setembro de 2005.
- [JBIG2, 2005] Página Oficial dos Comitês JBIG e JPEG, URL: <http://www.jpeg.org/jbig/jbigpt2.html?langsel=pt>, consultado em Junho 2005.

- [JPEG, 1992] ISO/IEC 10918-1/ITU-T Recommendation T.81, Digital compression and coding of continuous-tone still images, 1992.
- [JPEG2000, 2000] ISO/IEC 15444, Information technology – JPEG2000 Image coding system, 2000.
- [Lu *et al.*, 2002] Ligang Lu, Zhou Wang and Alan C. Bovik: “*Full Reference Video Quality Assessment Considering Structural Distortion and No-Reference Quality Evaluation of MPEG Video*”, IEEE International Conference on Multimedia and Expo (ICME2002), Swiss Federal Institute of Technology, EPFL, Lausanne, Switzerland, August 26-29, 2002.
- [MPEG1, 1993] ISO/IEC 11172, Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbits, 1993.
- [MPEG2, 1995] ISO/IEC 13818, Information technology: generic coding of moving pictures and associated audio information, 1995.
- [MPEG4-2, 2001] ISO/IEC 14496-2, Coding of audio-visual objects – Part 2: Visual, 2001
- [MSA, 2004] MSA (Multimedia Systems and Applications), Image Quality Computation, URL: <http://bmrc.berkeley.edu/courseware/cs294/fall97/assignment/psnr.html>, consultado em Dezembro 2004.
- [MRI_A, 2005] MRI (Multimedia Research Institute), Video Codecs, RLE, URL: <http://www.multimedia.cx/msrle.txt>, consultado em Setembro 2005.
- [MRI_B, 2005] MRI (Multimedia Research Institute), Video Codecs, Microsoft Video-1, URL: <http://www.multimedia.cx/video1.txt>, consultado em Setembro 2005.
- [Papoulis, 1975] Papoulis, A.: “*A New Algorithm in Spectral Analysis and Band-Limited Extrapolation.*”, IEEE Transactions on Circuits and Systems, 22(9):735-742, 1975.
- [Pereira e Ebrahimi, 2002] Fernando Pereira, Touradj Ebrahimi: “*The MPEG-4 book*”, Pearson Education, Inc., 2002.
ISBN: 0-13-061621-4

REFERÊNCIAS BIBLIOGRÁFICAS

- [Pei e Yeh, 1997] Pei, S. C., M. H. Yeh.: “*An introduction to finite frames.*”, IEEE Signal Processing Mag., 14(6):84-96, 1997.
- [Phoenix, 2005] Phoenix Imaging Machine Vision Systems, URL: <http://www.vision-lights.com/Lighting%20Products.htm>, consultado em Setembro de 2005.
- [Photosonics, 2005] Photosonics International, URL:<http://www.photosonicsinternational.co.uk>, consultado em Junho de 2005.
- [Rao e Hwang, 1996] K. R. Rao, J. J. Hwang: “*Techniques & Standards for Image – Video & Audio Coding*”, Prentice Hall PTR, 1996.
ISBN: 0-13-309907-5
- [Redlake, 2005] Redlake, URL:<http://www.redlake.com>, consultado em Junho de 2005.
- [Reis, 1996] Manuel José Cabral dos Santos Reis: “*Reconstrução de Imagem, Método de dimensão mínima no domínio do tempo*”, Tese apresentada à Universidade de Aveiro, Maio de 1996.
- [Reis, 2000] Reis, M. J. C. S.: “*Representação multi-resolução de sinais baseados em Splines e sobre-amostragem.*”, Tese de Doutoramento apresentada à Universidade de Aveiro, Portugal, 2000.
- [Richardson, 2003] Iain E. G. Richardson: “*H.264 and MPEG-4 Video Compression: Video Coding for Next – generation Multimedia*”, John Wiley & Sons, Ltd., 2003.
ISBN: 0-470-84837-5
- [Salomon, 2000] David Salomon: “*Data Compression – The Complete Reference*”, Springer-Verlag New York, Inc., 2000.
ISBN 0-387-95045-1
- [Sanz e Huang, 1983] Sanz, Jorge L. C., Thomas S. Huang.: “*Some Aspects of Band-Limited Signal Extrapolation: Models, Discrete Approximations, and Noise.*”, IEEE Transactions on Acous. Spe. And Sig. Proc., 31(6):1492-1501, 1987.
- [Shapiro e Stockman, 2001] Linda G. Shapiro, George C. Stockman: “*Computer Vision*”, Prentice-Hall, Inc., 2001.
ISBN 0-8493-3491-8

- [Shi e Sun, 2000] Yun Q. Shi, Huifang Sun: “*Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards*”, CRC Press LLC, 2000.
ISBN: 0-13-030796-3
- [Skodras *et al.*, 2000] A.N. Skodras, C. A. Christopoulos and T. Ebrahimi: “*JPEG2000: The Upcoming Still Image Compression Standard*”, 11th Portuguese Conference on Pattern Recognition, Porto, Portugal, May 11th-12th, pp 359-366, 2000.
- [Soares, 2003] Soares, S. F. S. P.: “*Algoritmos de Reconstrução de Sinal e Correção de Erros*”, Tese de Doutoramento apresentada à Universidade de Aveiro, Portugal, 2003.
- [Sonka *et al.*, 1999] Milan Sonka, Vaclaw Hlavac, Roger Boyle: “*Image Processing, Analysis, and Machine Vision*”, Brooks/Cole Publishing Company, 1999.
ISBN: 0-534-95393-x
- [Techonline, 2005] Techonline, “*An Overview of Video Compression Algorithms*” by Andrew Davis, URL: <http://www.techonline.com/community/home/20028>, consultado em Junho 2005.
- [Vieira e Ferreira, 1997] Vieira, J. M. N., P. J. S. G. Ferreira: “*Interpolation, spectrum analysis, error-control coding, and fault-tolerant computing*”, in Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing, ICASSP 97, (III):1831-1834, 1997.
- [VQEG, 2005] Video Quality Experts Group, URL: <http://www.its.bldrdoc.gov/vqeg/>, consultado em Setembro de 2005.
- [Wade e Swanson, 2001] Nicholas J. Wade and Michael T. Swanson: “*Visual Perception: An Introduction*”, Psychology Press Ltd, 2001.
ISBN 1-84169-203-4
- [Webster *et al.*, 1993] Arthur A. Webster, Colen T. Jones, Margaret H. Pinson, Stephen D. Voran and Stephen Wolf: “*An objective video quality assessment system based on human perception*”, in Proceedings SPIE, vol 1913, pp. 15-26, 1993.

ANEXO

Resultados obtidos para a Sequência 2

SEQUÊNCIA 2 – RESULTADOS PARA BANDA DE 32 AMOSTRAS

Esta sequência possui um total de 1085 (35×31) sinais cada um com 1024 amostras (Tabela 5.1). Na Figura A.1 podem observar-se 54 *frames* da sequência 2 observada. Nesta sequência, para que pudessem ser geradas e posteriormente reconstruídas três novas sequências, com um menor número de *frames* do que a sequência observada, utilizou-se como valor de banda do FPB 32 amostras. De acordo com a Figura 5.7, foram criadas três novas sequências com 512, 256 e 128 *frames*.

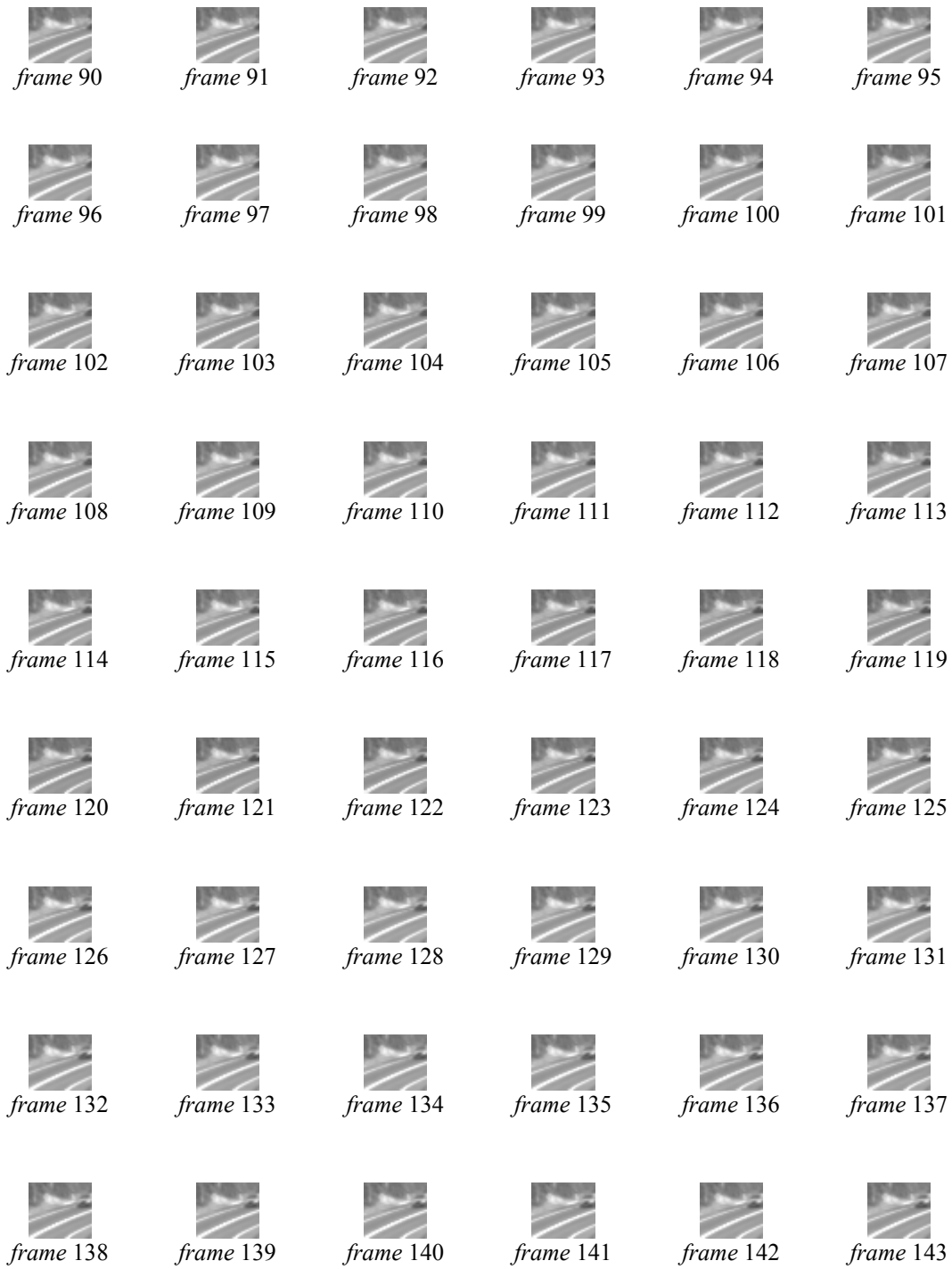


Figura A.1 – Alguns *frames* da sequência 2 observada, banda de 32 amostras no FPB

Na reconstrução do sinal observado (1024 amostras) a partir dos sinais de 512, 256 e 128 amostras, foram obtidos os resultados apresentados na Tabela A.1, nesta tabela apresentam-se apenas os valores máximo, médio e mínimo de todos os parâmetros analisados.

Os resultados seguintes foram obtidos utilizando uma banda de amostras do FPB igual a 32, isto para que a convergência do algoritmo esteja garantida de acordo com a Tabela 5.2.

Tabela A.1 – Sequência 2, Resultados da reconstrução dos sinais de 128, 256 e 512 *frames* (banda de 32 amostras)

Nº de <i>Frames</i> do Sinal a Reconstruir	Banda do FPB	Nº de zeros entre <i>Frames</i>	Erro Final do Algoritmo de Reconstrução		RMSE		PSNR (dB)	
			Max	Med	Max	Med	Max	Med
128	32	7	Max	0.0399	Max	1.9239	Max	116.56
			Med	0.0146	Med	0.3457	Med	62.67
			Min	5.8820e ⁻⁴	Min	3.7875e ⁻⁴	Min	42.45
256	32	3	Max	0.0349	Max	1.6031	Max	198.65
			Med	0.0128	Med	0.2881	Med	64.33
			Min	5.0185e ⁻⁶	Min	2.9779e ⁻⁸	Min	44.03
512	32	1	Max	0.0260	Max	1.0907	Max	207.71
			Med	0.0096	Med	0.1960	Med	67.68
			Min	2.6921e ⁻⁶	Min	1.0496e ⁻⁸	Min	47.38

SEQUÊNCIA 2 – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 128 *FRAMES*

Nas Figuras A.2 e A.3 estão apresentados os valores do RMSE e PSNR obtidos na reconstrução de cada um dos 1085 sinais de 128 amostras, para 1024 amostras, correspondentes aos sinais de cada um dos *pixels* dos *frames*. Os resultados apresentados estão normalizados.

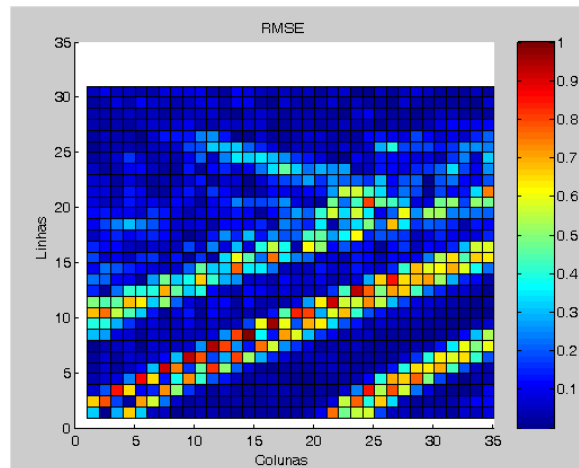


Figura A.2 – Sequência 2, RMSE, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 128 *frames* (banda de 32 amostras)

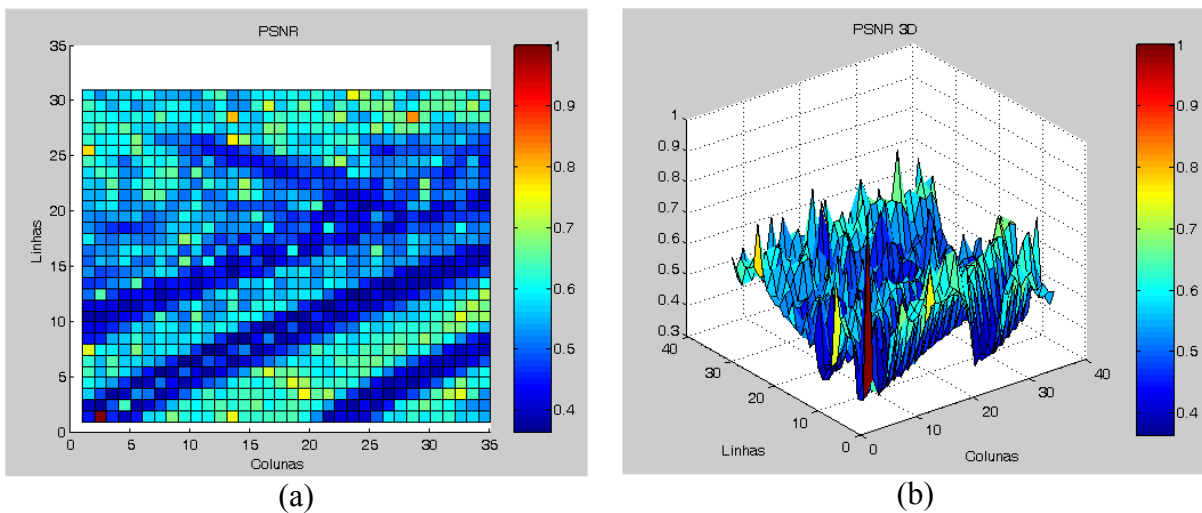


Figura A.3 – Sequência 2, PSNR, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 128 *frames* (banda de 32 amostras)

Na reconstrução dos sinais com 128 amostras para sinais com 1024 amostras obtiveram-se os seguintes resultados, 51.71% dos sinais reconstruídos possuem um PSNR maior ou igual a 62.67 dB (valor médio). Os restantes 48.29% apresentam um PSNR igual ou superior a 42.45 dB (valor mínimo) e inferior ao PSNR médio. O valor máximo de PSNR obtido nesta reconstrução foi 116.56 dB.

Na Figura A.4 são apresentados 54 *frames* da sequência de 128 *frames* reconstruída que podem ser comparados com os *frames* da Figura A.1.

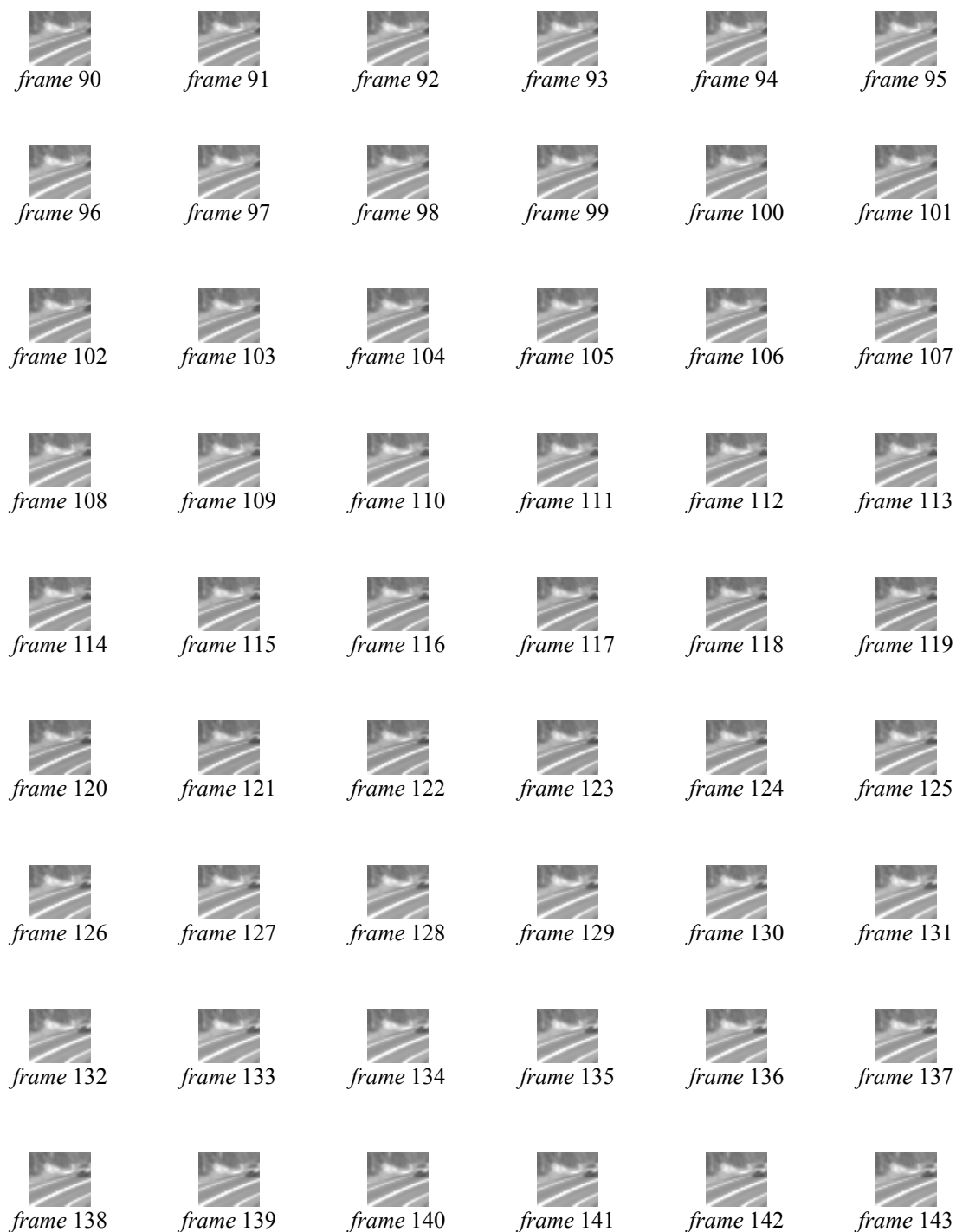


Figura A.4 – Alguns *frames* da sequência 2 reconstruída a partir da sequência de 128 *frames*, banda de 32 amostras no FPB

SEQUÊNCIA 2 – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 256 FRAMES

Nas Figuras A.5 e A.6 estão apresentados os valores do RMSE e PSNR obtidos na reconstrução de cada um dos 1085 sinais de 256 amostras, para 1024 amostras, correspondentes aos sinais de cada um dos *pixels* dos *frames*. Os resultados apresentados estão normalizados.

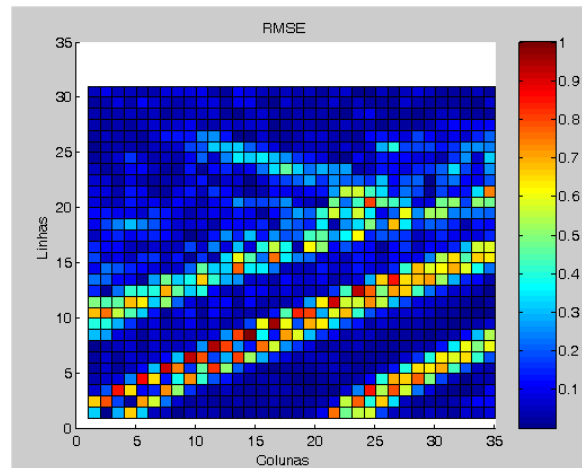


Figura A.5 – Sequência 2, RMSE, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 256 *frames* (banda de 32 amostras)

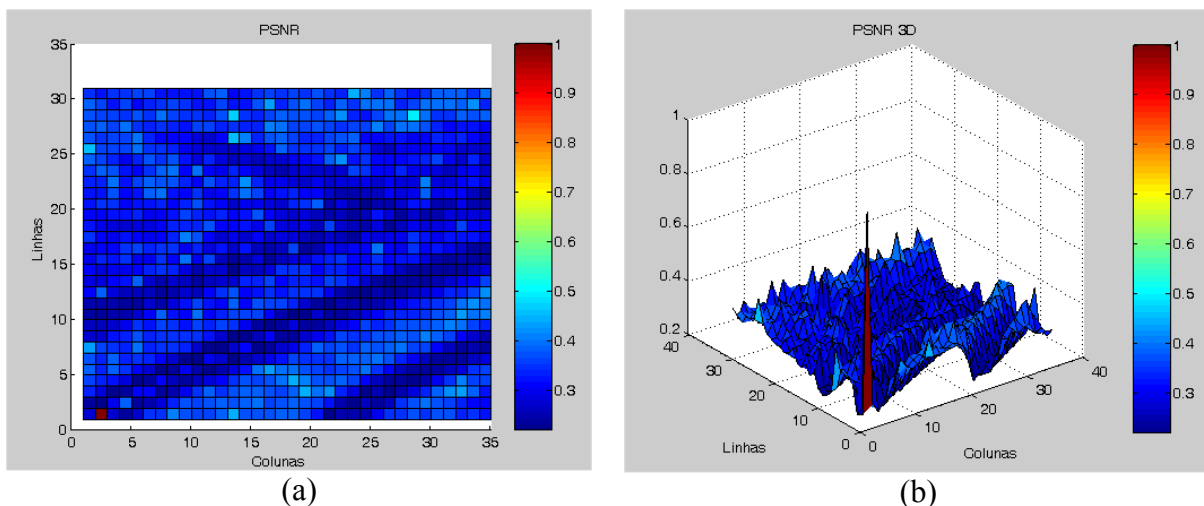


Figura A.6 – Sequência 2, PSNR, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 256 *frames* (banda de 32 amostras)

Na reconstrução dos sinais com 256 amostras para sinais com 1024 amostras obtiveram-se os seguintes resultados, 51.61% dos sinais reconstruídos possuem um PSNR maior ou igual a 64.33 dB (valor médio). Os restantes 48.39% apresentam um PSNR igual ou superior a

44.03 dB (valor mínimo) e inferior ao PSNR médio. O valor máximo de PSNR obtido nesta reconstrução foi 198.65 dB.

Na Figura A.7 são apresentados 54 *frames* da sequência de 256 *frames* reconstruída que podem ser comparados com os *frames* da Figura A.1.

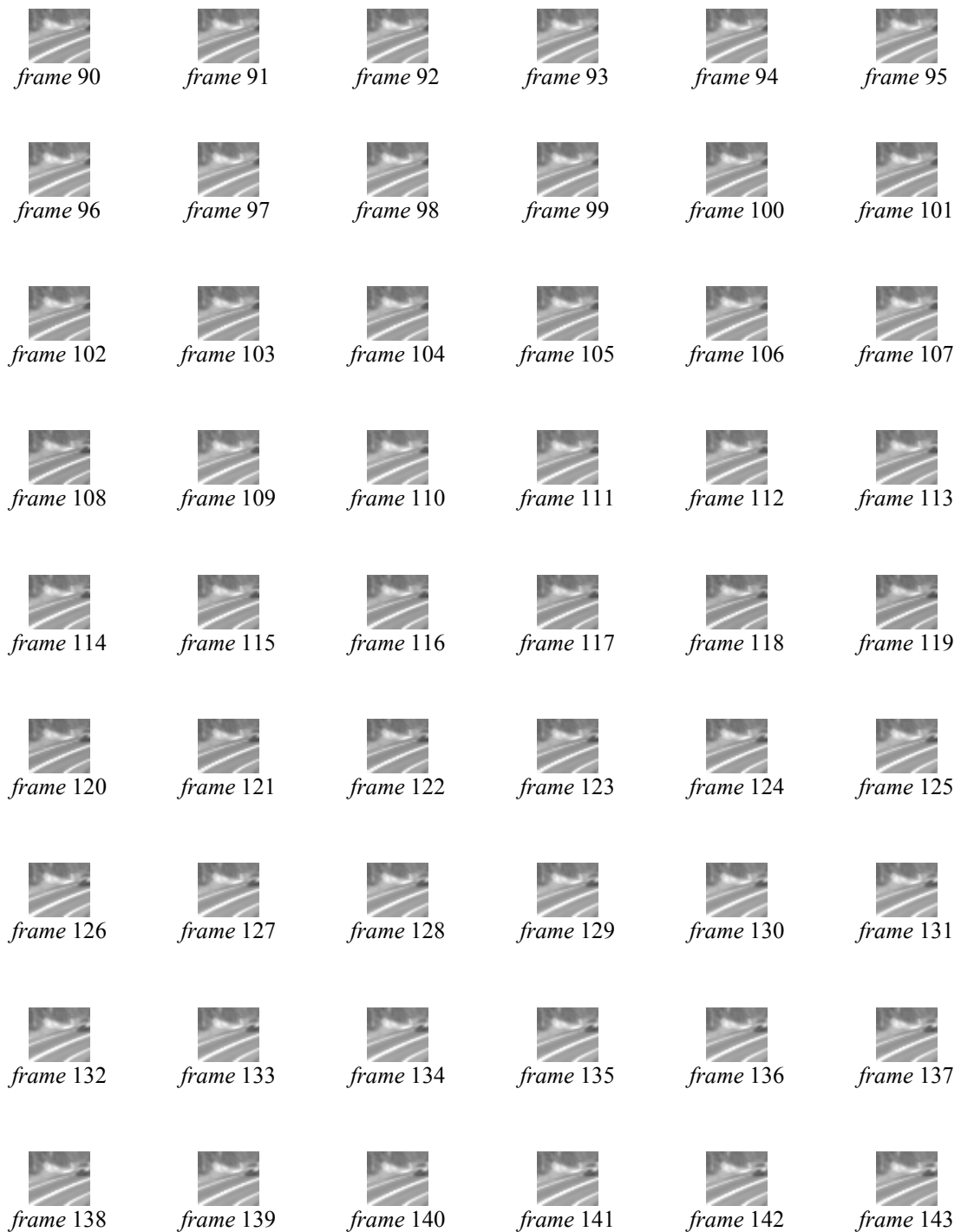


Figura A.7 – Alguns *frames* da sequência 2 reconstruída a partir da sequência de 256 *frames*, banda de 32 amostras no FPB

SEQUÊNCIA 2 – RECONSTRUÇÃO A PARTIR DA SEQUÊNCIA DE 512 FRAMES

Nas Figuras A.8 e A.9 estão apresentados os valores do RMSE e PSNR obtidos na reconstrução de cada um dos 1085 sinais de 512 amostras, para 1024 amostras, correspondentes aos sinais de cada um dos *pixels* dos *frames*. Os resultados aqui apresentados estão normalizados.

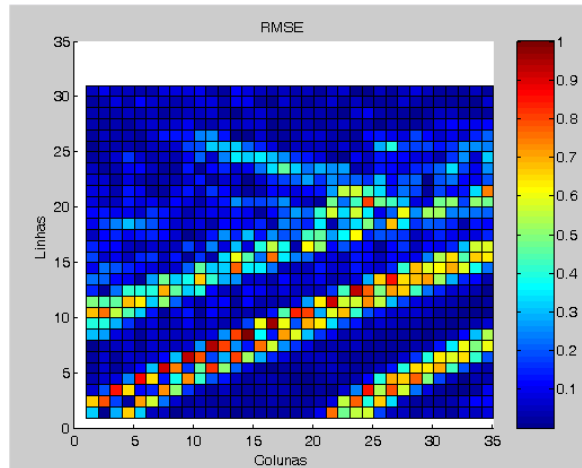


Figura A.8 – Sequência 2, RMSE, resultados por *pixel* a 2D obtidos na reconstrução dos sinais da sequência de 512 *frames* (banda de 32 amostras)

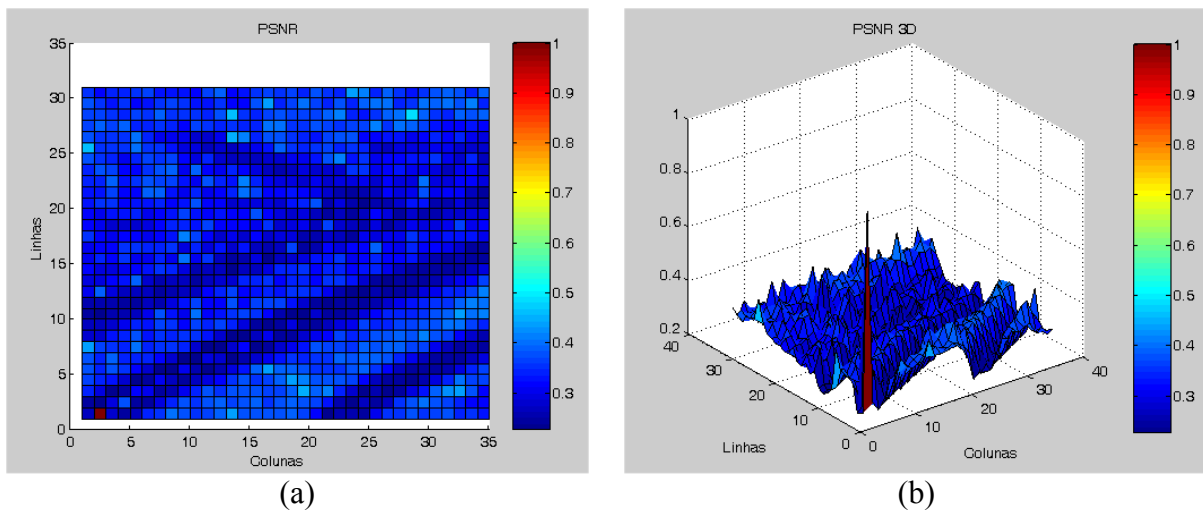


Figura A.9 – Sequência 2, PSNR, resultados por *pixel* a 2D (a) e a 3D (b) obtidos na reconstrução dos sinais da sequência de 512 *frames* (banda de 32 amostras)

Na reconstrução dos sinais com 512 amostras para sinais com 1024 amostras obtiveram-se os seguintes resultados, 51.61% dos sinais reconstruídos possuem um PSNR maior ou igual a 67.68 dB (valor médio). Os restantes 48.39% apresentam um PSNR igual ou superior a 47.38 dB (valor mínimo) e inferior ao PSNR médio. Como se pode observar

através das Figuras A.8 e A.9, os resultados da reconstrução idênticos aos da reconstrução dos sinais de 256 amostras, no entanto os valores mínimo e médio do PSNR aumentaram ligeiramente, na ordem dos 3 dB. O valor máximo do PSNR é igual a 207,71 dB.

Na Figura A.10 são apresentados 54 *frames* da sequência de 512 *frames* reconstruída que podem ser comparados com os *frames* da Figura A.1.

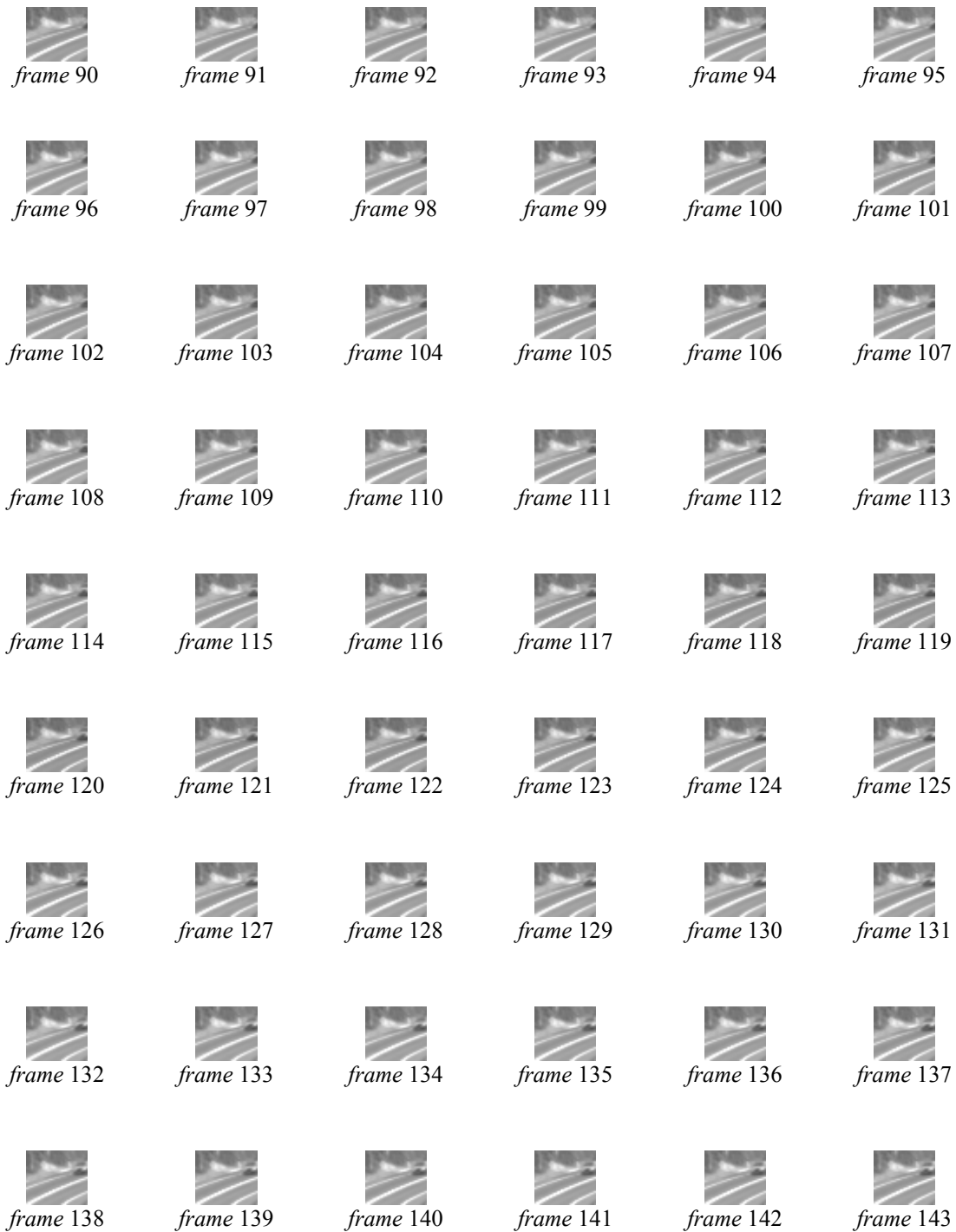


Figura A.10 – Alguns *frames* da sequência 2 reconstruída a partir da sequência de 512 *frames*, banda de 32 amostras no FPB