

Universidade Federal do Rio de Janeiro  
Escola Politécnica  
Departamento de Eletrônica e de Computação

**Codificação de Imagens Estéreo utilizando Métodos de  
Compressão baseados na Recorrência de Padrões  
Multiescala**

Autor:

---

Thiago Pedra Signorelli

Orientador:

---

Prof. Eduardo Antônio Barros da Silva, Ph. D.

Examinador:

---

Prof. Gelson Vieira Mendonça, Ph. D.

Examinador:

---

Prof. Sergio Lima Netto, Ph. D.

DEL

Dezembro de 2009

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Escola Politécnica – Departamento de Eletrônica e de Computação

Centro de Tecnologia, bloco H, sala H-217, Cidade Universitária

Rio de Janeiro – RJ CEP 21949-900

Este exemplar é de propriedade da Universidade Federal do Rio de Janeiro, que poderá incluí-lo em base de dados, armazenar em computador, microfilmear ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

## **DEDICATÓRIA**

Dedico este trabalho aos meus pais, Vicente e Marise, que constituem meu modelo de família e caráter, bem como ao meu irmão, Bruno, pelo incentivo durante os momentos mais difíceis me impedindo de desistir.

Dedico à memória de meus avós, Vincenzo, Gelsomina, Francisco e Maria do Céu, fundamentais durante a minha criação, sendo meu modelo de força e trabalho.

## **AGRADECIMENTOS**

Agradeço ao professor, orientador e amigo Eduardo pela paciência durante as explicações, por ter acreditado em mim e por ter me conduzido ao longo do árduo caminho de elaboração deste trabalho.

Agradeço aos amigos Nelson e José Fernando pelos esclarecimentos a respeito do código fonte do algoritmo MMP.

Por fim, agradeço a toda equipe de professores do DEL por me ensinarem os fundamentos teóricos necessários ao desenvolvimento deste projeto.

## RESUMO

Este projeto tem como ponto de partida o algoritmo *Multidimensional Multiscale Parser* (MMP). Essa nova classe de algoritmos baseia-se no casamento aproximado de padrões recorrentes com escala, ou seja, o algoritmo tenta representar um sinal de entrada utilizando versões contraídas e expandidas de blocos previamente codificados.

O casamento aproximado é feito com elementos pertencentes a um dicionário seguindo um critério de controle. Caso tal critério não seja satisfeito, o bloco é, então, particionado e o processo se repete com novas tentativas de casamentos.

O dicionário é atualizado à medida que o sinal é processado sendo, portanto, adaptativo. O critério de controle pode ainda ser ajustado de forma a permitir uma compressão sem perdas.

O objetivo deste trabalho é melhorar o desempenho de um codificador existente para compressão de imagens estéreo usando o MMP. Com esse intuito, foram implementadas melhorias tais como: segmentação flexível, divisão do dicionário em contextos e controle de redundância entre os elementos do dicionário.

Baseando-se no codificador anterior foram usadas técnicas que permitem alcançar resultados de alta qualidade das imagens reconstruídas sem, no entanto, transmitir o mapa de disparidades. Dessa forma, evitam-se transtornos referentes ao seu cálculo preciso, fases de pré e pós processamento, geração de imagens de erro, além de sua codificação e transmissão.

Em resumo, neste trabalho evita-se todo o processo de estimação de disparidades, gerando, no entanto, imagens reconstruídas que apresentam qualidade comparável com aquelas originadas através de métodos de codificação de imagens estéreo em que tal mapa é transmitido.

Palavras-Chave: compressão de imagens, recorrência de padrões multiescalas, casamento de padrões, imagens estéreo.

## ABSTRACT

This project has the *Multidimensional Multiscale Parser* (MMP) algorithm as a starting point. This new class of algorithms is based on the approximate match of patterns that belong to a dictionary. In other words, the algorithm tries to represent an input signal using expanded and contracted versions of previously encoded blocks.

The block matching is made with elements that belong to a dictionary using a control criterion. If this criterion is not satisfied, the block is then partitioned and the process is repeated with new attempts to establish a better match.

The dictionary is updated as the signal is being processed and is, therefore, adaptive. The criterion of control can be adjusted to allow lossless compression.

The objective of this work is to improve the performance of an existing encoder that uses the MMP to compress stereo images. To achieve this objective, improvements have been made, such as: flexible segmentation, dictionary division in contexts and redundancy control among dictionary elements.

Based on the previous encoder, were used techniques that enable the encoder to achieve high quality results in the reconstructed images without, however, transmitting the disparity map. That is, the burden of its calculation, the pre and post-processing phases, the generation of the error images, as well as their coding and transmission, are not necessary.

In summary, this work avoids the entire disparities estimation process, creating, thought, reconstructed images that have a comparable quality to those generated by encoding methods of stereo images that send this map.

Key-words: image compression, multi-scale recurrent patterns, block match, stereo images.

## SIGLAS

UFRJ – Universidade Federal do Rio de Janeiro

MMP – *Multidimensional Multiscale Parser*

ROB – *row of blocks*

# Sumário

## Capítulo 1 – Introdução

1.1 Tema.....	01
1.2 Delimitação.....	01
1.3 Justificativa.....	01
1.4 Objetivos.....	03
1.5 Metodologia.....	03
1.6 Descrição.....	04

## Capítulo 2 – O algoritmo MMP

2.1 Descrição do MMP.....	06
2.2 Aplicação do algoritmo MMP às imagens.....	08
2.2.1 O processo de codificação.....	08
2.2.2 A atualização do dicionário.....	14
2.2.3 O processo de decodificação.....	20
2.3 Resultados.....	20
2.4 Conclusão.....	27

## Capítulo 3 – Melhorias adicionadas ao algoritmo MMP

3.1 Uso de técnicas de predição.....	29
3.1.1 O algoritmo MMP com uso de predição.....	30
3.1.2 O dicionário do algoritmo MMP-I.....	31
3.2 Controle de redundância dos elementos do dicionário.....	32
3.3 Novas técnicas de atualização do dicionário.....	34
3.4 Divisão do dicionário em gavetas.....	35
3.5 Segmentação flexível.....	37
3.6 Resultados.....	42



3.7 Conclusão.....	44
<b>Capítulo 4 – MMP Estéreo</b>	
4.1 O MMP Estéreo.....	46
4.2 A imagem esquerda.....	46
4.2.1 O processo de codificação da imagem esquerda.....	46
4.2.2 O processo de decodificação da imagem esquerda.....	48
4.3 A imagem direita.....	48
4.3.1 Os blocos deslocados.....	49
4.3.2 Os blocos deformados.....	52
4.3.2.1 Cálculo do mapa de disparidades.....	52
4.3.2.2 Construção do dicionário com elementos deformados.....	54
4.3.3 O processo de codificação da imagem direita.....	56
4.3.4 O processo de decodificação da imagem direita.....	57
4.4 Resultados.....	57
4.5 Conclusão.....	64
<b>Capítulo 5 – Conclusões</b>	
Referências Bibliográficas.....	68
Apêndice A.....	69
Apêndice B.....	73
Apêndice C.....	75

# Lista de Figuras

Figura 2.1 - Esquema de codificação do MMP.....	08
Figura 2.2 - Diagrama de uma árvore binária que representa um bloco 8 x 8 e suas possíveis divisões, partindo de uma segmentação vertical.....	09
Figura 2.3 - Elemento de índice 0 do nível 4 do dicionário inicial.....	10
Figura 2.4 - Elemento de índice 4 do nível 4 do dicionário inicial.....	10
Figura 2.5 - Diagrama representando um nó de uma árvore binária e seus nós filhos....	12
Figura 2.6 - Árvore ótima após o processo de segmentação.....	13
Figura 2.7 - Representação da contração e expansão de um bloco da imagem.....	15
Figura 2.8(a) - Bloco segmentado horizontalmente e sua árvore de segmentação.....	16
Figura 2.8 (b) - Bloco superior segmentado verticalmente e sua árvore de segmentação.....	17
Figura 2.8 (c) - Bloco inferior segmentado verticalmente e sua árvore de segmentação.....	17
Figura 2.9 (a) - Primeira atualização do dicionário.....	18
Figura 2.9 (b) - Segunda atualização do dicionário.....	19
Figura 2.9 (c) - Terceira atualização do dicionário.....	19
Figura 2.10 – Performance R-D com a imagem Aerial (512 x 512).....	21
Figura 2.11 – Performance R-D com a imagem Baboon (512 x 512).....	22
Figura 2.12 – Performance R-D com a imagem Barbara (512 x 512).....	23
Figura 2.13 – Performance R-D com a imagem Bridge (512 x 512).....	24
Figura 2.14 – Performance R-D com a imagem Gold (512 x 512).....	25
Figura 2.15 – Performance R-D com a imagem Lena (512 x 512).....	26
Figura 2.16 – Performance R-D com a imagem PP1205 (512 x 512).....	27
Figura 3.1 – Nova representação dos elementos do dicionário.....	33
Figura 3.2 – Divisão de um nível do dicionário em gavetas.....	37
Figura 3.3 – Representação de um nó da árvore de segmentação cheia usando partição flexível.....	40
Figura 3.4 – Resultados obtidos para a imagem Lena.....	42
Figura 3.5 – Resultados obtidos para a imagem Goldhill.....	43
Figura 3.6 – Resultados obtidos para a imagem PP1205.....	43

Figura 4.1 – ROBs correspondentes de duas imagens que formam o par estéreo.....	50
Figura 4.2 – Deslocamento realizado em um bloco da imagem esquerda reconstruída a fim de codificar um bloco da imagem direita pertencente à mesma ROB.....	51
Figura 4.3 – Cálculo do mapa de disparidades pixel a pixel.....	53
Figura 4.4 – Cálculo da média simples das colunas do mapa de disparidades pixel a pixel para construção do mapa de disparidades da ROB em questão.....	54
Figura 4.5 – Construção do dicionário de blocos deformados utilizando a imagem esquerda reconstruída e o mapa de disparidades.....	55
Figura 4.6 – Resultados para a imagem esquerda do par Aqua.....	58
Figura 4.7 – Resultados para a imagem esquerda do par Corridor.....	59
Figura 4.8 – Resultados para a imagem esquerda do par Man.....	59
Figura 4.9 – Resultados para a imagem esquerda do par Saxo.....	60
Figura 4.10 – Resultados para a imagem direita do par Aqua.....	60
Figura 4.11 – Resultados para a imagem direita do par Corridor.....	61
Figura 4.12 – Resultados para a imagem direita do par Man.....	61
Figura 4.13 – Resultados para a imagem direita do par Saxo.....	62
Figura 4.14 – Resultados para a média das imagens do par Aqua.....	62
Figura 4.15 – Resultados para a média das imagens do par Corridor.....	63
Figura 4.16 – Resultados para a média das imagens do par Man.....	63
Figura 4.17 – Resultados para a média das imagens do par Saxo.....	64
Figura A.1 – Aerial (512 x 512).....	69
Figura A.2 – Baboon (512 x 512).....	70
Figura A.3 – Barbara (512 x 512).....	70
Figura A.4 – Bridge (512 x 512).....	71
Figura A.5 – Gold (512 x 512).....	71
Figura A.6 – Lena (512 x 512).....	72
Figura A.7 – PP1205 (512 x 512).....	72
Figura B.1 – Aqua (360 x 288).....	73
Figura B.2 – Corridor (256 x 256).....	73
Figura B.3 – Man (384 x 384).....	74
Figura B.4 – Saxo (700 x 576).....	74
Figura C.1 – Par estéreo Aqua reconstruído.....	75
Figura C.2 – Par estéreo Corridor reconstruído.....	75

Figura C.3 – Par estéreo Man reconstruído.....	76
Figura C.4 – Par estéreo Saxo reconstruído.....	76

# Capítulo 1

## Introdução

### 1.1 Tema

O presente trabalho encontra-se inserido na área de processamento digital de imagens, mais precisamente na codificação e decodificação de imagens estéreo. Pretende-se, através dele, propor um método eficaz de compressão para tal tipo de imagens, utilizando uma nova classe de algoritmos baseada no casamento aproximado de padrões recorrentes com escala.

### 1.2 Delimitação

Este projeto é considerado útil para profissionais relacionados à área de processamento de imagens. A codificação/decodificação de imagens estéreo é um tema que vem ganhando importância cada vez maior devido às suas inúmeras aplicações, explicadas pela percepção de profundidade fornecida por esses sistemas, capazes de dar mais realismo ao conteúdo apresentado.

Entre tais aplicações destacam-se: cinema 3D, operações cirúrgicas à distância e vídeo conferência.

### 1.3 Justificativa

A geração e utilização de forma cada vez maior de informações em formato digital levaram à necessidade da criação de algoritmos de compressão que sejam

capazes de comprimir esses dados sem comprometer de maneira significativa a qualidade do conteúdo a ser armazenado ou transmitido.

No caso de dados multimídia (imagens, vídeos ou áudio), o espaço e a banda utilizados são muito grandes, principalmente no que se trata de informações visuais, o que ressalta ainda mais a importância do desenvolvimento de técnicas de compressão que sejam eficazes e adequadas aos dados em questão.

Dentro do contexto de informações visuais, destacam-se as imagens estéreo, na medida em que estas pressupõem no mínimo o dobro de informação a ser armazenada ou transmitida em relação às imagens convencionais, provocando a necessidade do desenvolvimento de técnicas que comprimam eficientemente esse tipo de imagem.

O uso de imagens estéreo leva a ferramentas poderosas, uma vez que elas são capazes de trazer mais realismo devido a percepção de profundidade fornecida por esses sistemas. No entanto, seu uso irrestrito ainda não é possível devido a alguns desafios que precisam ser superados: utilização de um display especial que dispense o uso de incômodos óculos e a sua compressão eficaz, visto que armazenam uma grande quantidade de dados.

Esse projeto, portanto, pretende aplicar uma nova classe de algoritmos chamada *Multidimensional Multiscale Parser* (MMP) na codificação de imagens estéreo.

O MMP, ao contrário dos algoritmos de compressão mais usados atualmente, não se baseia em transformadas e sim no casamento aproximado de padrões recorrentes com escala. Assim, o MMP codifica a imagem sem levar em conta nenhuma premissa sobre o seu conteúdo, ou seja, o processo de compressão independe das características da imagem, o que o difere dos demais métodos baseados em transformadas, os quais pressupõem que as imagens processadas são suaves, isto é, a maior parte da sua energia está concentrada em frequências mais baixas e, conseqüentemente, possuem baixo conteúdo em altas frequências, o que pode limitar a sua utilização.

Como consequência desse cenário, surgiu a necessidade de se desenvolver o MMP Estéreo. Com isso, pretende-se comprimir de forma eficiente esse tipo de imagem a partir de uma nova classe de algoritmos.

## 1.4 Objetivos

O objetivo deste trabalho é utilizar o MMP para comprimir imagens estéreo, explorando de forma eficiente as características desse tipo de imagem, resultando em uma melhor compressão. Uma vez que o MMP utiliza padrões da imagem previamente codificados para codificar os demais, parece ser uma boa idéia usá-lo para processar imagens estéreo, pois os padrões aprendidos no processo de compressão da imagem de referência podem ser usados para codificar a outra imagem do par.

Em outras palavras, pretende-se melhorar as taxas de compressão utilizando elementos previamente codificados da imagem de referência (esquerda) para codificar os blocos da outra imagem (direita). Dessa forma, torna-se possível explorar as redundâncias do par estéreo a fim de atingir melhores resultados.

Portanto, a partir deste trabalho, pretende-se melhorar o desempenho do MMP na compressão de imagens estéreo, adaptando-o às características das mesmas.

## 1.5 Metodologia

Como já exposto, o MMP representa uma nova classe de algoritmos que utiliza concatenações, contrações e dilatações de blocos já codificados da imagem. O MMP faz uso de um dicionário adaptativo, o qual é atualizado à medida que a imagem vai sendo codificada.

O que se pretende agora é melhorar o desempenho de um codificador existente para compressão de imagens estéreo usando o MMP [1]. A partir desse momento, serão descritas as técnicas aplicadas para tornar tal algoritmo eficaz na codificação desse tipo de imagens.

Uma vez que as imagens estéreo são captadas por lentes que, em geral, se encontram próximas e na mesma linha epipolar, pode-se concluir que o par de imagens possui uma grande redundância entre si. Assim, espera-se que uma imagem seja uma espécie de versão deslocada da outra.

Dessa forma, uma primeira idéia é utilizar os elementos codificados de uma imagem tida como referência para codificar a imagem restante. Para tal, é criado um novo dicionário formado por elementos deslocados entre blocos consecutivos de uma determinada linha da imagem de referência a fim de usá-los para codificar os blocos da linha correspondente da outra imagem. Com a inclusão desses elementos, pretende-se enriquecer o dicionário com padrões que efetivamente representem a outra imagem, evitando, assim, a divisão dos blocos, o que resulta em uma melhora na taxa de compressão, na medida em que menos *flags* e índices são transmitidos.

O problema surge quando as distâncias do objeto a cada uma das lentes não são aproximadamente iguais. Nesse caso, torna-se necessário o uso de outra técnica, uma vez que muitas vezes passa-se a não conseguir um bom casamento com os blocos deslocados. O que se faz para corrigir o problema é calcular um mapa de disparidades de duas linhas correspondentes das imagens para estimar os blocos da próxima linha da outra imagem (a que não é a de referência). A estimação é feita com o objetivo de não se transmitir o mapa, o que implicaria em um aumento da taxa, prejudicando o processo de compressão.

Os novos elementos criados a partir do mapa de disparidades e da imagem de referência são inseridos em um novo dicionário. Esses blocos recebem o nome de elementos deformados.

Assim, a criação de dois novos dicionários (deslocados e deformados) possibilita a utilização de elementos que efetivamente representam a outra imagem do par, provocando uma melhora na taxa de compressão, uma vez que se explorarão as redundâncias entre as imagens estéreo. Vale lembrar que a imagem de referência é codificada utilizando as técnicas do MMP convencional que se baseia nas redundâncias presentes dentro da própria imagem para comprimi-la.

## **1.6 Descrição**

O texto, a partir deste momento, encontra-se dividido em quatro capítulos.



No capítulo 2, será feita a descrição detalhada do algoritmo MMP original. Assim, serão expostas as técnicas utilizadas por essa nova classe de algoritmos responsáveis por viabilizar o processo de codificação e decodificação, bem como a compressão eficiente de imagens. Ao final desse capítulo, serão apresentados os resultados obtidos através da codificação de imagens com a versão original do MMP.

No capítulo 3, serão descritas as melhorias incorporadas ao algoritmo MMP original com o objetivo de aprimorar o processo de codificação, permitindo a reconstrução de imagens com boa qualidade a taxas mais baixas.

O capítulo 4 está centrado no MMP Estéreo. Dessa forma, será explicado o seu funcionamento completo, descrevendo os processos de codificação e decodificação executados para cada uma das imagens (esquerda e direita), bem como as técnicas usadas no processamento de cada uma delas a fim de permitir a exploração das redundâncias existentes dentro das mesmas, bem como daquelas presentes entre as imagens que compõem o par.

No capítulo 5, encerra-se o trabalho com as conclusões a respeito das técnicas adotadas e dos resultados obtidos utilizando-as.

# Capítulo 2

## O algoritmo MMP

### Introdução

Neste capítulo será feita uma descrição detalhada do algoritmo *Multidimensional Multiscale Parser* (MMP) [3].

O MMP representa uma nova classe de algoritmos de compressão com perdas baseada no casamento aproximado de padrões recorrentes com escala. Nele, o sinal de entrada é codificado através do casamento com elementos pertencentes a um dicionário seguindo um critério de controle, através do qual é determinado se o bloco será particionado ou não.

O dicionário é atualizado à medida que o sinal é processado sendo, portanto, adaptativo. O critério de controle pode ainda ser ajustado de forma a permitir uma compressão sem perdas.

### 2.1 Descrição do MMP

Ao processar um sinal de entrada, o MMP verifica qual elemento do dicionário representa uma melhor aproximação para o sinal em questão seguindo um critério de escolha.

O critério utilizado neste projeto representa uma relação entre taxa e distorção, na qual o custo de codificação do elemento do dicionário ( $J$ ) é dado pela soma da sua distorção em relação ao sinal de entrada ( $D$ ) com o produto de uma constante ( $\lambda$ ) pela taxa de bits gasta para codificar esse elemento do dicionário ( $R$ ). Dessa forma, o custo de codificação de um dado elemento do dicionário obedece à equação:

$$J = D + \lambda R;$$

A constante  $\lambda$  é um parâmetro de entrada sendo, portanto, definida no início do processo de codificação. Valores elevados de  $\lambda$  priorizam a taxa em detrimento da distorção, enquanto pequenos valores de  $\lambda$  funcionam de maneira contrária, priorizando a distorção em detrimento da taxa.

O MMP utiliza um dicionário constituído de elementos de diferentes dimensões, alocados em diferentes escalas ou níveis. Em cada escala são armazenados elementos de mesmo tamanho.

À medida que o sinal de entrada vai sendo processado, o dicionário é atualizado com concatenações, contrações e expansões de elementos previamente codificados. Estes novos elementos são inseridos na sua correspondente escala, a qual é definida de acordo com a dimensão de seus elementos.

Esses novos vetores constituem representações mais próximas de padrões presentes nas partes do sinal já codificadas. Por esse motivo, dizemos que o dicionário é adaptativo, na medida em que ele é capaz de “aprender” com os padrões já processados do sinal de entrada, permitindo o aproveitamento desses novos elementos na codificação de outras partes do mesmo sinal, explorando, assim, as redundâncias presentes no mesmo.

A representação do sinal por elementos de tamanho variável é uma das principais características deste algoritmo. A princípio, escolhem-se elementos do dicionário com dimensões maiores para representar o sinal de entrada. Esses elementos, no entanto, podem não atender ao requisito mínimo de qualidade. Quando isso ocorre, tenta-se representar o sinal de entrada por elementos de dimensões menores.

Uma das grandes vantagens deste algoritmo é o fato de que tanto o codificador como o decodificador são capazes de construir o mesmo dicionário, não havendo a necessidade de transmiti-lo, o que causaria um aumento da taxa. Dessa forma, a única informação a ser transmitida ao decodificador é uma seqüência de bits (*bitstream*), indicando se houve ou não divisões na representação daquele sinal e o índice que representa o elemento do dicionário escolhido para o casamento aproximado.

## 2.2 Aplicação do algoritmo MMP às imagens

### 2.2.1 O processo de codificação

Inicialmente, a imagem é dividida em blocos de tamanho  $L_b \times C_b$ . Cada bloco ( $L_b \times C_b$ ) é processado sequencialmente da esquerda para a direita até que a imagem seja totalmente codificada. A figura 2.1 [5] ilustra a divisão de uma imagem em blocos de mesma dimensão e a captação do primeiro bloco a ser processado.

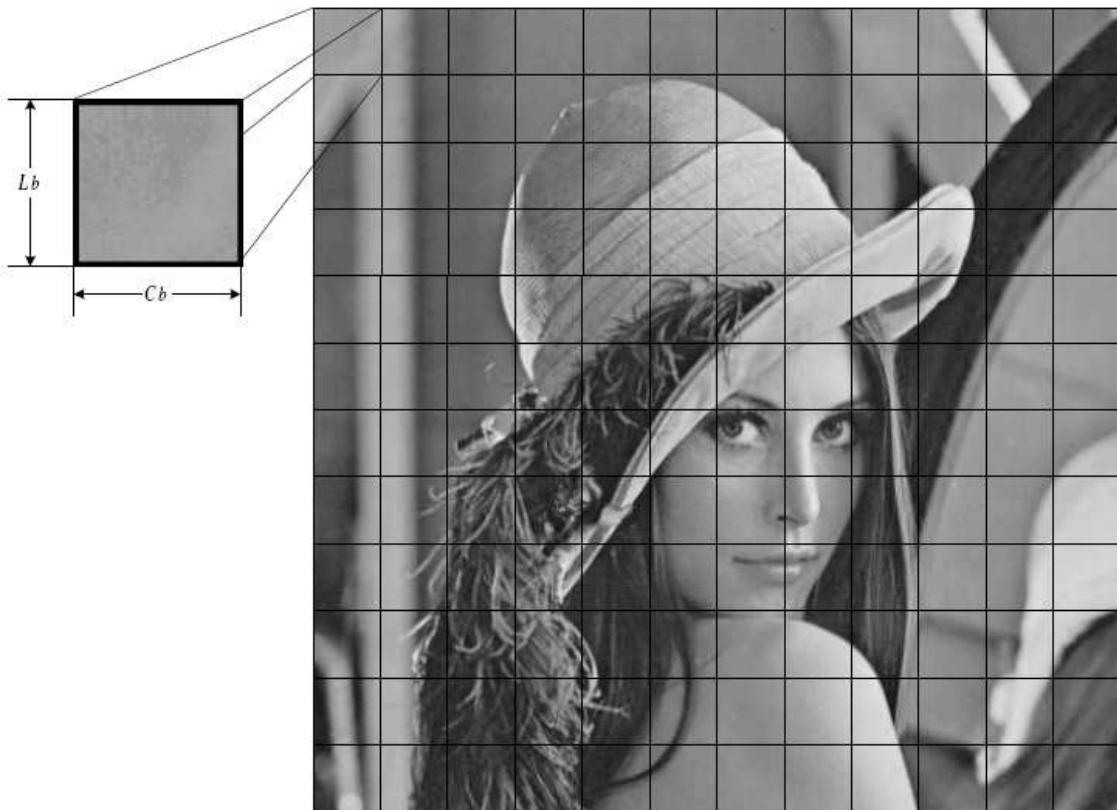


Figura 2.1 - Esquema de codificação do MMP.

Fonte: JÚNIOR, W. S. S., “Compressão de Imagens Utilizando Recorrência de Padrões Multiescalas com Segmentação Flexível”, COPPE/UFRJ, Dezembro 2004.

O dicionário é formado por vários níveis. O número de níveis está relacionado ao tamanho dos blocos utilizados para dividir a imagem e ao número de divisões que podem ser realizadas de acordo com uma árvore binária.

A figura 2.2 [1] ilustra a estrutura em árvore de um bloco 8 x 8, partindo de uma segmentação inicial vertical. A segmentação inicial, entretanto, pode também ser feita na direção horizontal. Uma vez estabelecida a direção de segmentação inicial, prossegue-se dividindo o bloco em direções alternadas (vertical seguida de horizontal ou vice-versa) até a chegada ao nível 0, constituído de elementos de tamanho 1 x 1, ou seja, um único pixel.

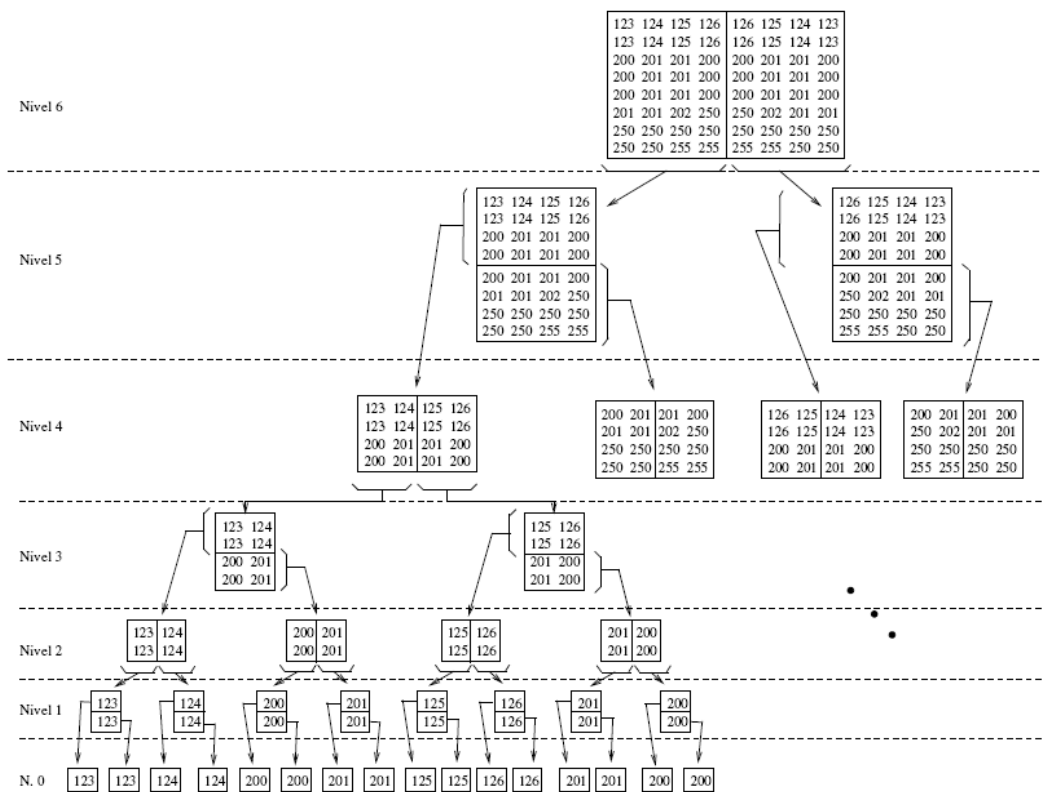


Figura 2.2 - Diagrama de uma árvore binária que representa um bloco 8 x 8 e suas possíveis divisões, partindo de uma segmentação vertical.

Fonte: DUARTE, M. H. V., “Codificação de Imagens Estéreo Usando Recorrência de Padrões”, COPPE/UFRJ, Agosto 2003.

Antes de começar o processamento da imagem, o MMP constrói um dicionário inicial a fim de permitir o início do processo de codificação. Os elementos do dicionário são gerados partindo de um valor mínimo (*min\_pixel*), o qual é incrementado de um valor fixo chamado *step* até que se atinja o valor máximo (*max\_pixel*). Os valores

$min\_pixel$  e  $max\_pixel$  representam, respectivamente, os valores de luminância mínimo e máximo que podem ser utilizados.

As figuras 2.3 e 2.4 ilustram, respectivamente, os elementos de índice 0 e 4 pertencentes ao nível 4 do dicionário inicial.

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Figura 2.3 - Elemento de índice 0 do nível 4 do dicionário inicial.

4	4	4	4
4	4	4	4
4	4	4	4
4	4	4	4

Figura 2.4 - Elemento de índice 4 do nível 4 do dicionário inicial.

Construído o dicionário inicial, tem início o processo de codificação. Dado o primeiro bloco de tamanho  $L_b \times C_b$ , o algoritmo procura no nível mais alto do dicionário o elemento que permite o melhor casamento segundo o critério de otimização adotado. Neste projeto, o critério adotado é uma relação entre taxa e distorção dada pela equação:

$$J = D + \lambda R, \text{ onde: } J = \text{custo de codificação do elemento.}$$

$D$  = distorção entre o elemento do dicionário e o bloco que está sendo codificado.

$\lambda$  = constante (multiplicador de Lagrange).

$R$  = taxa de bits.

A distorção é obtida calculando-se o erro quadrático entre o elemento do dicionário e o bloco da imagem. A taxa corresponde ao número de bits gastos para representar o índice do vetor no dicionário e o *flag* que indica se houve ou não divisão na árvore.

O início da busca pelo elemento que permite o melhor casamento começa no nível mais alto do dicionário. Achado o vetor, calcula-se o custo correspondente. O algoritmo desce, então, um nível na árvore, dividindo o bloco original em dois. A segmentação inicial pode ser feita na direção vertical ou horizontal.

A busca pelo melhor casamento é feita, agora, no nível do dicionário logo abaixo, onde o número de pixels de cada elemento corresponde à metade da escala acima. O procedimento é análogo ao do nível anterior, ou seja, procura-se para a primeira metade o elemento desse nível que gera o menor custo  $J$ . O mesmo é feito para a outra metade. O processo é repetido até que estejam calculados todos os custos da árvore para um determinado bloco da imagem.

Uma vez obtida a árvore cheia, inicia-se o processo de poda, isto é, a obtenção da árvore ótima que melhor representa o bloco da imagem. A podagem é feita da seguinte maneira:

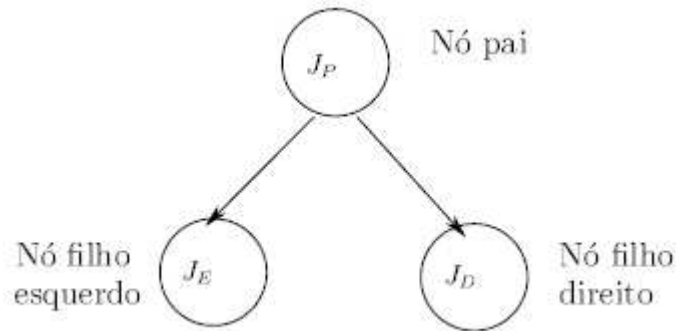


Figura 2.5 - Diagrama representando um nó de uma árvore binária e seus nós filhos. Fonte: DUARTE, M. H. V., “Codificação de Imagens Estéreo Usando Recorrência de Padrões”, COPPE/UFRJ, Agosto 2003.

Seja  $J_P$  o custo do nó pai e  $J_E$  e  $J_D$  os custos dos nós filho esquerdo e direito, respectivamente (conforme ilustrado na figura 2.5 [1]). Compara-se  $J_P$  com  $J_F = J_E + J_D + \lambda R(flag \text{ que indica segmentação})$ , onde  $J_F$  representa o custo total dos nós filho. Caso o custo do nó pai seja superior ao custo total dos nós filho ( $J_P > J_F$ ), decide-se por não podar a árvore, indicando que para este nó a divisão é a melhor escolha. Em outras palavras, a divisão significa que representar este bloco da imagem por dois blocos separadamente é melhor que representá-lo por um bloco maior. Decidido que a segmentação é a melhor opção, atribui-se um  $flag = 1$  ao nó pai e atualiza-se o seu custo, substituindo-o por  $J_F$ .

Caso contrário ( $J_P \leq J_F$ ), a árvore será podada, indicando que não é necessário dividir o nó, na medida em que o algoritmo já encontrou um melhor casamento usando um elemento maior do dicionário. Dessa forma, atribui-se um  $flag = 0$  ao nó pai e o índice que representa o elemento do dicionário com o qual foi obtido o casamento é guardado.

O processo de podagem prossegue partindo dos nós folha até o nó raiz e é encerrado apenas quando a árvore é completamente percorrida. O resultado é a obtenção da árvore de segmentação ótima, informando como a codificação deste bloco da imagem deve ser realizada. Um exemplo de árvore ótima, obtida com o processo de podagem, é ilustrado na figura 2.6 [1].



Uma vez obtida a árvore ótima de segmentação, inicia-se o processo de codificação propriamente dito. Dessa forma, percorre-se tal árvore, começando pelo nó raiz e terminando nos nós folha ou terminais. A árvore é percorrida da esquerda para direita, o que significa que o nó esquerdo é sempre o primeiro a ser analisado seguido do direito.

Toda vez que o algoritmo encontra um  $flag = 1$ , ele o envia ao decodificador e desce um nível na árvore. Ao encontrar um nó terminal, é enviado o  $flag = 0$  (indicando que não houve divisão), seguido do índice que representa o elemento do dicionário com o qual o melhor casamento foi obtido. A única situação em que o  $flag = 0$  não é enviado ocorre quando o elemento pertence ao nível 0 do dicionário. Nesse caso, o  $flag = 0$  não precisa ser transmitido, na medida em que a partição é obrigatória. *Flags* e índices são codificados com codificador aritmético.

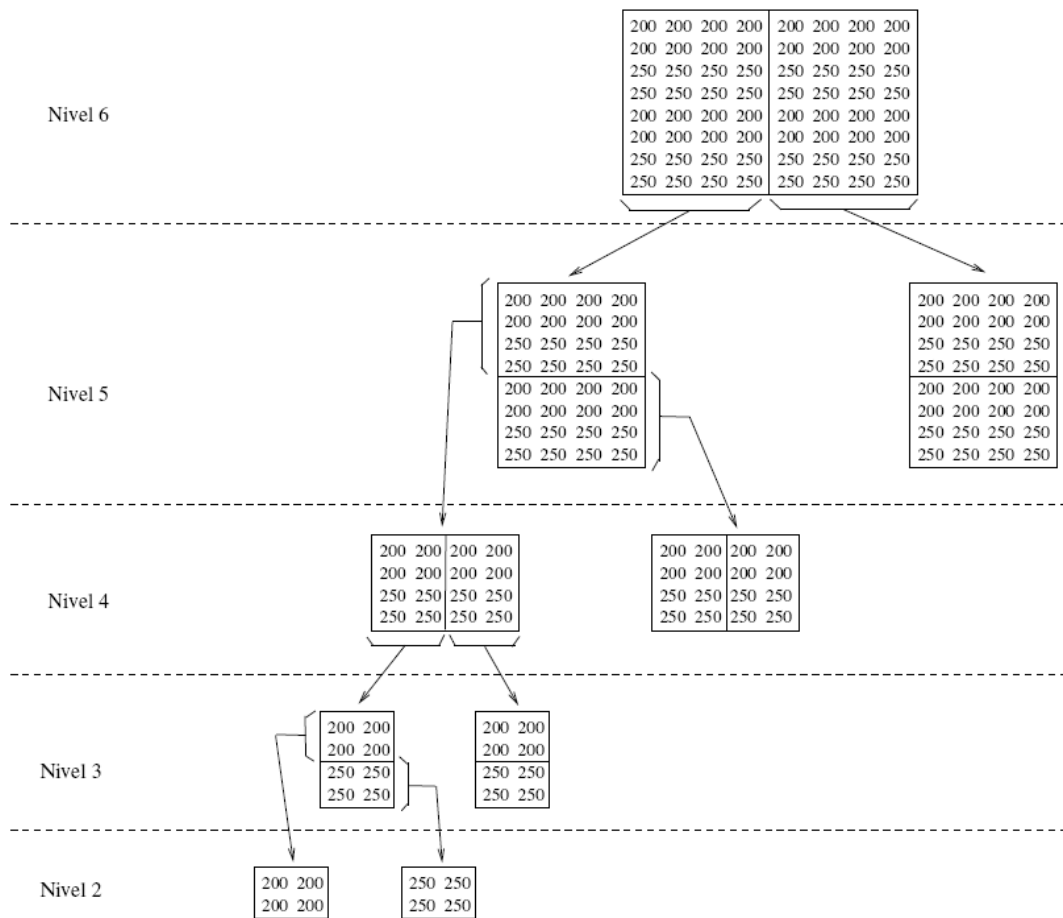


Figura 2.6 - Árvore ótima após o processo de segmentação.

Fonte: DUARTE, M. H. V., “Codificação de Imagens Estéreo Usando Recorrência de Padrões”, COPPE/UFRJ, Agosto 2003.

### **2.2.2 A atualização do dicionário**

Antes do início do processo de codificação, o dicionário é inicializado. Entretanto, o dicionário do MMP não é estático, ao contrário, é adaptativo, o que significa que novos elementos vão sendo incorporados ao longo do processo de codificação. Portanto, o dicionário “aprende” padrões que já ocorreram durante o processamento, viabilizando uma melhor compressão caso os mesmos se tornem recorrentes, explorando, dessa forma, as redundâncias existentes dentro da própria imagem.

Na árvore ótima de segmentação, sempre que se chega a dois nós folha, concatenam-se os elementos e insere-se a concatenação no nível do dicionário imediatamente superior ao dos elementos terminais. Esse novo elemento, formado pela concatenação dos blocos, é ainda contraído e expandido, gerando novos vetores que são inseridos nos seus respectivos níveis. Assim, caso tais padrões voltem a ocorrer, o dicionário conta, a partir deste momento, com elementos mais representativos para esse padrão, melhorando o processo de compressão. Em outras palavras, pode-se dizer que o dicionário sofreu um processo de aprendizado.

As contrações e expansões do bloco concatenado podem ser feitas de diversas maneiras. Exemplos de contração e expansão de um bloco são apresentados na figura 2.7 [1].

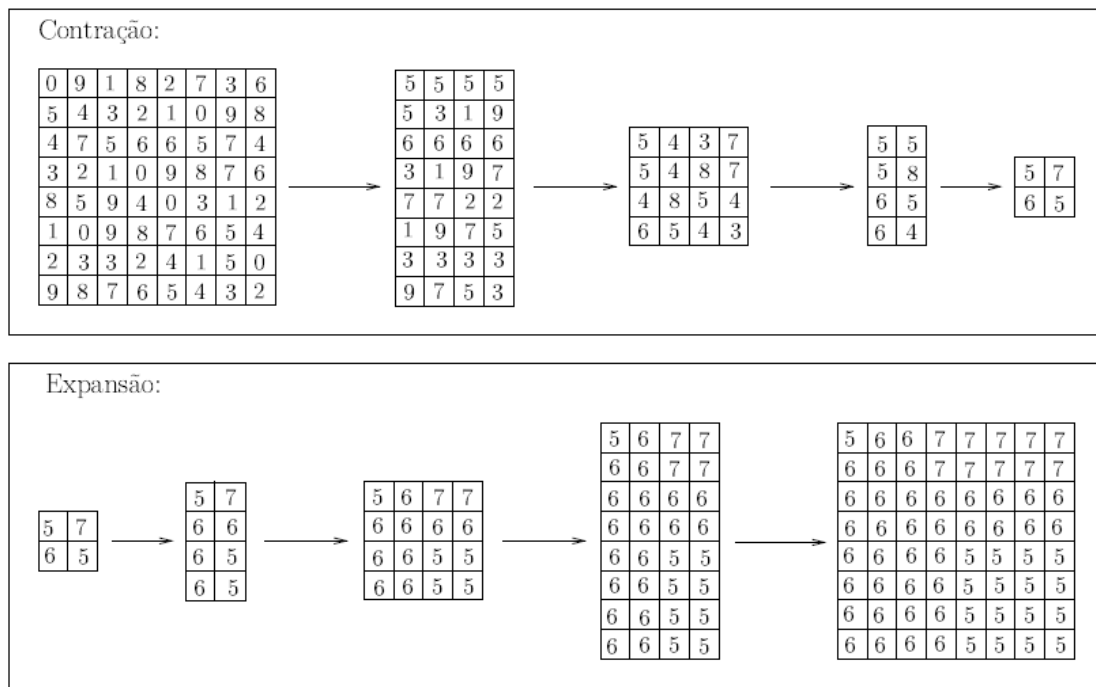


Figura 2.7 - Representação da contração e expansão de um bloco da imagem.

Fonte: DUARTE, M. H. V., “Codificação de Imagens Estéreo Usando Recorrência de Padrões”, COPPE/UFRJ, Agosto 2003.

Os blocos contraídos e expandidos são fundamentais para um aprendizado mais rápido do dicionário. O objetivo é fazer com que os níveis mais altos conttenham elementos cada vez mais representativos dos padrões que ocorrem na imagem de forma a permitir bons casamentos em níveis elevados, diminuindo a transmissão de *flags* e índices e, portanto, viabilizando a obtenção de melhores taxas de compressão.

A inserção de elementos no dicionário não é, entretanto, ilimitada. Na implementação original [3], o número máximo de elementos que cada nível pode ter é de 32760. Como o dicionário está sempre em crescimento, esse valor muitas vezes é atingido, criando a necessidade de se excluir alguns elementos do dicionário a fim de inserir novos.

Existem dois critérios básicos para decidir qual elemento deve ser excluído. O primeiro e mais usado baseia-se no número de vezes que um determinado elemento foi usado. Dessa forma, quando existe a necessidade de se excluir um vetor, exclui-se aquele menos usado. O novo bloco é inserido, então, no lugar do elemento excluído.

O outro critério utilizado baseia-se nos blocos mais usados recentemente. Assim, o vetor a ser excluído é aquele que não foi utilizado por um período mais longo de tempo. Neste projeto, o critério de exclusão adotado foi o primeiro, ou seja, o elemento menos vezes usado.

As figuras 2.8 e 2.9 [5] ilustram a geração da árvore de segmentação ótima de um bloco da imagem e a posterior atualização do dicionário.

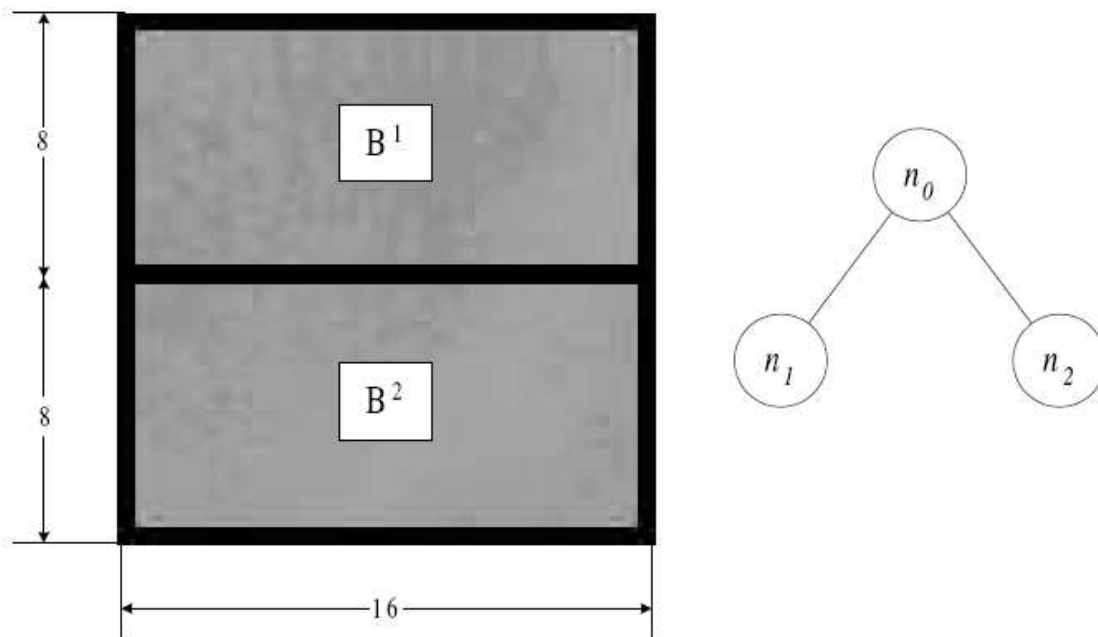


Figura 2.8(a) - Bloco segmentado horizontalmente e sua árvore de segmentação.

Fonte: JÚNIOR, W. S. S., “Compressão de Imagens Utilizando Recorrência de Padrões Multiescalas com Segmentação Flexível”, COPPE/UFRJ, Dezembro 2004.

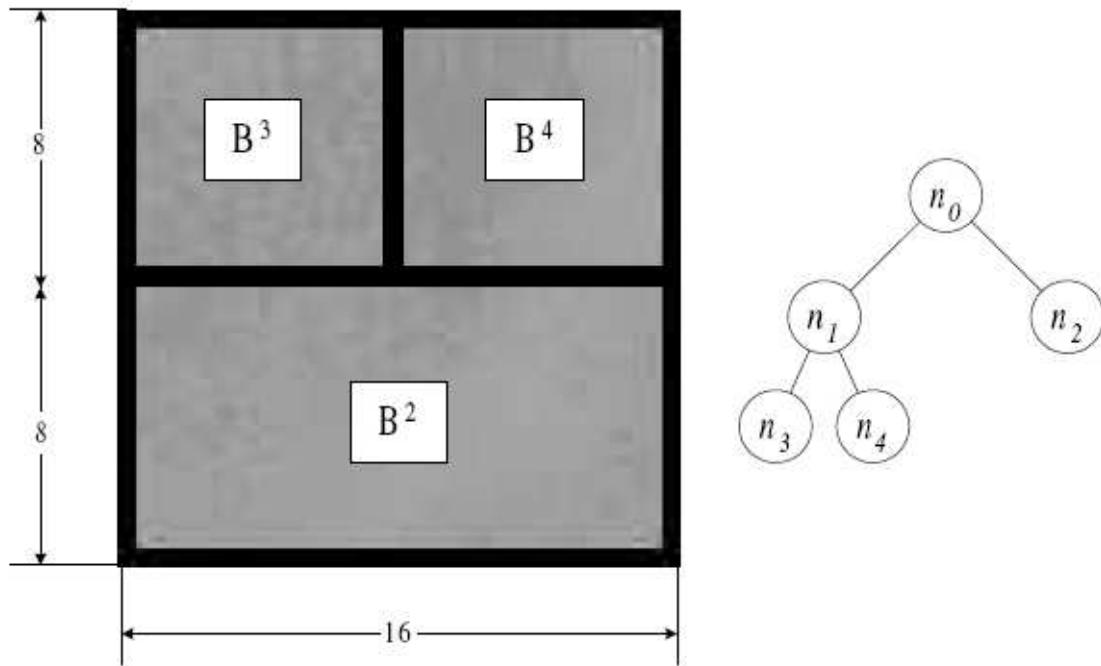


Figura 2.8 (b) - Bloco superior segmentado verticalmente e sua árvore de segmentação.  
 Fonte: JÚNIOR, W. S. S., “Compressão de Imagens Utilizando Recorrência de Padrões Multiescalas com Segmentação Flexível”, COPPE/UFRJ, Dezembro 2004.

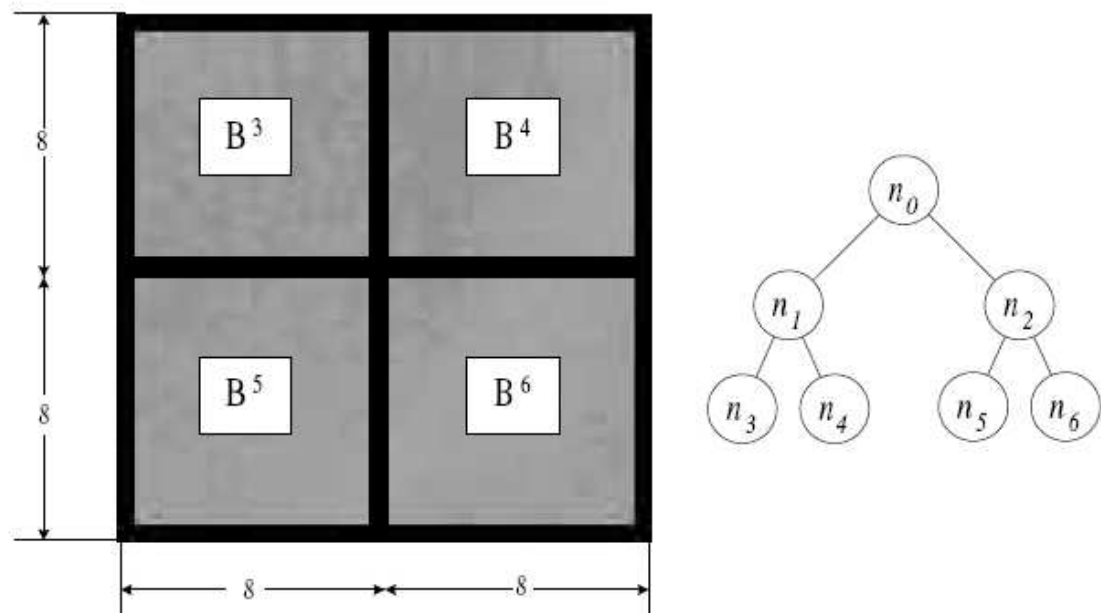


Figura 2.8 (c) - Bloco inferior segmentado verticalmente e sua árvore de segmentação.  
Fonte: JÚNIOR, W. S. S., “Compressão de Imagens Utilizando Recorrência de Padrões Multiescalas com Segmentação Flexível”, COPPE/UFRJ, Dezembro 2004.

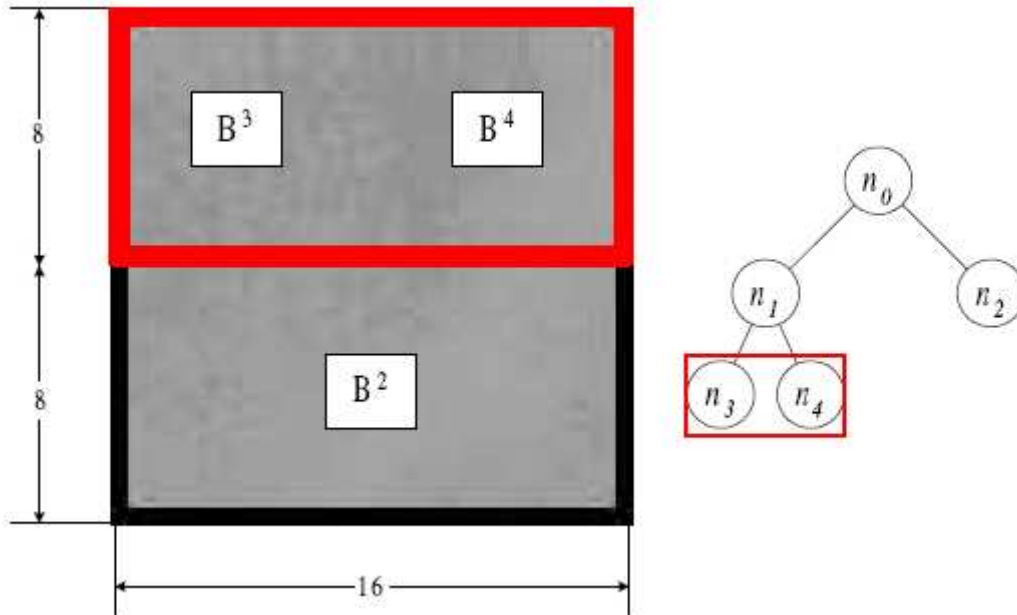


Figura 2.9 (a) - Primeira atualização do dicionário.

Fonte: JÚNIOR, W. S. S., “Compressão de Imagens Utilizando Recorrência de Padrões Multiescalas com Segmentação Flexível”, COPPE/UFRJ, Dezembro 2004.

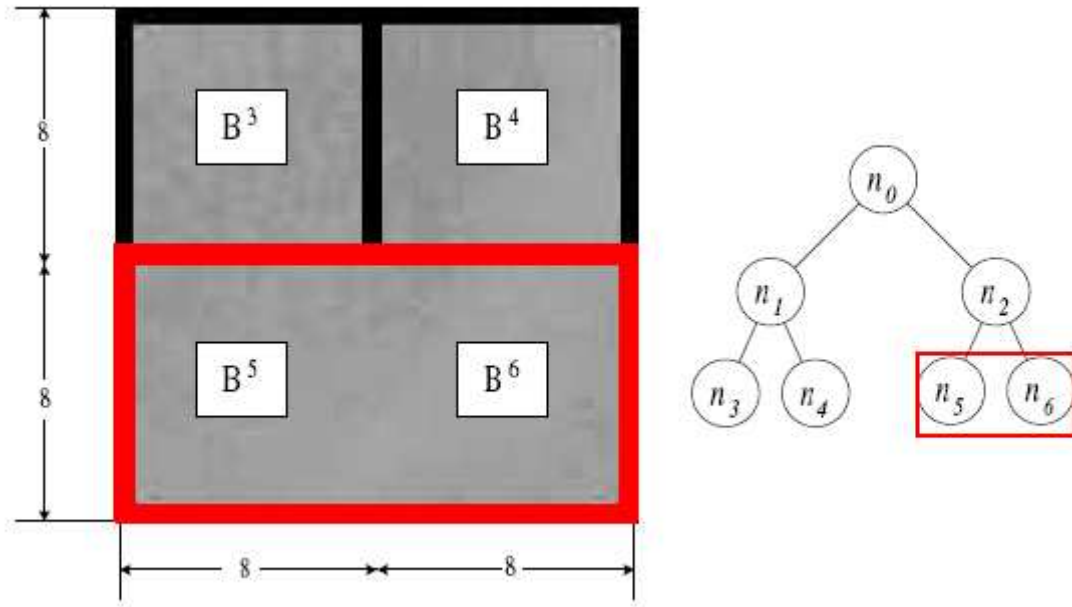


Figura 2.9 (b) - Segunda atualização do dicionário.

Fonte: JÚNIOR, W. S. S., “Compressão de Imagens Utilizando Recorrência de Padrões Multiescalas com Segmentação Flexível”, COPPE/UFRJ, Dezembro 2004.

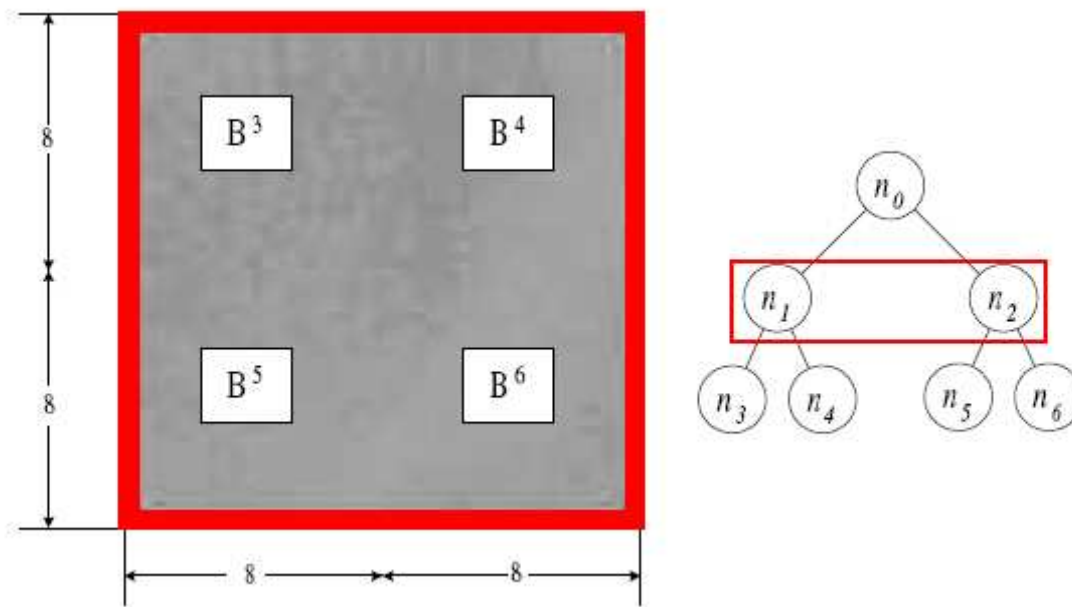


Figura 2.9 (c) - Terceira atualização do dicionário.

Fonte: JÚNIOR, W. S. S., “Compressão de Imagens Utilizando Recorrência de Padrões Multiescalas com Segmentação Flexível”, COPPE/UFRJ, Dezembro 2004.

### **2.2.3 O processo de decodificação**

Antes de começar o processo de decodificação, o decodificador inicializa o seu dicionário inicial. Os dicionários iniciais do codificador e decodificar devem ser exatamente iguais de forma que os elementos escolhidos durante os processos de codificação e decodificação sejam os mesmos. Para tal, o decodificador deve receber os valores *min\_pixel*, *max\_pixel* e *step* usados pelo codificador para construir o seu dicionário inicial.

Inicializado o dicionário, tem início o processo de decodificação. Durante esse processo, o decodificador recebe os *flags* e índices utilizados, monta a árvore de segmentação ótima do bloco que está sendo decodificado e reconstrói a imagem.

Quando dois nós terminais são detectados, o decodificador, tal como o codificador, atualiza o seu dicionário com a concatenação desses blocos e, posteriormente, com a contração e expansão do bloco concatenado. Com isso, os dicionários do codificador e decodificador se mantêm sincronizados de forma que um determinado índice represente o mesmo elemento tanto no codificador como no decodificador.

O processo de decodificação é bem mais rápido que o de codificação, na medida em que o decodificador não precisa calcular os custos de cada elemento do dicionário e realizar o procedimento de poda a fim de obter a árvore de segmentação ótima.

## **2.3 Resultados**

Nesta seção são apresentados os resultados obtidos através do processo de codificação/decodificação de imagens utilizando o algoritmo MMP original com estimação taxa distorção (2D-MMP-RD). Tais resultados podem ser observados nos gráficos das figuras 2.10 a 2.16 [3].



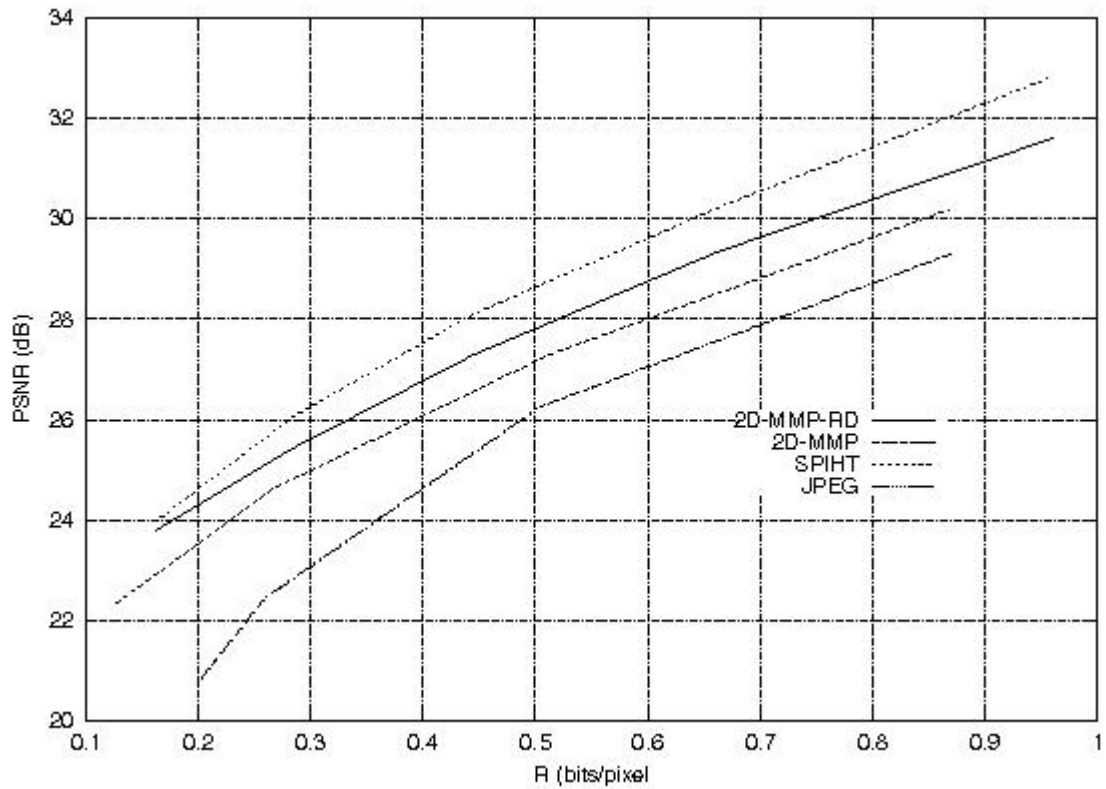


Figura 2.10 – Performance R-D com a imagem Aerial (512 x 512).

Fonte: CARVALHO, M. B., “Multidimensional Signal Compression using Multiscale Recurrent Patterns”, COPPE/UFRJ, Abril 2001.

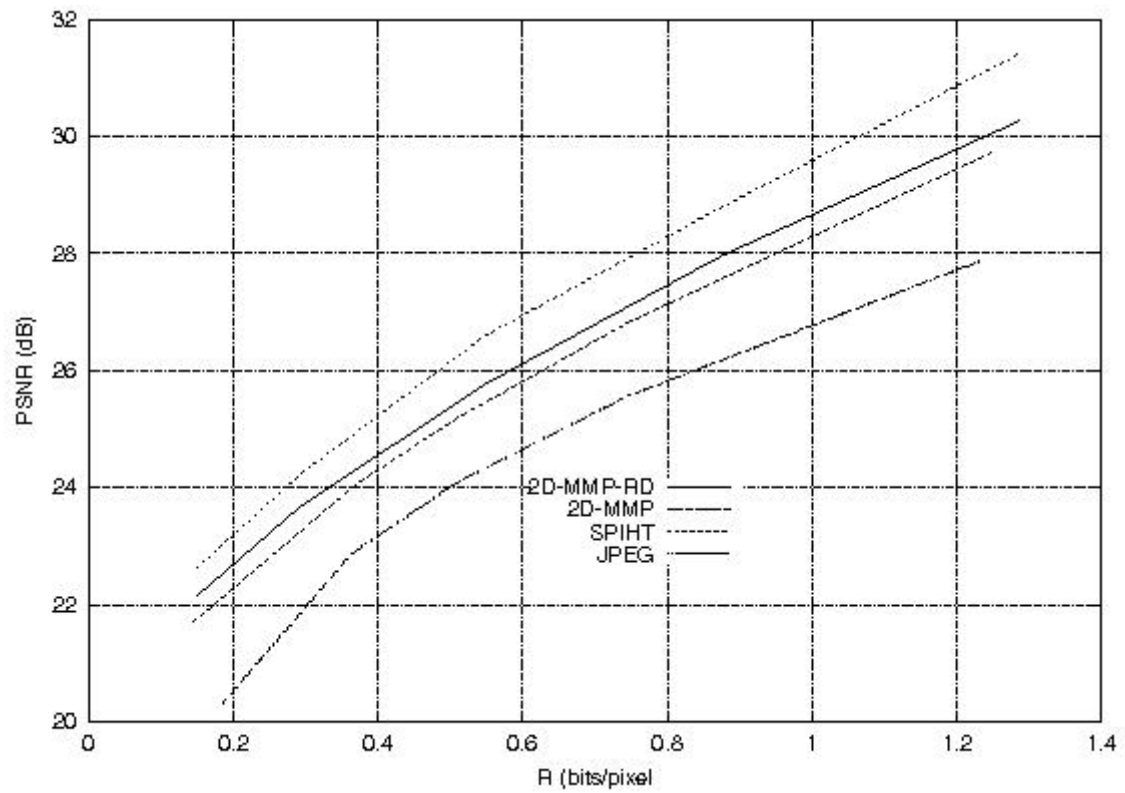


Figura 2.11 – Performance R-D com a imagem Baboon (512 x 512).

Fonte: CARVALHO, M. B., “Multidimensional Signal Compression using Multiscale Recurrent Patterns”, COPPE/UFRJ, Abril 2001.

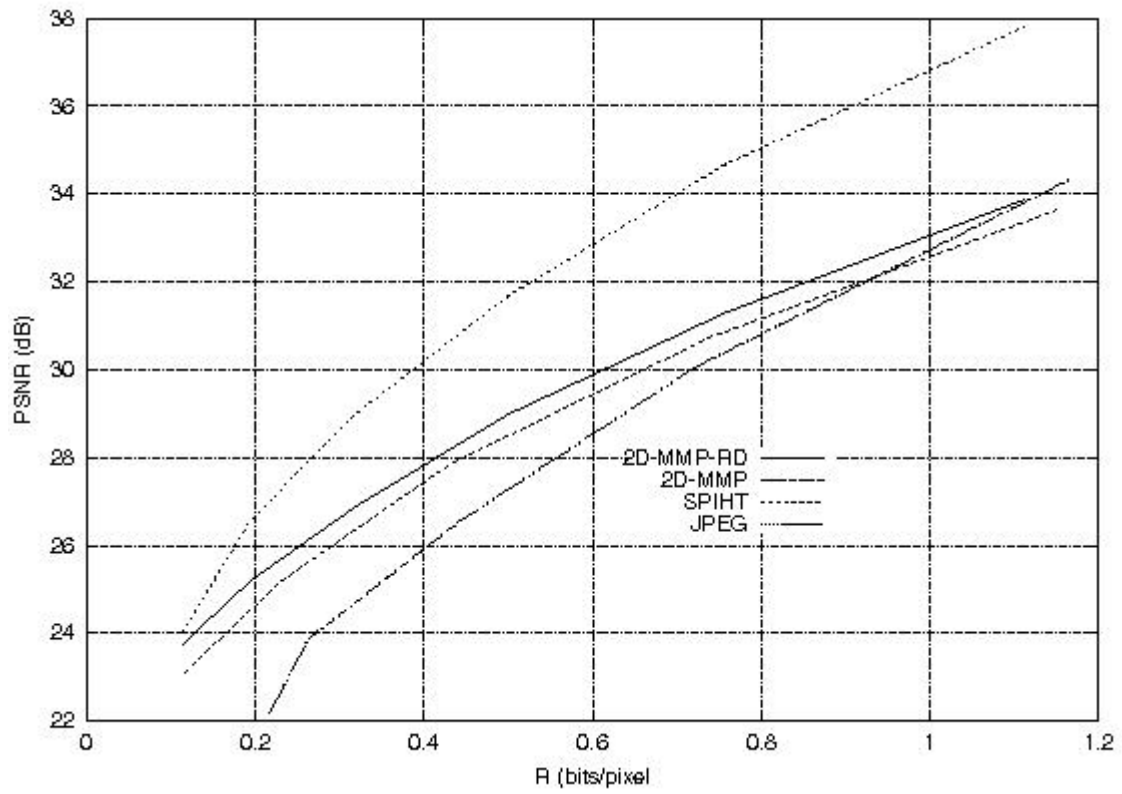


Figura 2.12 – Performance R-D com a imagem Barbara (512 x 512).

Fonte: CARVALHO, M. B., “Multidimensional Signal Compression using Multiscale Recurrent Patterns”, COPPE/UFRJ, Abril 2001.

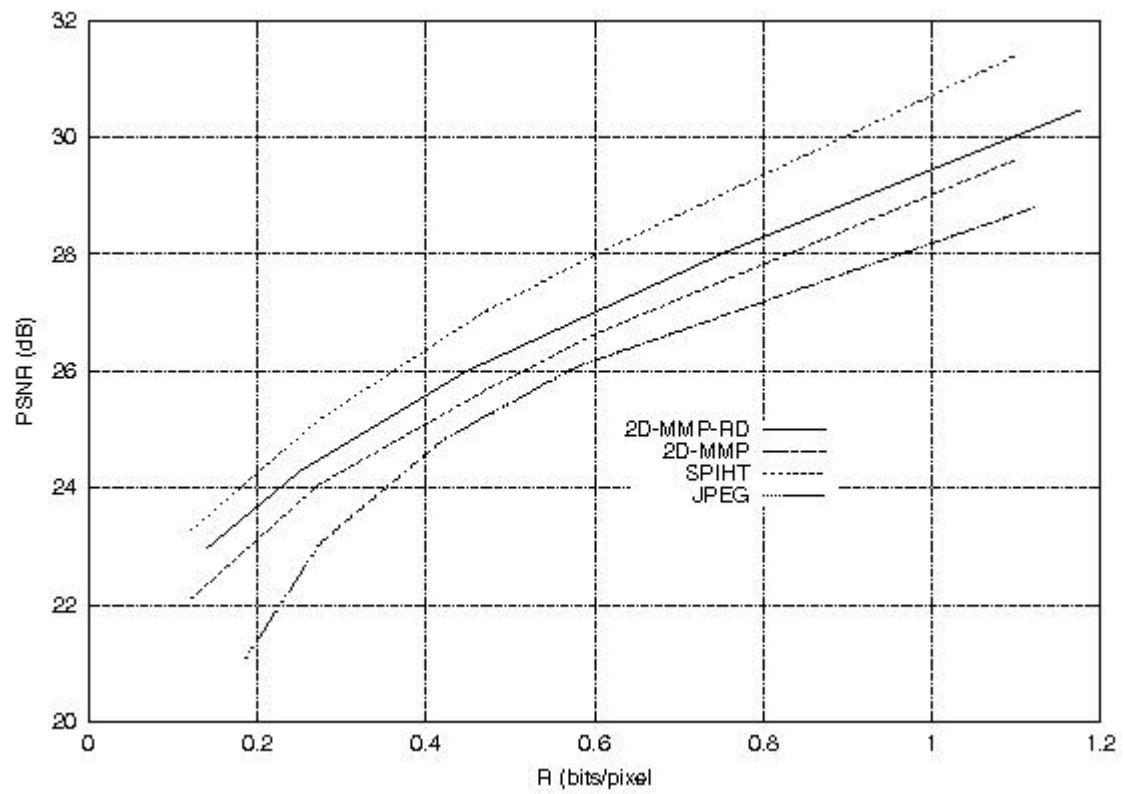


Figura 2.13 – Performance R-D com a imagem Bridge (512 x 512).

Fonte: CARVALHO, M. B., “Multidimensional Signal Compression using Multiscale Recurrent Patterns”, COPPE/UFRJ, Abril 2001.

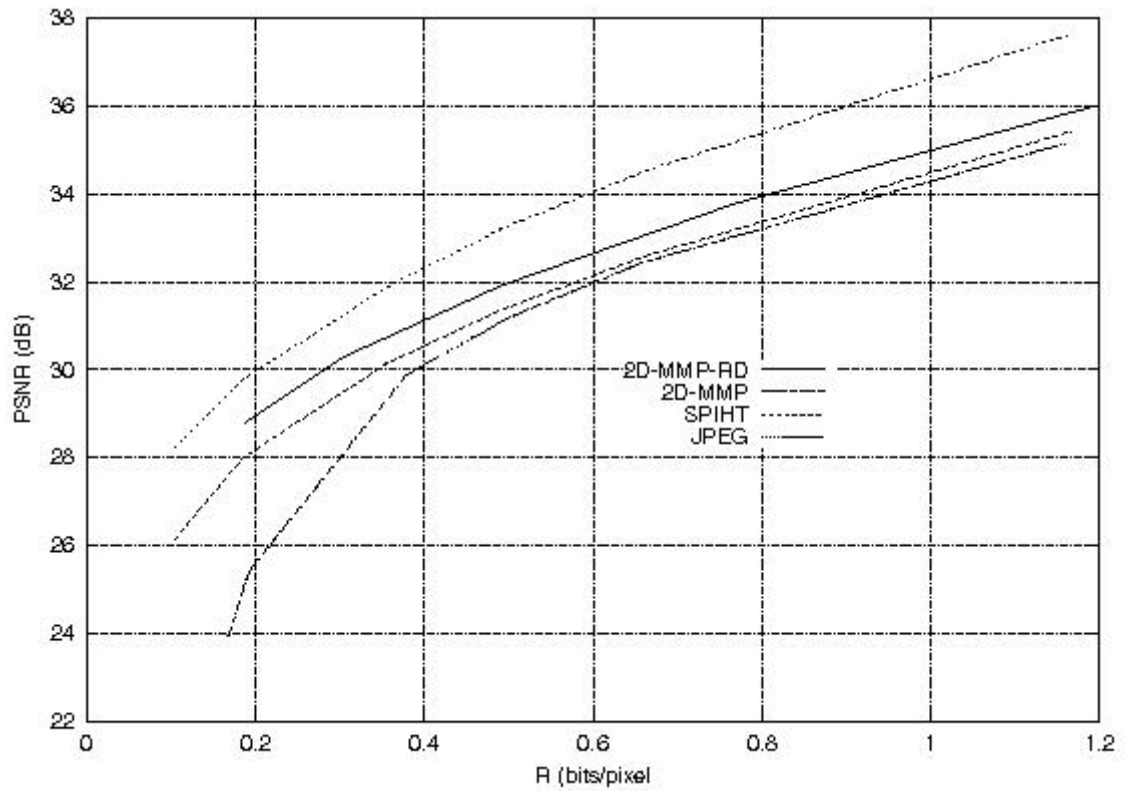


Figura 2.14 – Performance R-D com a imagem Gold (512 x 512).

Fonte: CARVALHO, M. B., “Multidimensional Signal Compression using Multiscale Recurrent Patterns”, COPPE/UFRJ, Abril 2001.

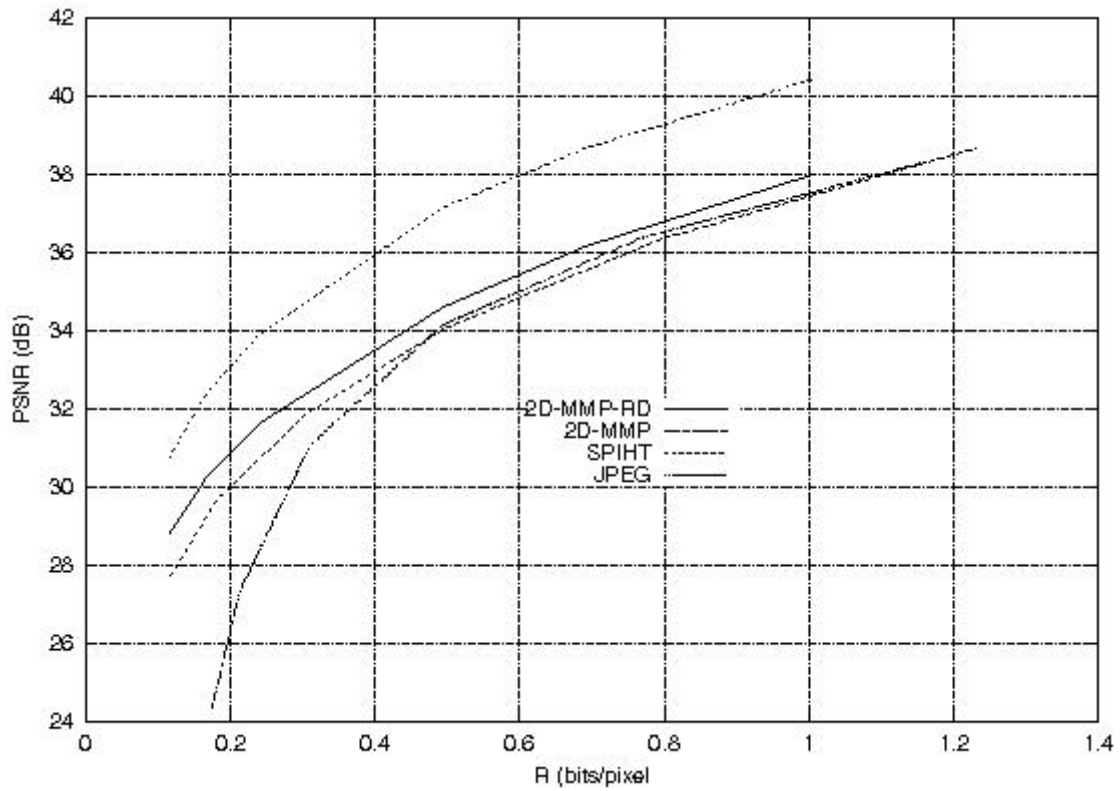


Figura 2.15 – Performance R-D com a imagem Lena (512 x 512).

Fonte: CARVALHO, M. B., “Multidimensional Signal Compression using Multiscale Recurrent Patterns”, COPPE/UFRJ, Abril 2001.

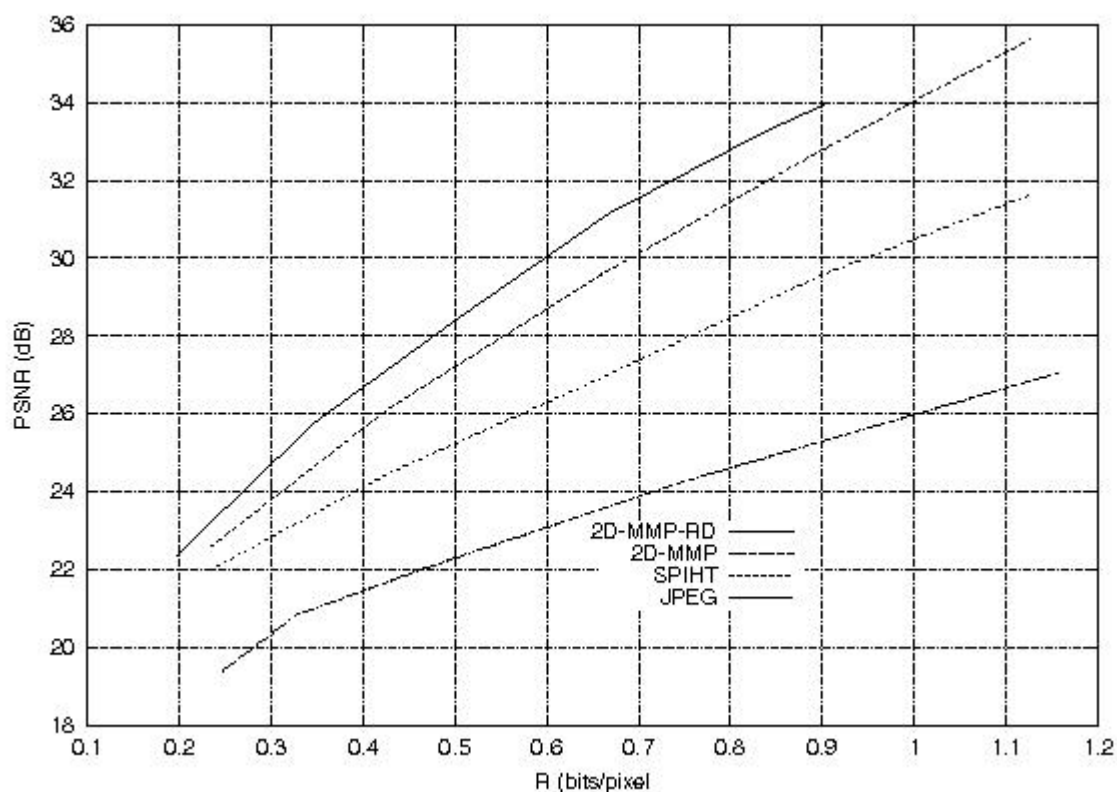


Figura 2.16 – Performance R-D com a imagem PP1205 (512 x 512).

Fonte: CARVALHO, M. B., “Multidimensional Signal Compression using Multiscale Recurrent Patterns”, COPPE/UFRJ, Abril 2001.

## 2.4 Conclusão

Neste capítulo foi feita uma descrição detalhada do MMP original (2D-MMP-RD) e sua aplicação no processamento de imagens.

A descrição deste algoritmo se fez necessária por se tratar de uma nova proposta de codificação. O MMP apresenta algumas vantagens em relação aos métodos já conhecidos de codificação de imagens, como aqueles baseados no padrão JPEG. Dentre elas podemos citar: o uso automático de blocos de dimensões variáveis e a flexibilidade de definição do tamanho inicial dos blocos.

Com este capítulo encerra-se a descrição do algoritmo MMP original, concluindo-se, com base nos resultados obtidos, que ele representa uma nova e promissora classe de algoritmos de codificação de imagens. No próximo capítulo, serão

descritas as melhorias adicionadas a este algoritmo, as quais possibilitaram um grande avanço no processo de codificação.



# Capítulo 3

## Melhorias adicionadas ao algoritmo MMP

### Introdução

Neste capítulo será feita a descrição detalhada das melhorias adicionadas ao algoritmo MMP original com o objetivo de aprimorar o processo de codificação. No final do mesmo, serão apresentados os resultados obtidos com a adoção de tais técnicas.

### 3.1 Uso de técnicas de predição

Técnicas de codificação que utilizam predição apresentam a característica de gerar resíduos cuja função de distribuição de probabilidade apresenta picos em torno de zero.

Testes experimentais indicaram uma melhora no desempenho do MMP para sinais de entrada cujas funções de distribuição de probabilidade apresentam picos. Isso ocorre, porque as distribuições deste tipo tendem a gerar blocos mais uniformes, favorecendo o uso de elementos de dimensões maiores, além de aumentar a probabilidade de que os novos elementos inseridos no dicionário durante o procedimento de atualização sejam usados.

Esses resultados revelam a vantagem de se trabalhar com sinais de entrada cujas funções de probabilidade apresentem picos. Uma forma eficiente de gerar tais sinais é a adoção de técnicas de codificação que utilizam predição. Seguindo essa linha, o algoritmo foi modificado de forma que passasse a usar tais métodos para gerar blocos de

resíduo que seriam codificados com o MMP. As técnicas de predição adotadas são inspiradas naquelas usadas no padrão H.264/AVC, com exceção do modo de predição DC, substituído por um novo modelo conhecido como *Most Frequent Value* (MFV). Esse modo seleciona entre os pixels vizinhos aquele que ocorre com maior frequência.

### 3.1.1 O algoritmo MMP com uso de predição

O algoritmo MMP-I [4] utiliza os pixels vizinhos de blocos já codificados para gerar os blocos de predição que, por sua vez, irão determinar os resíduos que serão processados. Os pixels utilizados para gerar esses blocos estão localizados acima e/ou à esquerda do bloco em questão.

Assim como no padrão H.264/AVC, o MMP-I utiliza diversos modos de predição. Ao codificar um bloco da imagem, o MMP-I escolhe dentre os modos de predição disponíveis aquele que atingiu o melhor compromisso RD (taxa x distorção). O modo escolhido é, então, transmitido ao decodificador através de um *flag* (*flag* de predição) codificado com codificador aritmético. Uma vez determinado o modo a ser utilizado, o decodificador é capaz de determinar o bloco de predição a ser usado. O próximo passo consiste em decodificar o bloco de resíduo. Realizada essa etapa, o decodificador está apto a reconstruir a imagem, somando os blocos de predição e resíduo.

Como no MMP os blocos da imagem original e as árvores de segmentação são percorridos da esquerda para a direita, partindo da primeira linha e descendo até a última, os pixels usados para gerar os blocos de predição já se encontram disponíveis no decodificador e, portanto, não precisam ser transmitidos.

A codificação é realizada de maneira hierárquica. Primeiramente, a predição é feita para blocos maiores. Dessa forma, seleciona-se o modo de predição mais satisfatório, isto é, aquele cujo resíduo correspondente é codificado com menor custo. Segmenta-se, então, o bloco em dois e o processo se repete para cada uma das metades, armazenando os custos obtidos em cada nó da árvore. Vale ressaltar que o cálculo dos custos passa a incluir também o custo dos *flags* de predição que são enviados ao decodificador.

O procedimento é repetido até que seja formada a árvore de segmentação cheia. O próximo passo é o processo de poda, de modo a obter a árvore de segmentação ótima. Esse procedimento é idêntico ao descrito no capítulo 2 para o algoritmo MMP original. Assim, compara-se o custo do nó pai ( $J_P$ ) com o custo total dos nós filho ( $J_F$ ). Se o custo do nó pai for superior ao custo total dos nós filho ( $J_P > J_F$ ), a árvore não é podada, caso contrário ( $J_P \leq J_F$ ), a poda é realizada.

### 3.1.2 O dicionário do algoritmo MMP-I

Diferentemente do algoritmo MMP original, o MMP-I procura um casamento aproximado entre os blocos de resíduo da imagem original e os elementos do dicionário. Tais blocos são obtidos através do cálculo da diferença entre os blocos da imagem original e os blocos de predição. No algoritmo MMP original, o casamento era feito entre os blocos da imagem e os elementos do dicionário.

As modificações adotadas trazem duas consequências diretas para o dicionário do MMP-I, tornando-o um pouco diferente daquele utilizado no MMP original:

- O novo dicionário passa a poder apresentar elementos com valores negativos. Dessa forma, a faixa de valores dos pixels dos blocos de resíduo é o dobro em relação ao algoritmo descrito no capítulo 2.
- O dicionário deve conter uma grande quantidade de elementos com valores próximos a zero. Vetores com valores afastados de zero podem ocorrer em menor quantidade. Isso porque, a tendência é que os blocos de resíduo sejam constituídos por pixels próximos a zero.

Como consequência deste cenário, foi proposta uma nova forma de inicialização do dicionário baseada não mais em um passo fixo, mas sim em um passo variável. Dessa forma, o passo (*step*) é pequeno em valores próximos de zero, tornando-se maior à medida que ocorre o afastamento em relação a esse valor. A função que descreve o passo adotado de acordo com o valor dos pixels é definida como:

$step = 2$ , se  $|pixel| \leq 10$

$step = 4$ , se  $10 < |pixel| \leq 22$

$step = 8$ , se  $22 < |pixel| \leq 86$

$step = 13$ , se  $|pixel| > 86$ .

## 3.2 Controle de redundância dos elementos do dicionário

O algoritmo MMP-I utiliza o mesmo procedimento de atualização do dicionário descrito no capítulo 2. Sempre que um bloco é inserido em um nível  $N$ , a partir da concatenação de dois elementos do nível  $N - 1$ , escala-se o bloco concatenado através de contrações e expansões, inserindo-o no nível correspondente.

Como consequência do processo de atualização, o número final de blocos em cada escala do dicionário é muito maior que o número de elementos efetivamente usados durante o processo de codificação.

Entretanto, cada vez que um novo elemento é inserido, um novo índice é necessário, aumentando a entropia média destes símbolos. Dessa forma, a introdução de novos elementos que não criam vantagens para o casamento aproximado de blocos futuros limita o desempenho do algoritmo. Com o objetivo de contornar esse problema, foi desenvolvido um método cujo objetivo é conter o crescimento excessivo do dicionário através da redução da redundância entre seus vetores.

O procedimento de controle de redundância somente permite que um novo elemento seja inserido caso a distorção entre o bloco a ser inserido e todos os elementos do dicionário seja superior a  $d^2$ . O parâmetro  $d$  é informado pelo usuário antes de se iniciar o processo de codificação e enviado ao decodificador.

Com isso, cria-se uma condição de distorção mínima entre todos os elementos pertencentes a um determinado nível do dicionário, onde cada bloco passa a ser representado por uma hipersfera de raio  $d$ , conforme pode ser observado na figura 3.1 para o caso de um espaço bidimensional.

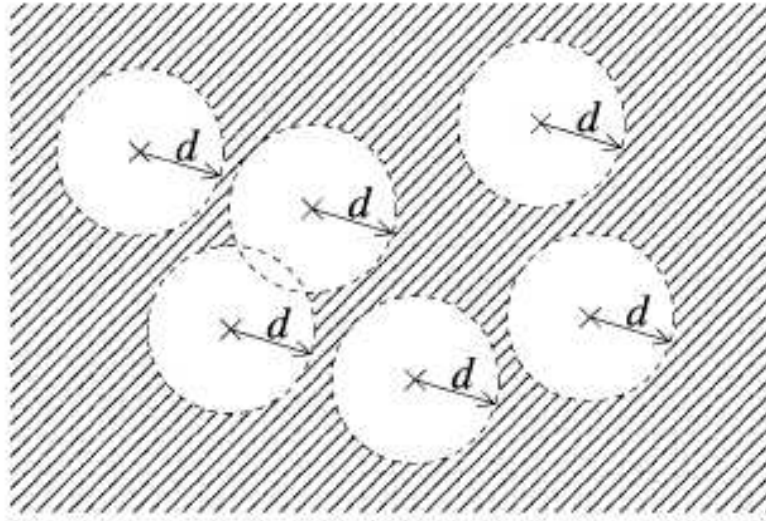


Figura 3.1 – Nova representação dos elementos do dicionário.

O parâmetro  $d$  controla a redundância entre os elementos de um determinado nível do dicionário e sua escolha deve ser feita de maneira cuidadosa. Se o raio das hiperesferas for muito pequeno, os vetores tornam-se muito próximos e o controle de redundância pode se tornar ineficaz. Por outro lado, se o parâmetro  $d$  for um número muito grande, o dicionário conterá enormes áreas vazias correspondentes a padrões que não serão representados satisfatoriamente.

Dessa forma, o valor ótimo para o parâmetro  $d$  torna-se uma função da taxa de compressão desejada. Para altas taxas, os elementos do dicionário devem ser próximos a fim de tentar representar da melhor maneira possível os diferentes padrões que ocorrem na imagem. Para taxas baixas, no entanto, os vetores do dicionário podem estar mais espaçados. Com isso, menos elementos serão inseridos e, conseqüentemente, a taxa de codificação dos índices será menor.

O problema surge pela necessidade de se conhecer *a priori* a taxa de bits, não disponível no momento de escolha do parâmetro  $d$ . Entretanto, tal fato pode ser contornado através de uma heurística que relaciona os parâmetros  $d$  e  $\lambda$ , na medida em que valores elevados de  $\lambda$  priorizam a taxa (taxas baixas) enquanto baixos valores, ao contrário, priorizam a distorção (taxas altas).

Após uma bateria de testes usando diferentes pares  $(d, \lambda)$  para um conjunto de imagens, chegou-se aos valores exibidos abaixo:

$d = 5$ , se  $\lambda \leq 15$

$d = 10$ , se  $15 < \lambda \leq 50$

$d = 20$ , para os demais casos.

### 3.3 Novas técnicas de atualização do dicionário

Algoritmos de casamento aproximado de padrões como o MMP exploram as similaridades existentes dentro de uma imagem baseando-se no fato de que um determinado bloco usado no casamento aproximado de uma determinada região apresenta uma grande probabilidade de ser novamente útil no casamento de regiões a serem codificadas.

Dessa forma, técnicas eficazes de atualização do dicionário tornam-se um ponto crucial no desempenho do MMP, na medida em que enriquecem o dicionário com padrões que efetivamente representam a imagem que está sendo processada.

No algoritmo MMP original, essas técnicas restringiam-se à concatenação, contração e expansão de blocos já codificados. Com o objetivo de aumentar a eficiência do processo de atualização, foram incorporadas técnicas que incluem novos elementos no dicionário durante esta etapa. A consequência imediata dessa nova proposta é o rápido crescimento do dicionário, necessitando, portanto, de uma seleção cuidadosa dos blocos que de fato serão inseridos. Para tal, torna-se clara a importância da estratégia de controle de redundância entre os elementos do dicionário, apresentada na seção anterior, como uma maneira de impedir a inserção de blocos que venham a comprometer o processo de codificação.

Entre as novas técnicas de atualização destacam-se: transformação geométrica e a inserção de blocos simétricos e deslocados. A transformação geométrica explora as relações espaciais entre os modos de predição. Os modos usados no algoritmo MMP-I tentam prever os padrões que ocorrem na imagem usando diferentes direções, as quais

apresentam relações espaciais entre si. As predições horizontal e vertical, por exemplo, podem ser relacionadas através do uso de rotações de  $90^\circ$ . Assim, para explorar as relações espaciais entre os modos de predição, passaram a ser inseridas quatro novas versões dos elementos do dicionário escolhidos no processo de codificação: dois rotacionados de  $90^\circ$  e  $-90^\circ$  e dois espelhados horizontal e verticalmente.

Prosseguindo com a busca de novas técnicas eficazes para a atualização do dicionário, decidiu-se por incluir ainda blocos simétricos, na medida em que estes tendem a ter a mesma estrutura direcional do bloco original, o que significa que podem ser úteis para codificar futuras regiões da imagem previstas com modos de predição similares.

O uso dos blocos deslocados, por sua vez, pode ser justificado pelo fato de que um determinado padrão em uma imagem pode recorrer em qualquer posição. Para levar isso em consideração, passou-se a incluir versões deslocadas do bloco original. O passo de deslocamento utilizado corresponde à metade ou a um quarto do tamanho do bloco em questão.

Os testes iniciais realizados com o objetivo de se verificar a eficiência das novas técnicas de atualização revelaram que, em geral, os novos elementos incluídos não introduziram melhoras no processo de codificação. Ao contrário, eles levaram a valores inferiores de PSNR para as imagens decodificadas. Isso pode ser explicado pelo crescimento da entropia dos índices, causado pela inserção dos novos blocos.

Entretanto, tais testes foram realizados sem o uso do controle de redundância entre os elementos. Testes posteriores, utilizando simultaneamente as duas técnicas, revelaram resultados satisfatórios, tornando clara a necessidade de uma seleção cuidadosa nos blocos a serem inseridos no dicionário, principalmente quando se adotam tais técnicas de atualização.

### **3.4 Divisão do dicionário em gavetas**

Outra forma de melhorar o desempenho do MMP é aumentar a eficiência no processo de codificação dos índices que representam os elementos do dicionário. O

algoritmo inicial, descrito no capítulo 2, utiliza o codificador aritmético com contexto adaptativo no processo de codificação. A definição do contexto é feita com base na escala do dicionário a que o símbolo que está sendo codificado pertence, sendo usado para a codificação dos *flags* de segmentação, modos de predição e índices. A escolha do contexto está implícita para o decodificador e, dessa forma, nenhuma informação nesse sentido precisa ser enviada.

Com o objetivo de aumentar a eficiência na codificação dos índices, foi proposto um esquema adaptativo para a escolha do contexto através da partição de cada nível do dicionário em gavetas, associando diferentes probabilidades de ocorrência a cada uma delas para cada escala do dicionário. Com isso, surgem modificações no processo de codificação, na medida em que cada elemento escolhido passa a ser codificado usando-se dois símbolos: o primeiro identifica a partição a que o elemento pertence, enquanto o segundo indica o índice do elemento dentro da partição. Dessa maneira, esse novo processo de codificação requer a transmissão de dois símbolos por elemento escolhido ao passo que o anterior, utilizado no MMP original, transmitia apenas um símbolo (índice que representava o elemento dentro do nível do dicionário).

A impressão inicial é de que aumentou a quantidade de informação necessária para codificar um determinado elemento do dicionário, o que na prática não é verdade. Isso ocorre, porque dependendo do critério usado para particionar o dicionário, pode-se conseguir com que a entropia do símbolo usado para identificar a partição a qual o elemento pertence seja bem menor que  $\lceil \log_2(N_p) \rceil$ , onde  $N_p$  corresponde ao número de partições. Portanto, utilizando esse artifício, taxas mais baixas podem ser alcançadas.

Poderia ser argumentado que a entropia de um vetor do dicionário é a mesma, independentemente do número de passos usados para codificá-lo. Entretanto, a escolha do critério de segmentação de maneira conveniente permite que o codificador aritmético se aproxime da entropia de forma muito mais rápida, justificando a adoção dessa técnica. A figura 3.2 ilustra a partição de um nível do dicionário em gavetas.



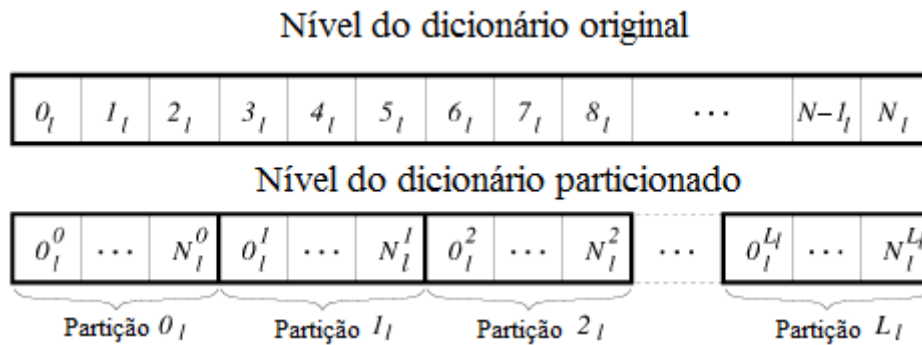


Figura 3.2 – Divisão de um nível do dicionário em gavetas.

Alguns critérios de partição do dicionário foram testados a fim de definir qual seria a melhor forma realizar a sua segmentação: modo de predição, escala original do bloco e regra de inserção do elemento no dicionário. No primeiro caso, cada partição contém elementos adicionados ao dicionário quando um determinado modo de predição foi usado. No segundo, cada gaveta contém elementos que foram criados a partir de uma escala  $N$  através da concatenação de dois elementos da escala  $N - 1$ . No terceiro e último, as partições foram criadas para distinguir os blocos criados a partir de cada um dos critérios de atualização do dicionário (blocos deslocados, rotacionados, etc.).

Comparando-se os resultados gerados pela segmentação do dicionário através de cada um dos critérios, observou-se que o segundo critério foi o que apresentou o melhor desempenho e, portanto, foi o adotado.

### 3.5 Segmentação flexível

No algoritmo MMP original, o processo de codificação iniciava-se com a divisão da imagem original em blocos de dimensão  $L_b \times C_b$ . O processamento se dava da esquerda para a direita, começando pela linha mais alta e descendo até que a imagem fosse percorrida por completo, de forma que cada bloco era codificado de maneira independente dos demais.

Uma vez retirado um bloco de tamanho  $L_b \times C_b$  da imagem original, o algoritmo procurava, no nível do dicionário correspondente, o elemento que apresentava o melhor

casamento segundo um critério de escolha, definido, neste trabalho, como uma relação entre taxa e distorção. O processamento do bloco prosseguia com a sua posterior divisão nas direções vertical ou horizontal e com a procura, na escala correspondente, do elemento que apresentava o menor custo.

A divisão inicial do bloco podia ser feita tanto na vertical como na horizontal, sendo definida através de um parâmetro informado pelo usuário antes do início da codificação. Determinada a direção inicial de segmentação, entretanto, o algoritmo processava o bloco dividindo-o em direções alternadas até a chegada ao nível mais baixo do dicionário, composto por elementos de dimensão 1 x 1, ou seja, um pixel.

Portanto, como consequência do processo de codificação, um determinado bloco em um dado momento poderia ser segmentado apenas em uma direção que dependeria apenas da escolha da direção inicial de partição.

Com a finalidade de aprimorar o processamento da imagem, foi proposta a adoção de uma segmentação flexível. Seguindo essa nova linha de pensamento, a divisão de um dado bloco seria realizada em ambas as direções (vertical e horizontal). Para cada segmentação, calcular-se-ia o custo total dos nós filho originados através de cada uma das partições e, no caso de necessidade de divisão, optar-se-ia por aquela cujos nós filho apresentaram o menor custo total ( $J_F$ ).

Dessa maneira, um determinado bloco poderia ser sempre dividido em qualquer uma das direções, excluindo-se os casos em que a dimensão do bloco não permite segmentação em uma das direções. Blocos de dimensão 4 x 1, por exemplo, não admitem segmentação vertical e, assim, em caso de necessidade de partição, são divididos na direção horizontal. Analogamente, blocos de dimensão 1 x 4, só podem sofrer segmentação vertical.

A nova proposta de segmentação traz consequências imediatas para o processo de codificação. A primeira é o aumento do número de níveis do dicionário que, a partir de então, deve ser capaz de gerar casamentos com blocos de dimensão que, outrora, nunca ocorriam durante a codificação. Supondo que sejam adotados os valores  $L_b = C_b = 8$  e definida a direção inicial de segmentação como vertical, pode-se concluir que blocos de dimensão 4 x 8 jamais ocorreriam seguindo a definição de segmentação adotada no MMP original. No entanto, quando se admite a segmentação flexível passa-

se a permitir que tanto blocos de dimensão 4 x 8 como 8 x 4 sejam codificados, surgindo a necessidade de se construir um dicionário que apresente níveis cujos elementos possuam tais dimensões.

A segunda modificação está relacionada aos *flags* que indicam se um dado bloco deve ou não ser segmentado. No MMP original, era necessário informar ao decodificador apenas se o bloco havia sido particionado ou não. A direção de partição já era conhecida, na medida em que dependia somente da direção inicial de segmentação. Assim, a única informação transmitida ao decodificador era a direção de divisão inicial. Com a adoção da segmentação flexível, torna-se necessário informar ao decodificador não apenas se o bloco foi ou não segmentado, mas em caso de divisão, deve-se indicar ainda em qual direção ela ocorreu.

Como consequência desse cenário, aumenta-se o número de valores que os *flags* de segmentação podem assumir, passando de dois (*flag* = 0, indicando que o bloco não deve ser dividido e *flag* = 1, informando a necessidade de partição) para três, conforme definido abaixo:

*Flag* = 0, indicando que o bloco não deve ser segmentado;

*Flag* = 1, indicando segmentação na direção vertical;

*Flag* = 2, indicando segmentação na direção horizontal.

A terceira e última alteração refere-se à construção da árvore de segmentação. No algoritmo original, a codificação de um bloco de tamanho  $L_b \times C_b$  iniciava-se com a escolha, na escala correspondente, do elemento que apresentava o menor custo. Posteriormente, o bloco era dividido em dois novos blocos, sendo o casamento aproximado estabelecido para cada uma das metades. O processo se repetia até que fosse formada a árvore de segmentação cheia.

O próximo passo era o procedimento de poda, no qual, partindo dos nós folha, comparava-se o custo do nó pai ( $J_P$ ) com o custo total dos nós filho ( $J_F$ ). Caso o custo do nó pai fosse menor ou igual ao custo total dos nós filho ( $J_P \leq J_F$ ), o *flag* = 0 era atribuído ao nó e a árvore era podada, pois um melhor casamento havia sido conseguido representando o bloco de entrada por um elemento de tamanho maior. Caso contrário ( $J_P > J_F$ ), o *flag* = 1 era atribuído ao nó, a árvore não era podada e o custo do nó pai era

atualizado com o custo total dos nós filho, indicando que a codificação do bloco em questão era realizada de maneira mais eficiente segmentando-o em blocos menores.

A construção da árvore de segmentação cheia usando partição flexível é análoga à do MMP original, entretanto, o algoritmo, a partir de então, deve estabelecer casamentos aproximados em ambas as direções de segmentação. Dessa maneira, o bloco em questão é, inicialmente, dividido na direção vertical. Para cada uma das metades procura-se no nível correspondente do dicionário o elemento que apresenta o menor custo e armazena-se o valor em seu devido nó. Posteriormente, divide-se novamente o bloco original, agora, na direção horizontal. O procedimento se repete para cada uma das metades, sendo os custos armazenados nos nós correspondentes. A figura 3.3 ilustra um nó da árvore de segmentação cheia:

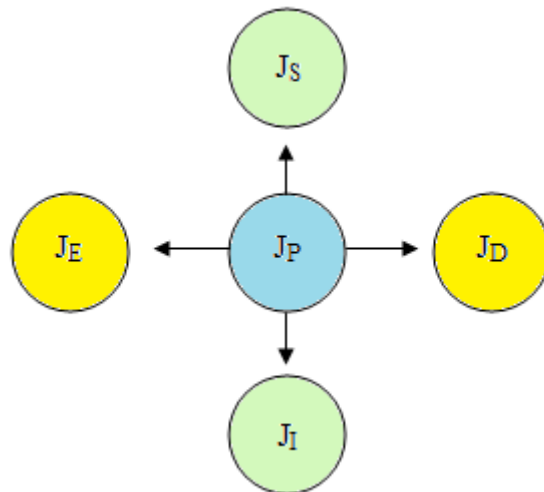


Figura 3.3 – Representação de um nó da árvore de segmentação cheia usando partição flexível.

O processo ocorre até que seja formada a árvore de segmentação cheia. O próximo passo é a determinação da árvore de segmentação ótima através da etapa de poda. Tal etapa também é realizada de maneira análoga à do algoritmo anterior. No entanto, existem, nesse momento, duas decisões a serem tomadas: a primeira consiste

em dividir ou não o bloco, enquanto a segunda está relacionada à escolha da direção de partição, em caso de segmentação.

O procedimento de poda ocorre da seguinte maneira: sejam  $J_P$ ,  $J_E$ ,  $J_D$ ,  $J_S$  e  $J_I$  os custos dos nós pai, filho esquerdo, filho direito, filho superior e filho inferior, respectivamente. Inicialmente, compara-se o custo do nó pai ( $J_P$ ) com o custo total dos nós filho verticais ( $J_{FV}$ ) dado por:

$$J_{FV} = J_E + J_D + \lambda R(\text{flag de segmentação vertical});$$

Caso o custo do nó pai seja inferior ou igual ao custo total dos nós filho resultantes da segmentação vertical ( $J_P \leq J_{FV}$ ), mantém-se inalterado o valor do custo do nó pai, atribui-se o *flag* = 0 (indicando que o bloco não deve ser dividido) e guarda-se o índice do elemento que gerou o menor custo. Nesse momento, ainda não se pode podar a árvore, pois ainda é necessário fazer a comparação com o custo total dos nós filho resultantes da segmentação horizontal. Caso contrário ( $J_P > J_{FV}$ ), a árvore não é podada, o custo do nó pai é atualizado (passando a valer  $J_{FV}$ ) e atribui-se o *flag* = 1, informando que até agora a melhor opção é a partição vertical.

O próximo passo é repetir o processo para os nós filho gerados a partir da partição horizontal. Para tal, calcula-se o custo total dos nós filho horizontais, conforme a equação abaixo:

$$J_{FH} = J_S + J_I + \lambda R(\text{flag de segmentação horizontal});$$

Compara-se, então, o valor do custo total dos nós filho horizontais com o custo do nó pai atualizado. Se o custo do nó pai for superior ao custo total dos nós filho horizontais ( $J_P > J_{FH}$ ), opta-se pela segmentação horizontal, atribuindo-se o *flag* = 2 ao nó em questão (indicando partição horizontal) e atualiza-se novamente o custo do nó pai ( $J_P$ ) que passa a valer  $J_{FH}$ . Como o custo do nó pai foi atualizado na etapa anterior, pode-se concluir que a segmentação horizontal é de fato a melhor opção, independentemente da decisão anterior. Caso contrário ( $J_P \leq J_{FH}$ ), nada é feito, na medida em que o *flag* atribuído ao nó já informa se a segmentação deve ser ou não realizada.

O procedimento é repetido até que toda árvore seja percorrida, começando-se pelos nós folha e terminando no nó raiz. O resultado é a construção da árvore de segmentação ótima que informa como o bloco deve ser codificado.

### 3.6 Resultados

Nesta seção são apresentados os resultados obtidos utilizando-se o MMP-I e o MMP-II. Tais resultados podem ser observados nos gráficos das figuras 3.4 a 3.6 [4]. Foram realizados dois testes: o MMP-I utiliza a técnica de predição, enquanto o MMP-II conta com todas as demais técnicas, exceto a segmentação flexível.

Como pode ser observado, os melhores resultados foram obtidos com o MMP-II.

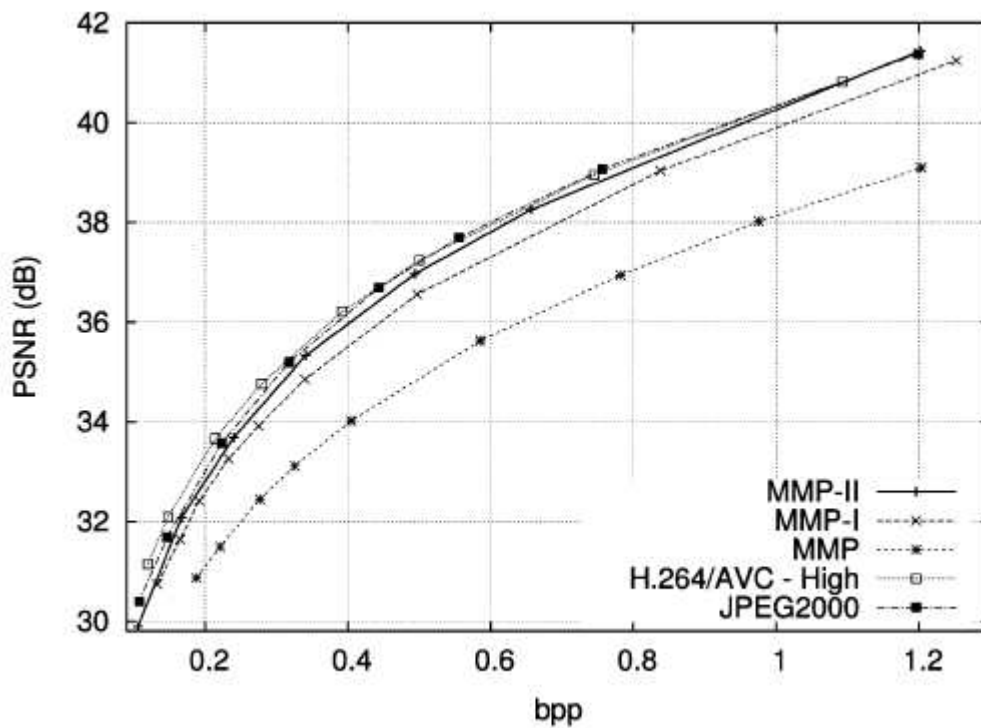


Figura 3.4 – Resultados obtidos para a imagem Lena.

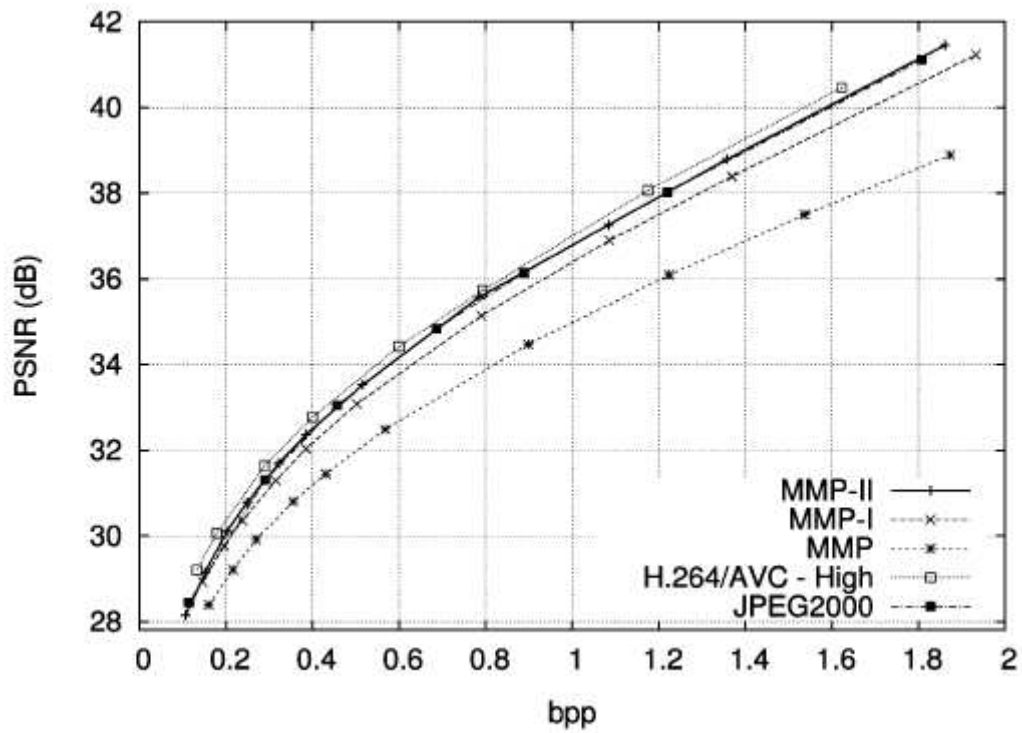


Figura 3.5 – Resultados obtidos para a imagem Goldhill.

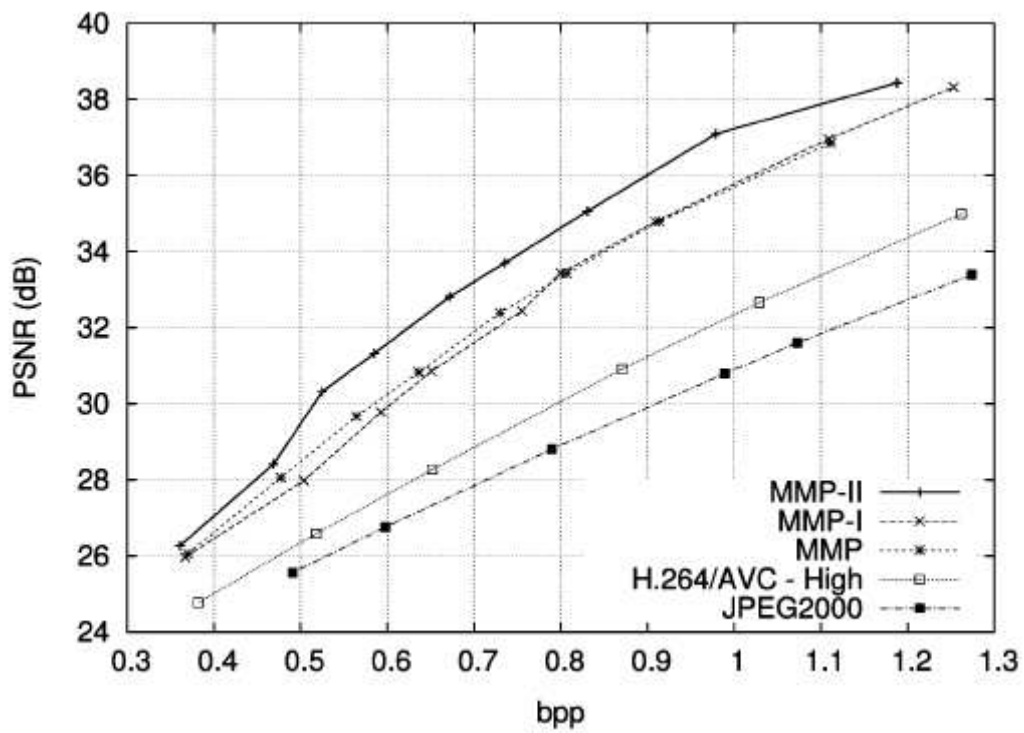


Figura 3.6 – Resultados obtidos para a imagem PP1205.

### **3.7 Conclusão**

Neste capítulo foram descritas as melhorias implementadas com o objetivo de melhorar o processo de codificação do MMP original. O resultado foi o surgimento de uma nova versão do algoritmo chamada MMP-II, apresentando melhores resultados de compressão.

A elaboração de tal versão tem importância fundamental para a implementação do MMP Estéreo, pois pretende-se aliar essa nova versão à técnica de segmentação flexível a fim de processar a imagem esquerda do par. Como tal imagem é a primeira a ser codificada, pode-se concluir que seu processamento se assemelha ao de uma imagem comum, na medida em que nesta etapa ainda não se pode aproveitar as redundâncias existentes entre as imagens que compõem o par. Tais redundâncias serão exploradas durante a codificação da imagem direita, uma vez que, neste momento, já pode-se aproveitar a codificação da imagem de referência no processamento da imagem restante. As técnicas que permitem o aproveitamento dessas redundâncias, visando melhorar o processo de codificação da imagem direita são descritas no próximo capítulo, onde será tratado o MMP Estéreo.



# Capítulo 4

## MMP Estéreo

### Introdução

Neste capítulo será descrito o algoritmo MMP aplicado a imagens estéreo (MMP Estéreo). A partir de agora, será investigado o uso do MMP de modo a explorar eficientemente as características deste tipo de imagem, visando obter melhores resultados.

A primeira tentativa no sentido de explorar as características estéreo das imagens foi introduzir um novo dicionário para o processamento da imagem direita, contendo apenas elementos deslocados entre dois blocos vizinhos de uma linha de blocos (*row of blocks* - ROB) correspondente da imagem esquerda reconstruída (imagem de referência).

Os deslocamentos representam indiretamente valores de disparidade existentes entre as imagens que compõem o par. Esta, porém, é uma consideração grosseira, pois aborda apenas um caso particular, na medida em que a disparidade só corresponde a um deslocamento simples quando o objeto da cena 3D se encontra a uma distância suficientemente grande das duas câmeras. Quando isto ocorre, deformações do objeto devido às distorções de perspectiva não são percebidas.

Com o objetivo de contornar este problema, foi criado um novo dicionário para o processo de codificação da imagem direita. Tal dicionário contém elementos conhecidos como deformados, uma vez que são gerados a partir de um mapa de disparidades aproximado em conjunto com a imagem de referência (esquerda). Assim como na versão anterior do MMP Estéreo, tal mapa não precisa ser transmitido, contornando a inconveniência de que tais mapas apresentam um alto custo para serem enviados.

Com a introdução desses novos elementos, pretende-se incorporar padrões que representem aproximações mais precisas para os blocos da imagem direita, possibilitando uma melhora nos resultados obtidos na sua codificação.

## **4.1 O MMP Estéreo**

O funcionamento do MMP Estéreo ocorre de maneira análoga ao do MMP original, com a diferença de que, a partir de então, deve-se processar duas imagens (par estéreo) ao invés de uma imagem isolada como anteriormente.

O algoritmo inicia o processo de codificação pela imagem esquerda (imagem de referência). Terminada essa etapa, tem início a codificação da imagem direita. Entretanto, nesse momento, o algoritmo pode utilizar informações da imagem já codificada para aprimorar a codificação da imagem restante. Por constituírem um par estéreo, tais imagens apresentam grande quantidade de informação redundante, justificando a importância da imagem esquerda no processamento da imagem direita.

Nos próximos itens, serão descritos detalhadamente os processos de codificação/decodificação das imagens esquerda e direita.

## **4.2 A imagem esquerda**

### **4.2.1 O processo de codificação da imagem esquerda**

O processo de codificação da imagem esquerda deve ocorrer exatamente da mesma forma que em versões anteriores do MMP, na medida em que esta é a primeira imagem do par a ser tratada.

Assim, a sua codificação ocorre da mesma forma que no MMP original, excluindo-se o fato de que foram incorporadas todas as melhorias descritas no capítulo 3. A introdução dessas novas técnicas possibilitou um grande avanço nos resultados obtidos, da mesma maneira que havia ocorrido no MMP-II.

Dessa forma, o processo de codificação da imagem esquerda conta com os seguintes aprimoramentos: predição de blocos, controle de redundância dos elementos, novas técnicas de atualização, divisão do dicionário em gavetas e segmentação flexível.

Ao se unirem as técnicas de predição e segmentação flexível, entretanto, houve a necessidade de se introduzirem algumas mudanças. Isso ocorre, porque a partir de então, diferentemente do algoritmo MMP original, em que cada *flag* de segmentação informava apenas se o bloco havia ou não sido particionado, tornou-se preciso transmitir com tais *flags* três informações.

A primeira consiste em informar se o bloco foi ou não segmentado. A segunda está relacionada ao modo de predição. Assim, em caso de partição do bloco, deve-se indicar se houve divisão ou não do bloco de predição. Finalmente, a terceira e última informação refere-se à direção de segmentação (vertical ou horizontal), em caso de divisão do bloco. Com isso, o *flag* de segmentação passa a poder assumir cinco valores diferentes, conforme definido abaixo:

*flag* = 0, blocos de resíduo e predição não segmentados;

*flag* = 1, blocos de resíduo e predição segmentados na vertical;

*flag* = 2, blocos de resíduo e predição segmentados na horizontal;

*flag* = 3, bloco de resíduo segmentado na vertical e bloco de predição não segmentado;

*flag* = 4, bloco de resíduo segmentado na horizontal e bloco de predição não segmentado.

Isso ocorre, porque durante o processamento de um determinado bloco (resíduo) pode-se decidir por segmentá-lo na direção vertical ou horizontal, mantendo ou não a predição utilizada para gerar o bloco de resíduo original. Dessa forma, a codificação de um dado bloco tem início com a geração do bloco de resíduo correspondente a partir do modo de predição cujo resíduo apresentou o menor custo segundo o critério de decisão. Ao realizar a divisão do bloco original em dois novos vetores, entretanto, existe a possibilidade de se codificar cada uma das metades com um modo de predição diferente ou pode-se manter a predição usada para gerar o bloco de resíduo original. Com isso, o *flag* de segmentação deve informar se houve divisão do bloco e, em caso de divisão,

deve indicar se houve ou não partição do bloco de predição e em qual direção a segmentação foi realizada.

Com exceção da alteração descrita acima, o processo de codificação desta imagem ocorre da mesma maneira que aquele descrito para o MMP-II. Assim, uma vez inicializado o dicionário tem início o processo de codificação bloco a bloco. Para cada elemento formador da imagem constrói-se a árvore de segmentação cheia e, em seguida, é realizado o procedimento de poda.

O processo é repetido até que a imagem seja completamente processada e o dicionário é atualizado com concatenações, contrações e expansões, bem como com os novos elementos oriundos das novas técnicas de geração de blocos.

#### **4.2.2 O processo de decodificação da imagem esquerda**

Uma vez inicializado o dicionário da mesma forma que havia sido feito durante o processo de codificação, tem início a decodificação da imagem esquerda.

Para cada bloco, o decodificador monta a árvore de segmentação ótima e atualiza o dicionário sempre que necessário, de maneira análoga ao procedimento realizado durante a codificação. Como os dicionários estão sempre em sincronia, um dado elemento apresenta sempre a mesma representação dentro de ambos os dicionários, possibilitando a reconstrução da imagem.

O processo é repetido até que todo o arquivo construído pelo codificador seja lido e a imagem completamente decodificada.

### **4.3 A imagem direita**

A primeira tentativa no sentido de explorar as características estéreo das imagens, melhorando os resultados para a imagem direita, foi introduzir um novo dicionário para o seu processamento, contendo apenas elementos deslocados entre dois blocos vizinhos de uma linha de blocos (*row of blocks* - ROB) correspondente da imagem esquerda reconstruída (imagem de referência).

Os deslocamentos representam indiretamente valores de disparidade existentes entre as imagens que compõem o par. Esta, porém, é uma consideração grosseira, pois aborda apenas um caso particular, na medida em que a disparidade só corresponde a um deslocamento simples quando o objeto da cena 3D se encontra a uma distância suficientemente grande das duas câmeras. Quando isto ocorre, deformações do objeto devido às distorções de perspectiva não são percebidas.

Na maior parte das imagens estéreo, a distância de um determinado ponto de um objeto da cena 3D até a câmera esquerda difere da distância do mesmo ponto à câmera direita. Esta diferença de distância pode ser estimada através dos vetores de disparidade. Para fazer uma estimação precisa, resultando em uma melhor qualidade da imagem reconstruída, muitas restrições devem ser atendidas, o que leva a um processo razoavelmente complexo. No caso deste projeto, deseja-se obter uma aproximação do mapa de disparidades. Para tal, realiza-se uma estimação pixel a pixel, resultando em um mapa denso e numa estimação mais precisa e consistente do que a estimação bloco a bloco.

Uma vez calculado o mapa de disparidades entre as duas imagens, pode-se estimar uma delas (direita) a partir da outra (esquerda) e do mapa em questão.

A principal desvantagem das técnicas que utilizam o mapa de disparidades é a necessidade de enviá-lo. Quando mais preciso ele for, maior o *overhead* para transmiti-lo. Com o objetivo de contornar tal problema, o mapa é calculado, neste projeto, de maneira aproximada e de tal modo que não precise ser enviado ao decodificador, na medida em que o cálculo é feito de forma que o mesmo seja capaz de reconstruí-lo.

Utilizando a imagem de referência juntamente com o mapa de disparidades, espera-se gerar elementos que representem estimativas bastante aproximadas dos blocos da imagem direita. Com estes vetores deseja-se aprimorar a representação de efeitos como a distorção de perspectiva causada pela distância entre as duas câmeras estéreo, melhorando a compressão desta imagem.

### **4.3.1 Os blocos deslocados**

A primeira idéia a fim de explorar as características das imagens estéreo, melhorando a compressão da imagem direita, foi a criação de um novo dicionário contendo blocos deslocados da imagem esquerda para codificar a imagem direita. O deslocamento na imagem esquerda é realizado na ROB correspondente à do bloco da imagem direita que está sendo codificado e utiliza um passo de meio pixel.

Como existe similaridade entre as ROBs correspondentes das duas imagens do par estéreo, espera-se que os padrões da imagem esquerda aumentem a eficiência do processo de codificação da imagem direita. A figura 4.1 [1] ilustra as imagens esquerda e direita e suas ROBs correspondentes. A figura 4.2 [1] mostra o deslocamento realizado em um bloco da imagem esquerda reconstruída.

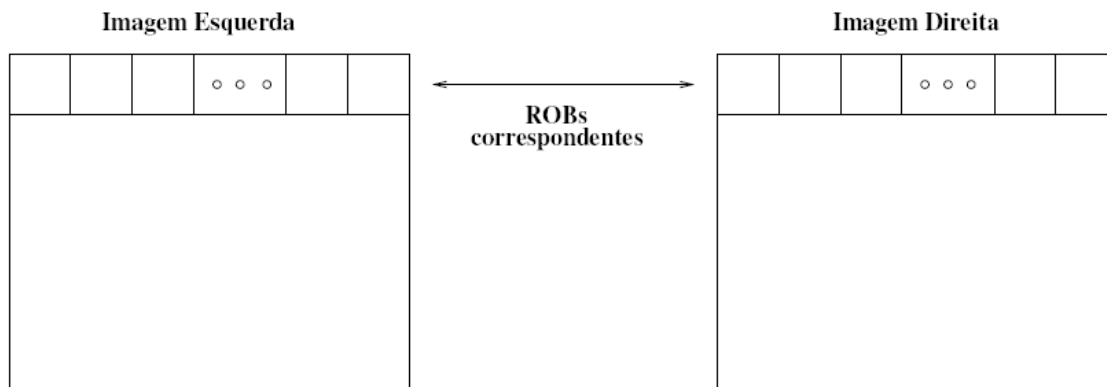


Figura 4.1 – ROBs correspondentes de duas imagens que formam o par estéreo. Fonte: DUARTE, M. H. V., “Codificação de Imagens Estéreo Usando Recorrência de Padrões”, COPPE/UFRJ, Agosto 2003.

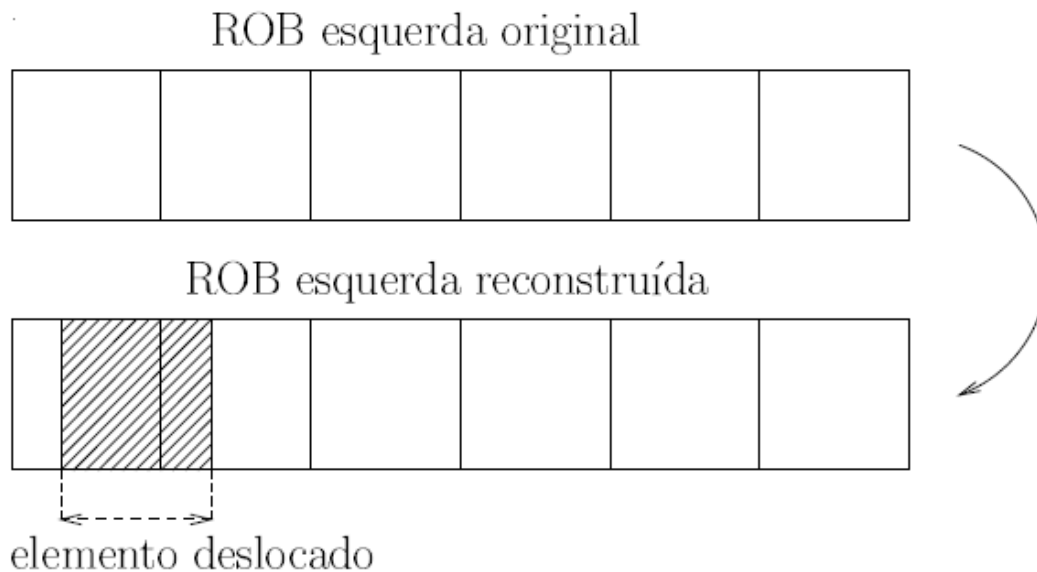


Figura 4.2 – Deslocamento realizado em um bloco da imagem esquerda reconstruída a fim de codificar um bloco da imagem direita pertencente à mesma ROB. Fonte: DUARTE, M. H. V., “Codificação de Imagens Estéreo Usando Recorrência de Padrões”, COPPE/UFRJ, Agosto 2003.

Na realidade, o dicionário de elementos deslocados é apenas virtual. Ele corresponde à ROB da imagem esquerda referente ao bloco da imagem direita que está sendo codificado. A identificação do elemento escolhido é feita através do número de deslocamentos de meio pixel realizados em relação à posição do bloco da imagem esquerda correspondente ao bloco da imagem direita que está sendo processado. Se o deslocamento for feito para esquerda, atribui-se um valor negativo ao índice, indicando o número de deslocamentos de meio pixel realizados nesse sentido, caso contrário, o valor é positivo. Dessa forma, o índice -3 refere-se a um casamento obtido deslocando-se um pixel e meio à esquerda (três deslocamentos de meio pixel). Analogamente, um casamento atingido com um deslocamento de dois pixels à direita é evidenciado pelo índice 4 (quatro deslocamentos de meio pixel).

Com o objetivo de evitar a transmissão de índices negativos, antes de seu envio é feito um mapeamento, conforme definido abaixo:

$$\text{*novo\_índice*} = -2 * \text{*índice*} - 1, \text{ se } \text{*índice*} < 0;$$

$$\text{*novo\_índice*} = 2 * \text{*índice*}, \text{ se } \text{*índice*} \geq 0.$$

Assim, o índice -3 seria mapeado para 5, enquanto o índice 4 assumiria o valor 8.

### **4.3.2 Os blocos deformados**

Os blocos deformados são gerados a partir da união do mapa de disparidades pixel a pixel e da imagem esquerda reconstruída. Com estes novos elementos deseja-se aprimorar a representação de efeitos como a distorção de perspectiva causada pela distância entre as duas câmeras estéreo, melhorando a compressão da imagem direita.

Uma vez que a imagem esquerda é totalmente codificada antes do início do processamento da imagem direita, pode-se concluir que quando se inicia a codificação de tal imagem já se encontra disponível a imagem esquerda reconstruída. Com isso, o próximo passo, antes da formação dos blocos deformados, é o cálculo do mapa de disparidades, descrito no item seguinte.

#### **4.3.2.1 Cálculo do mapa de disparidades**

A obtenção do mapa de disparidades é feita pixel a pixel. Para um determinado pixel da imagem direita, procura-se numa região da imagem esquerda o pixel correspondente, ou seja, aquele que resultou no menor erro quadrático entre as duas regiões em torno do pixel considerado. Tal região possui dimensão 5 x 5.

O procedimento é repetido para todos os pixels da ROB em questão, resultando no mapa de disparidades pixel a pixel, conforme ilustrado na figura 4.3 [1].

Supondo que o valor encontrado para um determinado pixel seja 2, pode-se concluir que partindo da mesma posição do pixel em questão deve-se deslocar um pixel (dois deslocamentos de meio pixel) à direita. O pixel obtido após o deslocamento é aquele que corresponde ao pixel da imagem direita que está sendo tratado nesse momento.

Valores negativos no mapa de disparidades indicam deslocamentos para a esquerda, ao passo que valores positivos determinam deslocamentos para a direita.



O próprio mapa de disparidades obtido entre duas ROBs correspondentes pode ser utilizado diretamente para gerar os elementos a serem usados na codificação da ROB direita seguinte, mas esta é uma estimativa grosseira, porque as disparidades entre um par de ROBs correspondentes pode ser muito diferente das disparidades do par de ROBs seguinte. Como uma tentativa de melhorar esta estimativa, obtém-se uma média simples das colunas das disparidades da ROB anterior. A figura 4.4 ilustra o cálculo do mapa de disparidades pixel a pixel entre duas ROBs, a obtenção do mapa final através da média simples de cada uma das colunas do mapa de disparidades pixel a pixel e sua utilização para o processamento da ROB seguinte.

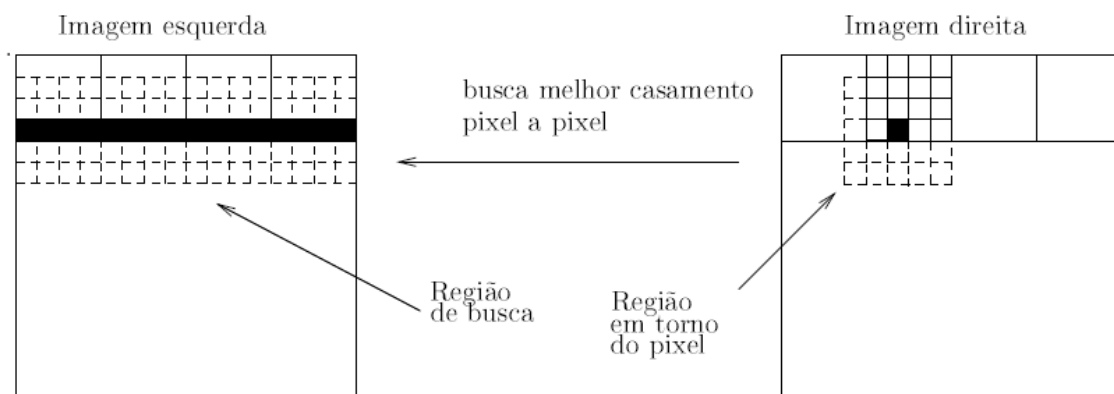


Figura 4.3 – Cálculo do mapa de disparidades pixel a pixel. Fonte: DUARTE, M. H. V., “Codificação de Imagens Estéreo Usando Recorrência de Padrões”, COPPE/UFRJ, Agosto 2003.

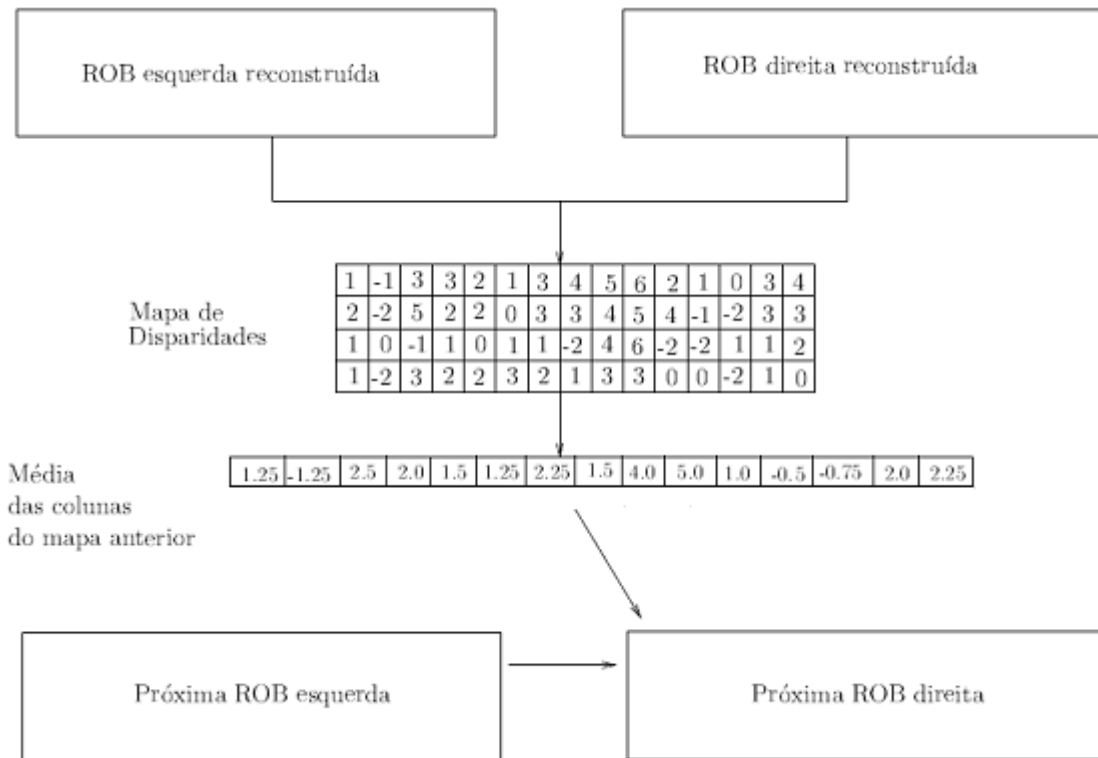


Figura 4.4 – Cálculo da média simples das colunas do mapa de disparidades pixel a pixel para construção do mapa de disparidades da ROB em questão.

Levando-se em conta o modo como os elementos deformados são gerados pode-se concluir que, o processamento da primeira ROB da imagem direita é realizado utilizando-se apenas os elementos do dicionário comum e os do dicionário de deslocados. Isso ocorre, porque nessa etapa do processamento não existem ainda ROB's anteriores, inviabilizando, portanto, o cálculo do mapa de disparidades.

O processo de cálculo do mapa de disparidades é repetido a cada mudança de ROB. Assim, para cada ROB (excluindo-se a primeira) recalcula-se o mapa de disparidades usando a ROB anterior. Tal procedimento é justificado pelo fato de que ROB's próximas tendem a apresentar disparidades parecidas.

#### 4.3.2.2 Construção do dicionário com elementos deformados

O dicionário que contém os blocos deformados possui a mesma dimensão de uma ROB. Uma vez calculado o mapa de disparidades, para cada coluna é verificado o valor da disparidade correspondente. Dessa forma, partindo da posição da coluna da imagem esquerda correspondente à coluna da imagem direita que está sendo tratada, são realizados  $|N|$  deslocamentos de meio pixel na imagem esquerda, onde  $N$  representa o valor obtido no mapa de disparidades na mesma posição da coluna da imagem direita em questão.

Os valores dos pixels da coluna da imagem esquerda são, então, copiados para a coluna do dicionário que corresponde à coluna da imagem direita. Valores negativos de  $N$  indicam  $|N|$  deslocamentos de meio pixel para a esquerda, ao passo que valores positivos, determinam  $|N|$  deslocamentos de meio pixel para a direita.

O procedimento é repetido para todas as colunas da ROB, resultando na construção do dicionário que contém os elementos deformados. Tal dicionário é temporário, na medida em que a cada mudança de ROB é necessário recalculer o mapa de disparidades e, dessa forma, torna-se preciso reconstruir o dicionário em questão.

A figura 4.5 ilustra a construção do dicionário de elementos deformados.

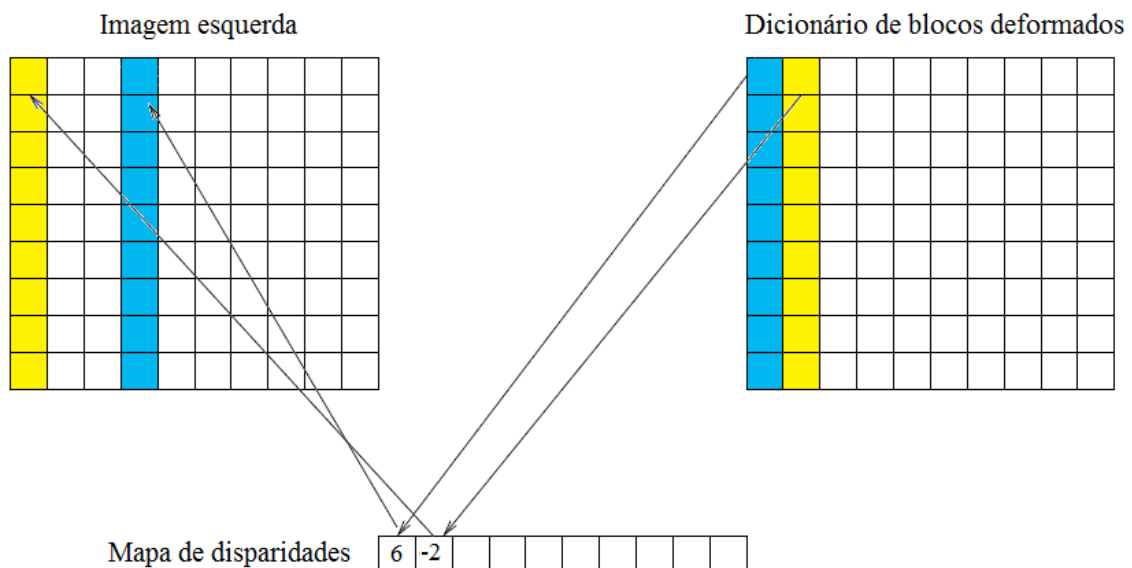


Figura 4.5 – Construção do dicionário de blocos deformados utilizando a imagem esquerda reconstruída e o mapa de disparidades.

### 4.3.3 O processo de codificação da imagem direita

O algoritmo MMP para a codificação da imagem direita inclui boa parte das melhorias adicionadas ao MMP original descritas no capítulo 3. Para o processamento desta imagem, além dos dicionários de elementos deslocados e deformados, inclui-se: controle de redundância dos elementos do dicionário, novas técnicas de atualização, divisão do dicionário comum em gavetas e segmentação flexível.

O processo de codificação desta imagem ocorre de forma análoga ao do MMP original. As alterações relevantes surgem em virtude, além da segmentação flexível, do fato de que esta imagem conta com três dicionários distintos para o seu processamento (dicionário comum, dicionário de deslocados e dicionário de deformados).

Uma vez inicializado o dicionário comum tem início o processo de codificação. Para um dado bloco, procura-se em cada dicionário o elemento que apresenta o menor custo. Determinado o melhor casamento, divide-se o bloco em dois e o procedimento se repete para cada uma das metades.

O processo é repetido até que seja formada a árvore de segmentação cheia. A próxima etapa é a determinação da árvore de segmentação ótima através do procedimento de poda que ocorre da mesma maneira como em versões anteriores.

A diferença surge em relação à identificação dos elementos utilizados. Para cada nó folha da árvore de segmentação ótima deve-se indicar qual elemento foi escolhido. A fim de determiná-lo, primeiramente, é enviado um *flag* que indica a qual dicionário o elemento escolhido pertence. Se o elemento pertencer ao dicionário comum, sua identificação prossegue com o envio de um *flag*, indicando em qual gaveta do nível correspondente ele se encontra, sendo seguido do índice que representa a sua posição na gaveta. Caso o vetor pertença aos demais dicionários, surge uma pequena modificação, na medida em que neles não existe a segmentação em gavetas. Dessa forma, assim como no caso anterior, transmite-se um *flag*, indicando a qual dicionário o elemento pertence, sendo seguido do deslocamento (em unidades de meio pixel), em relação à posição original, realizado.

A atualização do dicionário (dicionário comum) é feita de forma análoga à realizada anteriormente, com a pequena modificação de que caso o elemento escolhido pertença ao dicionário de deslocados ou deformados, tal elemento é inserido no dicionário comum, e não apenas a concatenação, na medida em que este padrão ainda não é conhecido.

#### **4.3.4 O processo de decodificação da imagem direita**

Uma vez inicializado o dicionário comum da mesma forma realizada na etapa de codificação, tem início o processo de decodificação.

O decodificador da imagem direita recebe os *flags* e índices e monta a árvore de segmentação ótima. Para cada nó folha, verifica-se a qual dicionário o elemento escolhido pertence. Se o elemento estiver localizado no dicionário comum, o decodificador sabe que o próximo símbolo representa o *flag* que indica a gaveta em que o elemento se encontra, sendo seguido do índice que o localiza na gaveta. Caso contrário, o próximo símbolo representa o deslocamento em relação à posição original (índice).

Vale lembrar que antes de serem transmitidos, os índices que representam o deslocamento, usados para identificar os elementos dos dicionários de deslocados e deformados, são mapeados. Tal mapeamento tem o objetivo de evitar a transmissão de índices negativos. Dessa maneira, o índice transmitido não representa o deslocamento que efetivamente havia sido feito. Assim, antes de realizar o deslocamento em um dos dicionários, o decodificador desfaz tal mapeamento, obtendo o real valor utilizado pelo codificador para identificar o elemento escolhido.

O procedimento prossegue até que todo o arquivo seja percorrido e a imagem completamente decodificada.

## **4.4 Resultados**

Nesta seção são apresentados os resultados obtidos através do processo de codificação/decodificação de imagens estéreo utilizando o MMP Estéreo. Tais resultados podem ser observados nos gráficos das figuras 4.6 a 4.17. Além disso, é realizada a comparação entre tal algoritmo e outros considerados “estado da arte”.

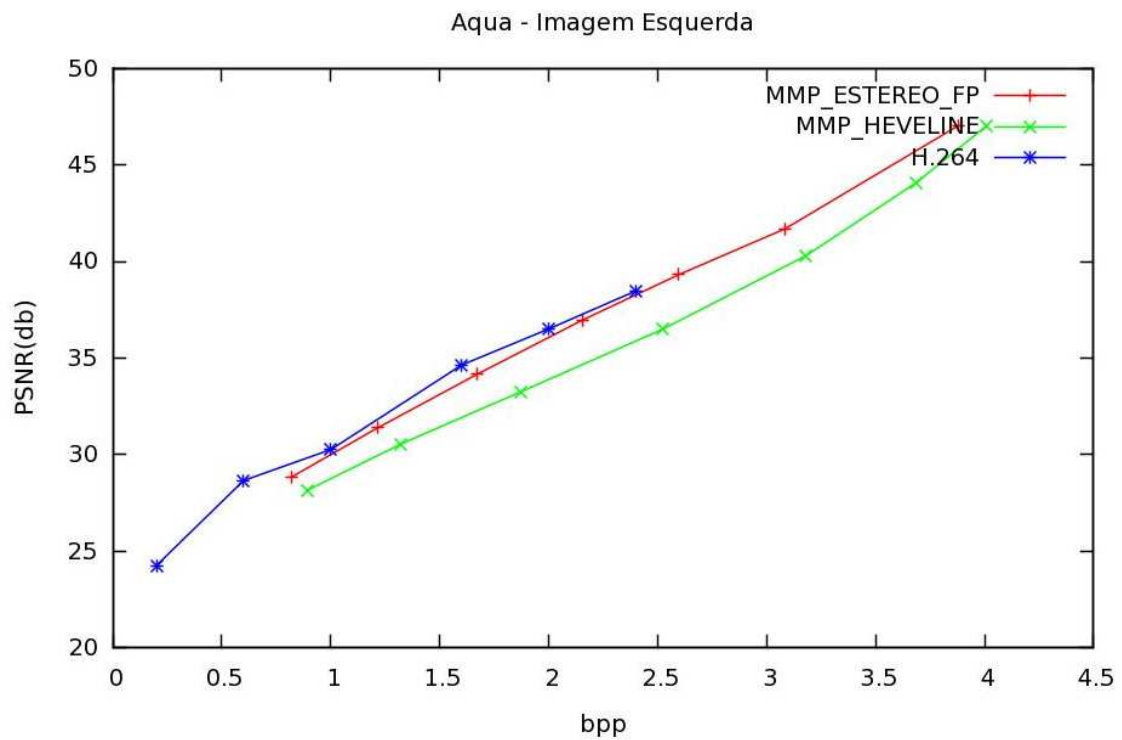


Figura 4.6 – Resultados para a imagem esquerda do par Aqua.

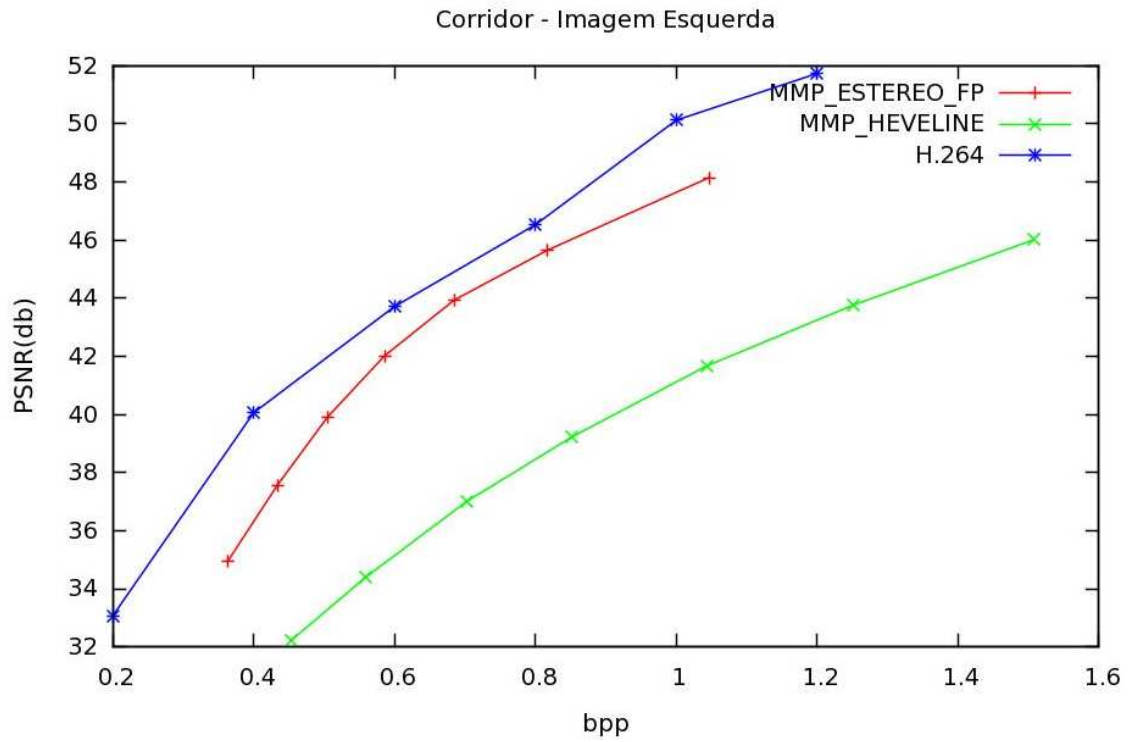


Figura 4.7 – Resultados para a imagem esquerda do par Corridor.

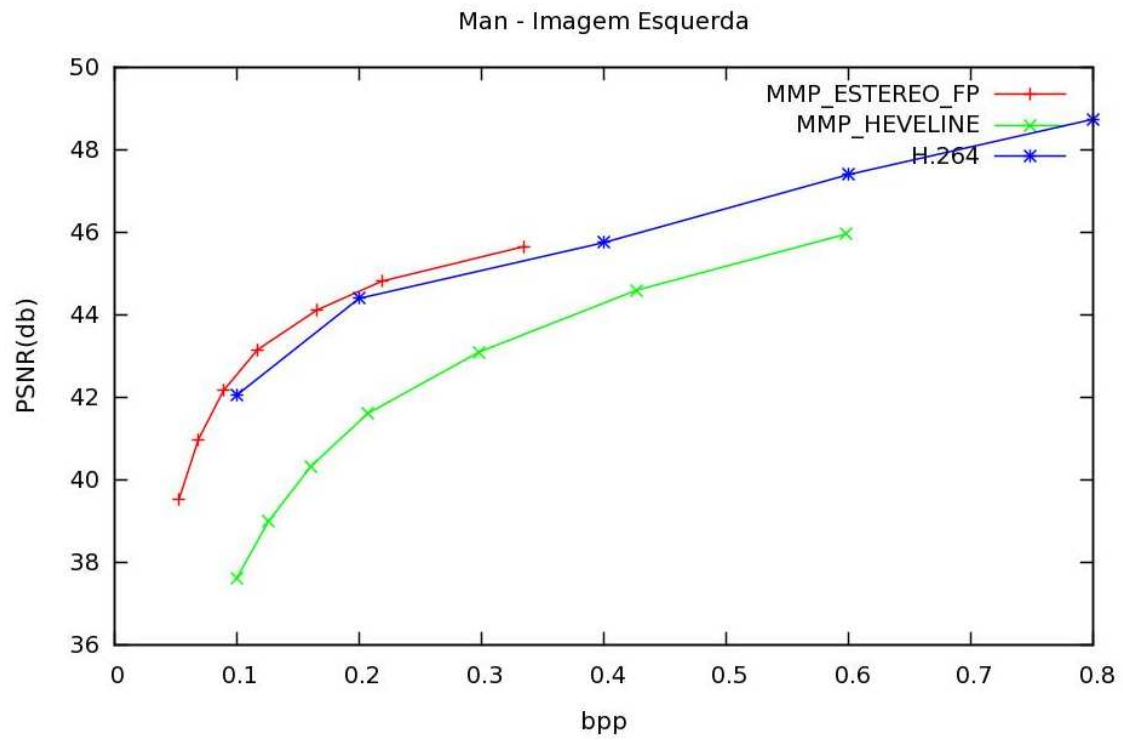


Figura 4.8 – Resultados para a imagem esquerda do par Man.

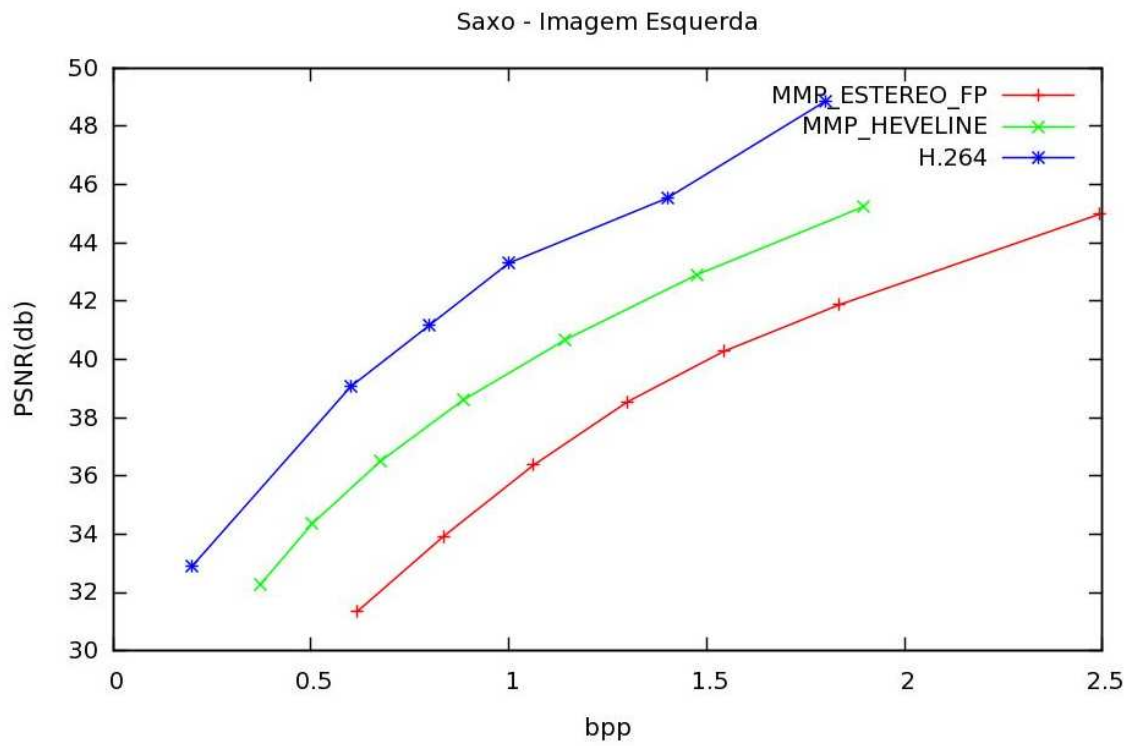


Figura 4.9 – Resultados para a imagem esquerda do par Saxo.

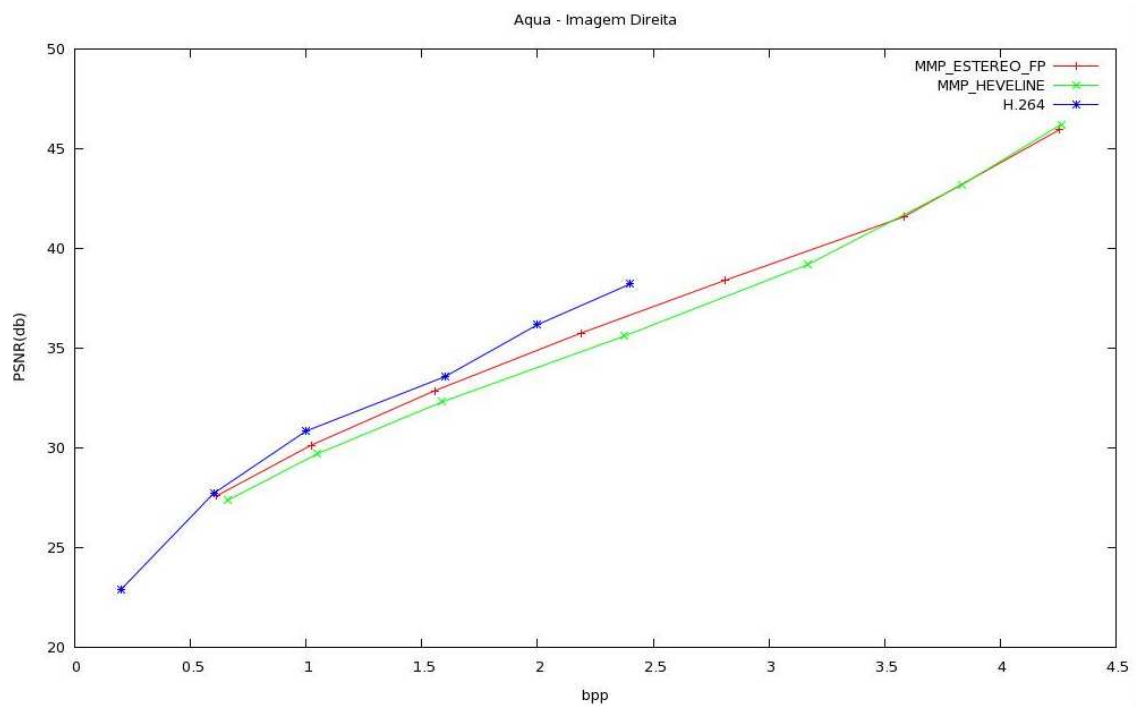




Figura 4.10 – Resultados para a imagem direita do par Aqua.

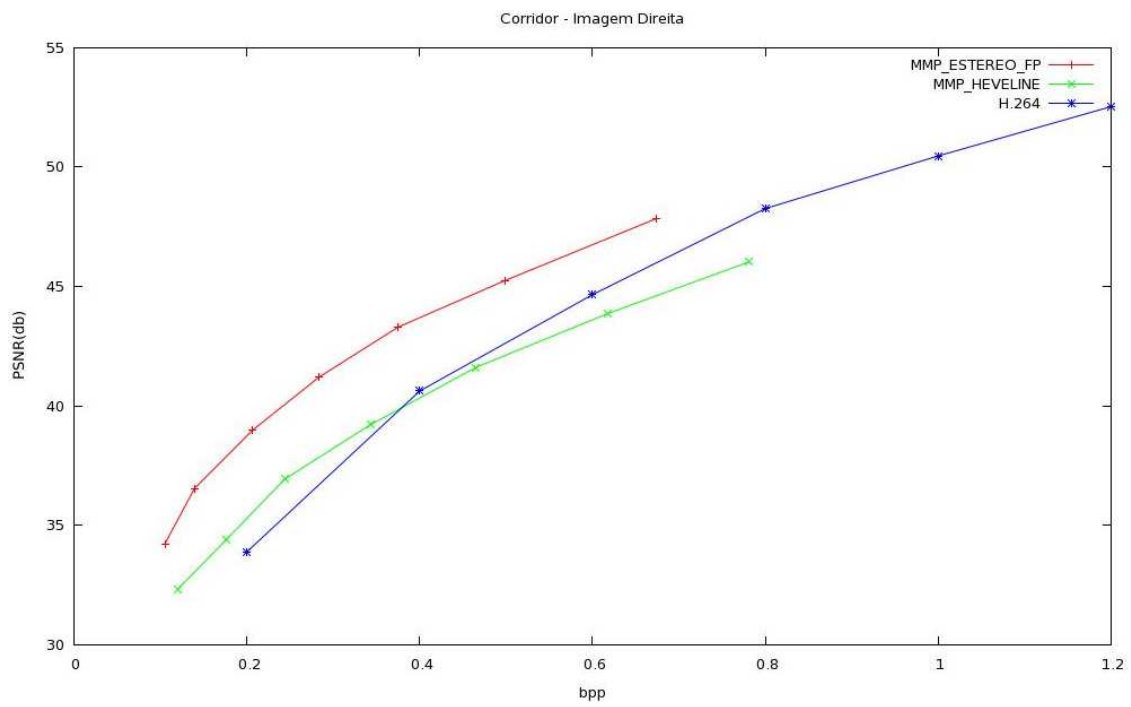


Figura 4.11 – Resultados para a imagem direita do par Corridor.

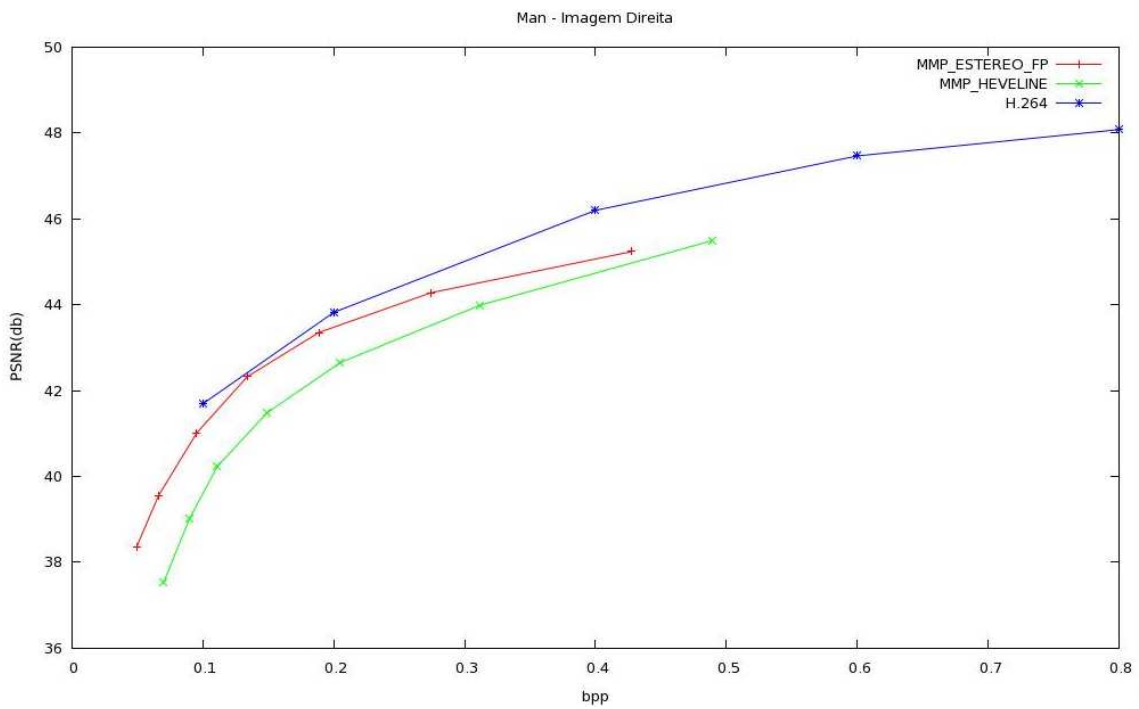


Figura 4.12 – Resultados para a imagem direita do par Man.

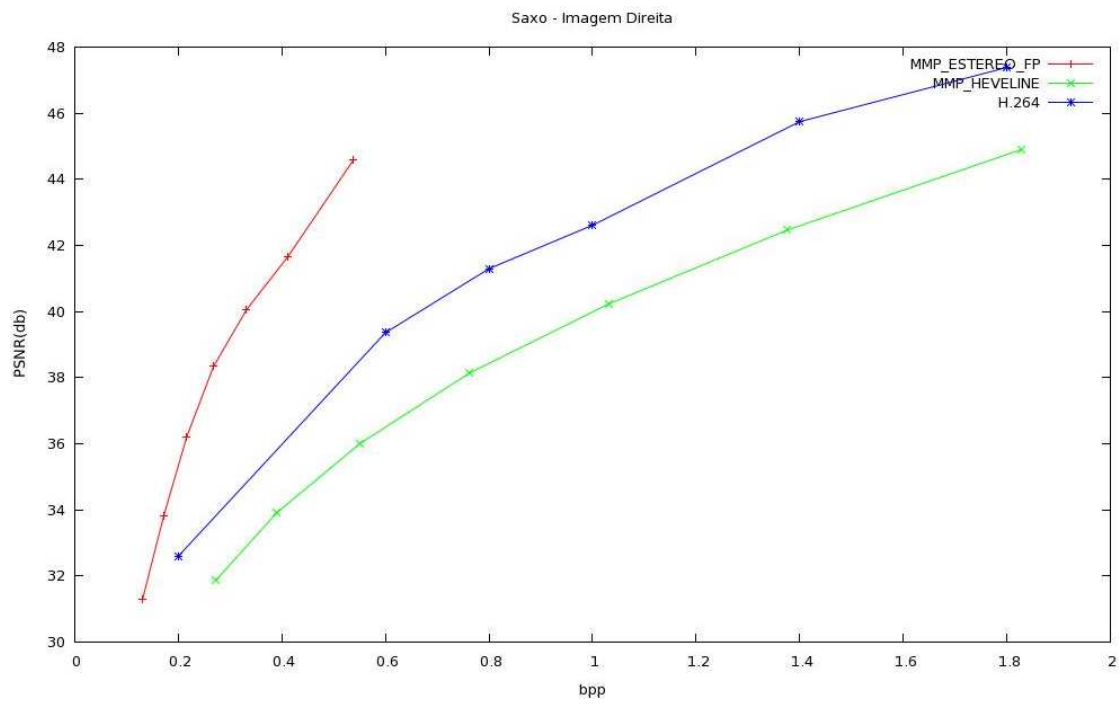


Figura 4.13 – Resultados para a imagem direita do par Saxo.

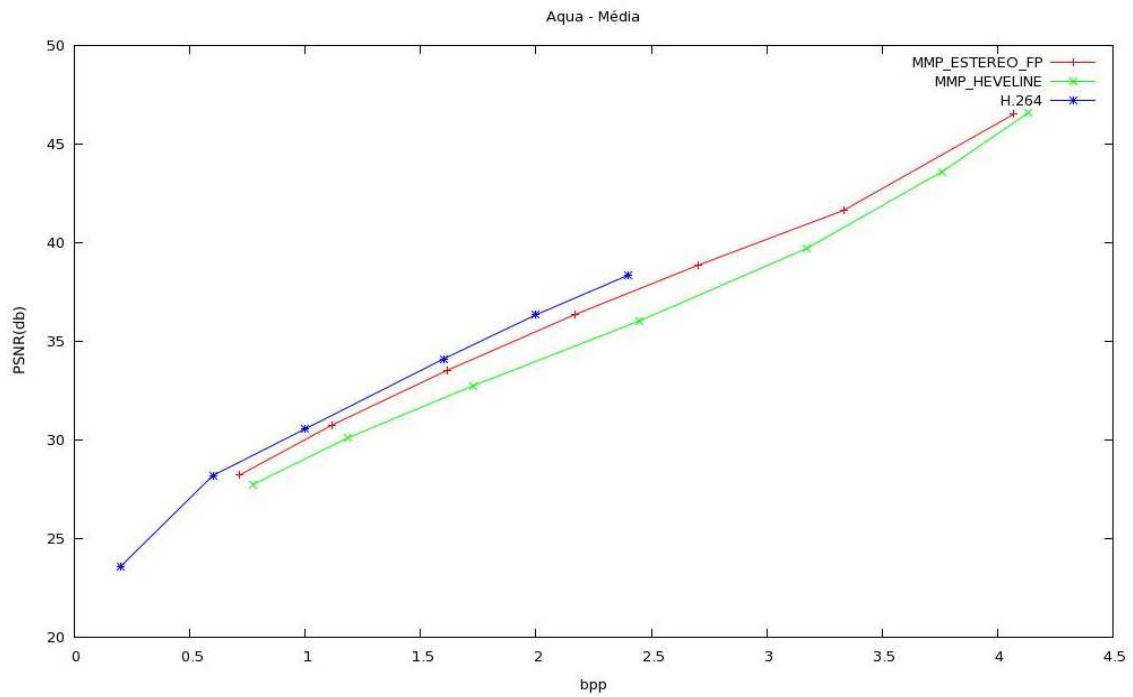


Figura 4.14 – Resultados para a média das imagens do par Aqua.

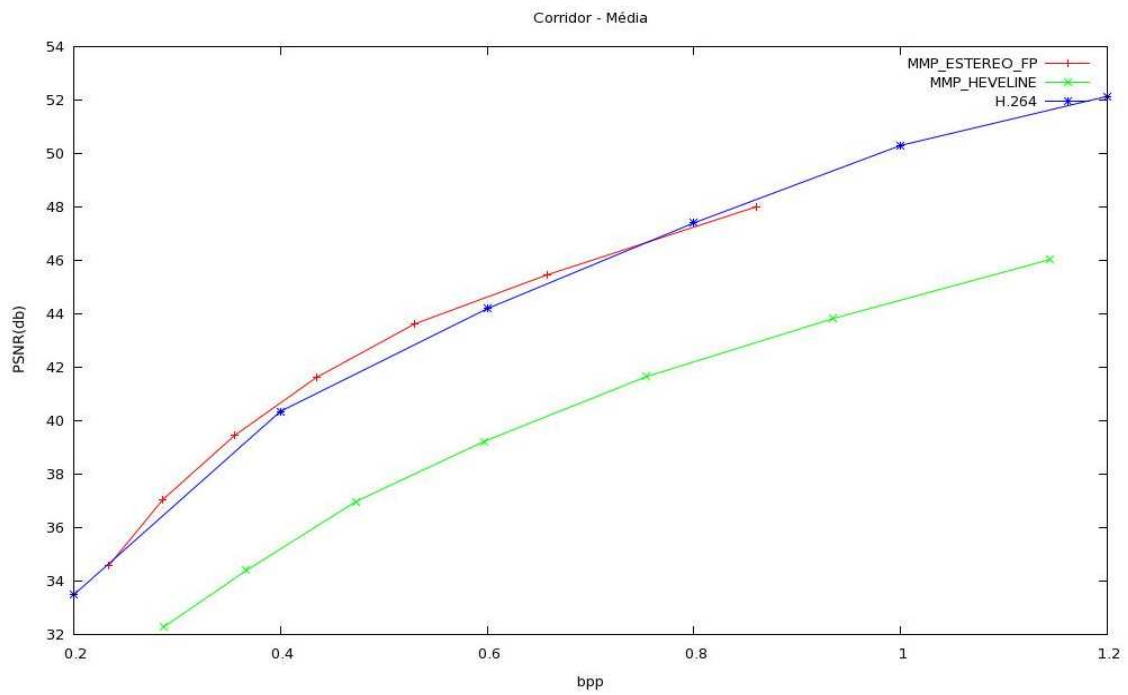


Figura 4.15 – Resultados para a média das imagens do par Corridor.

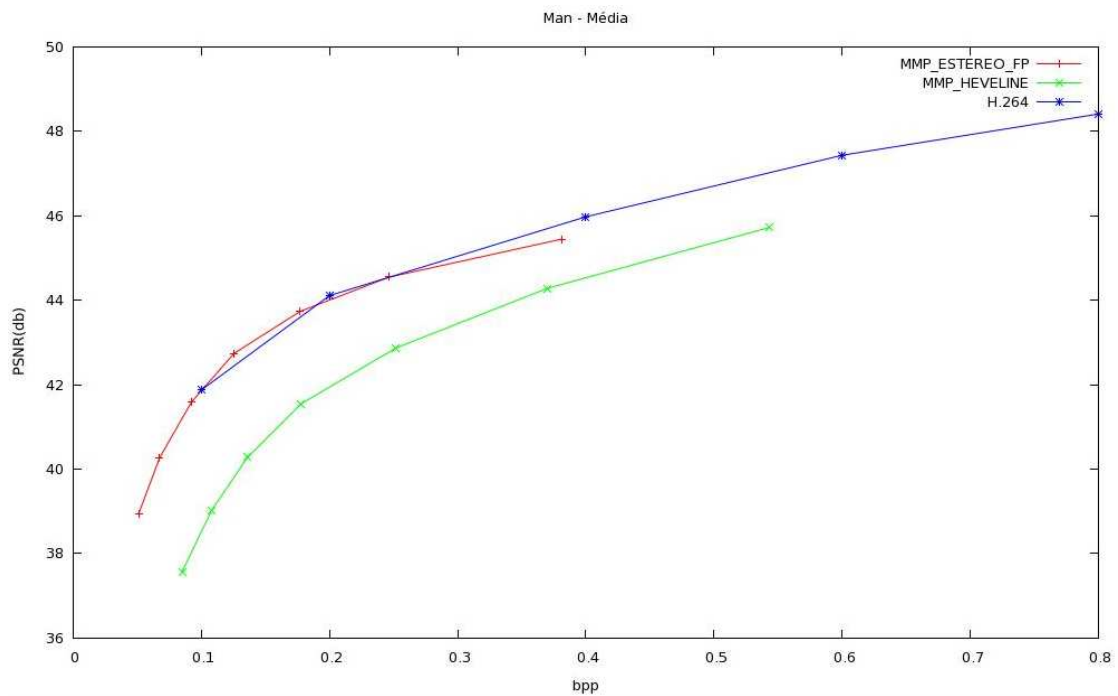


Figura 4.16 – Resultados para a média das imagens do par Man.

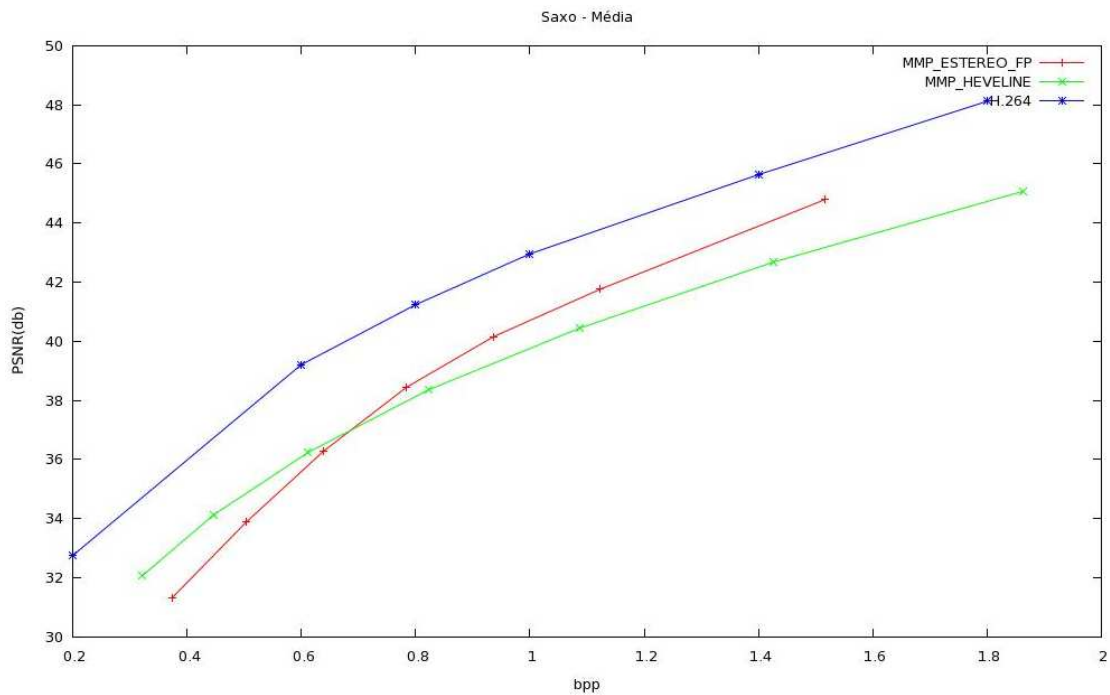


Figura 4.17 – Resultados para a média das imagens do par Saxo.

## 4.5 Conclusão

Neste capítulo foi descrito o algoritmo MMP Estéreo. Tal algoritmo consiste em duas etapas: codificação da imagem esquerda e codificação da imagem direita.

O processo de codificação da imagem esquerda ocorre exatamente da mesma forma que em versões anteriores do MMP, na medida em que esta é a primeira imagem do par a ser tratada. Dessa forma, até esse momento, não se encontram disponíveis informações que possibilitem o aproveitamento das redundâncias existentes entre as imagens que compõem o par. Portanto, a sua codificação ocorre da mesma forma que no MMP original, excluindo-se o fato de que foram incorporadas todas as melhorias descritas no capítulo 3. A introdução dessas novas técnicas possibilitou um grande avanço nos resultados obtidos, da mesma maneira que havia ocorrido no MMP-II.

Uma vez codificada a imagem esquerda tem início o processamento da imagem direita. A partir desse momento, entretanto, torna-se possível o aproveitamento das redundâncias existentes entre as imagens, na medida em que blocos utilizados na codificação da imagem esquerda podem ser reutilizados na codificação da imagem direita. A fim de permitir o aproveitamento de tais redundâncias foram implementadas duas técnicas. Na primeira foi introduzido um novo dicionário, contendo apenas elementos deslocados entre dois blocos vizinhos de uma ROB correspondente da imagem esquerda reconstruída (imagem de referência). Na segunda, por sua vez, foi criado um terceiro dicionário, contendo elementos conhecidos como deformados, uma vez que são gerados a partir de um mapa de disparidades aproximado em conjunto com a imagem de referência (esquerda). Assim, para o processamento da imagem direita são utilizados três dicionários: o primeiro contém elementos comuns (iniciais, concatenados, expandidos, contraídos, etc), o segundo contém os elementos deslocados e, finalmente, o terceiro é formado pelos blocos deformados.

A partir dos resultados obtidos, pode-se afirmar que a nova versão do MMP Estéreo apresentou resultados satisfatórios, uma vez que estes podem ser comparados com aqueles obtidos utilizando-se algoritmos tidos como “estado da arte” como o padrão H.264. Além disso, pôde-se observar uma melhora dos resultados em relação à versão anterior do MMP Estéreo.

# Capítulo 5

## Conclusões

Este trabalho teve como objetivo explorar de maneira eficiente as características das imagens estéreo para que, utilizando o MMP, fosse possível obter resultados satisfatórios no processo de compressão deste tipo de imagem.

O projeto foi dividido em duas fases: processamento da imagem esquerda, seguido do processamento da imagem direita.

A codificação da imagem esquerda é realizada por uma versão chamada MMP-II, à qual foi introduzida a técnica de partição flexível dos blocos. O MMP-II apresenta um conjunto de melhorias em relação ao algoritmo MMP original, entre as quais se pode citar: o uso de métodos de predição, controle de redundância entre os elementos, novas técnicas de atualização e divisão do dicionário em gavetas. Conforme descrito no capítulo 3, tais melhorias possibilitaram o aperfeiçoamento do processo de codificação, de forma que os resultados obtidos utilizando-se o MMP-II foram muito superiores em relação aos obtidos com o MMP original.

Uma vez terminada a codificação da imagem esquerda, tem início a segunda etapa, isto é, o processamento da imagem direita. A primeira tentativa no sentido de explorar as características estéreo das imagens foi introduzir um novo dicionário, contendo apenas elementos deslocados entre dois blocos vizinhos de uma linha de blocos correspondente da imagem esquerda reconstruída (imagem de referência).

Os deslocamentos representam indiretamente valores de disparidade existentes entre as imagens que compõem o par. Esta, porém, é uma consideração grosseira, pois aborda apenas um caso particular, na medida em que a disparidade só corresponde a um deslocamento simples quando o objeto da cena 3D se encontra a uma distância suficientemente grande das duas câmeras. Quando isto ocorre, deformações do objeto devido às distorções de perspectiva não são percebidas.

Com o objetivo de contornar este problema, foi criado um novo dicionário para o processo de codificação da imagem direita. Tal dicionário contém elementos conhecidos como deformados, uma vez que são gerados a partir de um mapa de disparidades aproximado em conjunto com a imagem de referência (esquerda). Assim como na versão anterior do MMP Estéreo, tal mapa não precisa ser transmitido, contornando a inconveniência de que tais mapas apresentam um alto custo para serem enviados.

Com isso, surge a nova versão do algoritmo MMP Estéreo. Através dos resultados apresentados no capítulo 4 pode-se concluir que as novas técnicas implementadas nessa versão de fato foram eficazes, na medida em que permitiram a obtenção de melhores resultados em relação à versão anterior do MMP Estéreo, implementada por Maria Heveline em sua tese “Codificação de Imagens Estéreo Usando Recorrência de Padrões”. Além disso, pode-se afirmar que o MMP constitui uma classe promissora de algoritmos para a codificação de imagens estéreo, uma vez que seus resultados podem ser comparados aos de algoritmos tidos como “estado da arte” como o padrão H.264.

# Referências Bibliográficas

[1] DUARTE, M. H. V., “Codificação de Imagens Estéreo Usando Recorrência de Padrões”, COPPE/UFRJ, Agosto 2003.

[2] PINAGÉ, F. S., “Avaliação do Desempenho de Algoritmos de Compressão de Imagens Usando Recorrência de Padrões Multiescala”, COPPE/UFRJ, Julho 2005.

[3] CARVALHO, M. B., “Multidimensional Signal Compression using Multiscale Recurrent Patterns”, COPPE/UFRJ, Abril 2001.

[4] RODRIGUES, N. M. M., SILVA, E. A. B., CARVALHO, M. B., FARIA, S. M. M., SILVA, V. M. M., “On Dictionary Adaptation for Recurrent Pattern Image Coding”, IEE Transactions on Image Processing, VOL. 17, NO. 9, Setembro 2008.

[5] JÚNIOR, W. S. S., “Compressão de Imagens Utilizando Recorrência de Padrões Multiescalas com Segmentação Flexível”, COPPE/UFRJ, Dezembro 2004.

[6] FRANCISCO, N. C., RODRIGUES, N. M. M., DA SILVA, E. A. B., CARVALHO, M. B., DE FARIA, S. M. M., DA SILVA, V. M. M., REIS, M., “Multiscale Recurrent Pattern Image Coding With a Flexible Partition Scheme”. IEEE International Conference on Image Processing, pp. San Diego, Outubro 2008.



# Apêndice A

## Imagens Originais



Figura A.1 – Aerial (512 x 512).

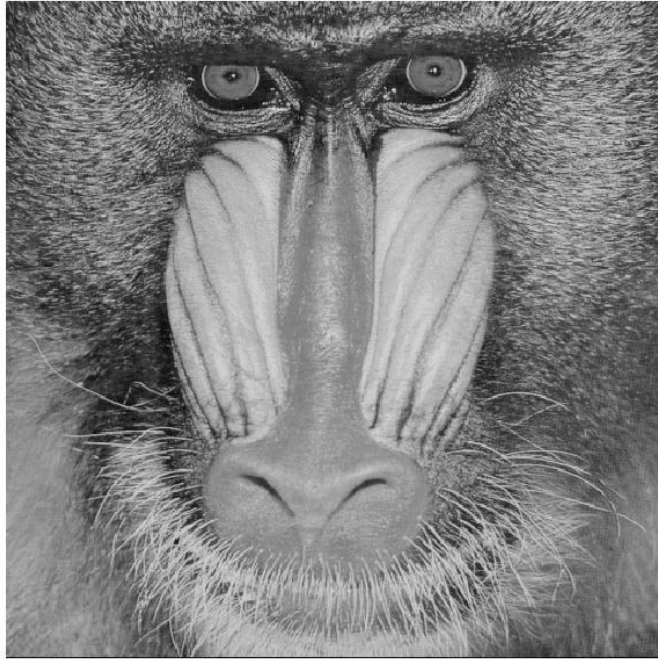


Figura A.2 – Baboon (512 x 512).



Figura A.3 – Barbara (512 x 512).



Figura A.4 – Bridge (512 x 512).



Figura A.5 – Gold (512 x 512).



Figura A.6 – Lena (512 x 512).

100 MACHON et al.: BAYESIAN APPROACH FOR THE ESTIMATION AND TRANSMISSION OF REGULARIZATION PARAMETERS 129

28 The convergence of this algorithm is established by realizing that it corresponds to the EM algorithm, where the complete data are the observations  $g$  and the unknown reconstruction  $f$ , that is  $z^t = (f^t, g^t)$  and

$$g = (I - \theta)z.$$

30 Details are provided in [20].

31 **B. Combining Information from the Coder: Gamma Priors**

32 It is clear that the described process for estimating the image and the hyperparameters can also be performed at the coder, where we use the original image  $f$  as observation  $g$  and again flat hyperpriors for the hyperparameters. In this case (20) becomes

$$f^{(n_c, \alpha_c, \beta)^{cod}} = \arg \min_x \{M(z, f | (\alpha_c, \alpha_r, \beta))\} \\ = \arg \min_x \{A(z | \alpha_c, \alpha_r) + B(f | x, \beta)\}. \quad (27)$$

34 and the hyperparameters are also estimated using the original image as observation, that is,

$$\hat{\alpha}_c^{cod}, \hat{\alpha}_r^{cod}, \hat{\beta}^{cod} = \arg \max_{\alpha_c, \alpha_r, \beta} \int_x p(x, f | (\alpha_c, \alpha_r, \beta)) dx. \quad (28)$$

36 It is clear that to obtain  $\hat{\alpha}_c^{cod}$ ,  $\hat{\alpha}_r^{cod}$  and  $\hat{\beta}^{cod}$  we only need to run Algorithm 1 or Algorithm 2 using the original image as observation.

38 A (quantized) version of  $\hat{\alpha}_c^{cod}$ ,  $\hat{\alpha}_r^{cod}$  and  $\hat{\beta}^{cod}$  is received by the decoder, and denoted, respectively, by  $m_c^{cod}$ ,  $m_r^{cod}$  and  $\eta^{cod}$ . They are used as prior information in guiding the estimation of the hyperparameters at the decoder. More specifically, they are used in defining the following hyperpriors for each hyperparameter

$$p(\alpha_c) \propto \alpha_c^{(m_c^{cod})-1} \exp[-I(m_c^{cod}) \alpha_c / m_c^{cod}] \quad (29)$$

$$p(\alpha_r) \propto \alpha_r^{(m_r^{cod})-1} \exp[-I(m_r^{cod}) \alpha_r / m_r^{cod}] \quad (30)$$

$$p(\beta) \propto \beta^{(m_\beta^{cod})-1} \exp[-I(m_\beta^{cod}) \beta / m_\beta^{cod}]. \quad (31)$$

40 Following again the hierarchical Bayesian approach to the reconstruction problem and using the gamma distributions in (29)–(31), we perform the estimation of the hyperparameters and the reconstruction using the following two steps.

41 1) Estimate  $\alpha_c$ ,  $\alpha_r$ ,  $\beta$  by (see Appendix II-A)

$$\hat{\alpha}_c = \frac{2I(m_c^{cod})}{2I(m_c^{cod}) + p} \quad (37)$$

$$\hat{\alpha}_r = \frac{2I(m_r^{cod})}{2I(m_r^{cod}) + p} \quad (38)$$

42 The derivation of the parameter estimation step when (33) is used instead of (32) is similar to the process described in Appendix II-A and it will therefore not be shown here. We notice that the reconstruction step is the same for the flat and gamma hyperprior cases.

43 Using steps 1 and 2 above the following algorithm is proposed for the simultaneous estimation of the hyperparameters and the image assuming gamma hyperpriors.

44 **Algorithm 3**

- 1) Choose  $\alpha_c^0$ ,  $\alpha_r^0$  and  $\beta^0$ .
- 2) Compute  $f^{(n_c, \alpha_c^0, \beta^0)}$  and  $f^{(n_r, \alpha_r^0, \beta^0)}$  from (A11), (A12) and (A13), (A14), respectively.
- 3) For  $k = 1, 2, \dots$ 
  - a) Estimate  $\alpha_c^k$ ,  $\alpha_r^k$  and  $\beta^k$  by substituting  $\alpha_c^{k-1}$ ,  $\alpha_r^{k-1}$  and  $\beta^{k-1}$  in the right hand side of (B2)–(B4).
  - b) Compute  $f^{(n_c, \alpha_c^k, \beta^k)}$  and  $f^{(n_r, \alpha_r^k, \beta^k)}$  from (A11), (A12) and (A13), (A14), respectively.
- 4) Go to 3 until  $\|f^{(n_c, \alpha_c^k, \beta^k)} - f^{(n_c, \alpha_c^{k-1}, \beta^{k-1})}\| + \|f^{(n_r, \alpha_r^k, \beta^k)} - f^{(n_r, \alpha_r^{k-1}, \beta^{k-1})}\|$  is less than a prescribed bound.
- 5) Using  $\alpha_c^k$ ,  $\alpha_r^k$ ,  $\beta^k$  calculate  $f^{(n_c, \alpha_c^k, \beta^k)}$  by solving (A15)–(A18).

46 The proof of the convergence of this algorithm is again based on the fact that it is an EM algorithm, (see (34)). Assuming that  $p \approx p - 2$  and  $q \approx q - 2$ , we can write (B2)–(B4) as

$$\frac{1}{\alpha_c^k} = \mu_c \frac{1}{m_c^{cod}} + (1 - \mu_c) \frac{1}{\alpha_c^{k-1}} \quad (34)$$

$$\frac{1}{\alpha_r^k} = \mu_r \frac{1}{m_r^{cod}} + (1 - \mu_r) \frac{1}{\alpha_r^{k-1}} \quad (35)$$

$$\frac{1}{\beta^k} = \nu \frac{1}{m_\beta^{cod}} + (1 - \nu) \frac{1}{\beta^{k-1}} \quad (36)$$

48 where

Figura A.7 – PP1205 (512 x 512).

# Apêndice B

## Imagens Estéreo Originais

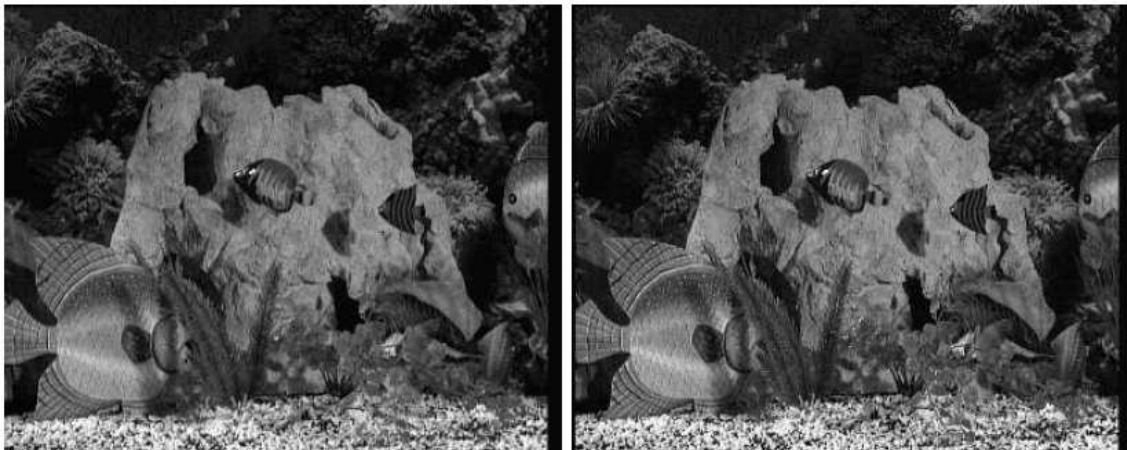


Figura B.1 – Aqua (360 x 288).

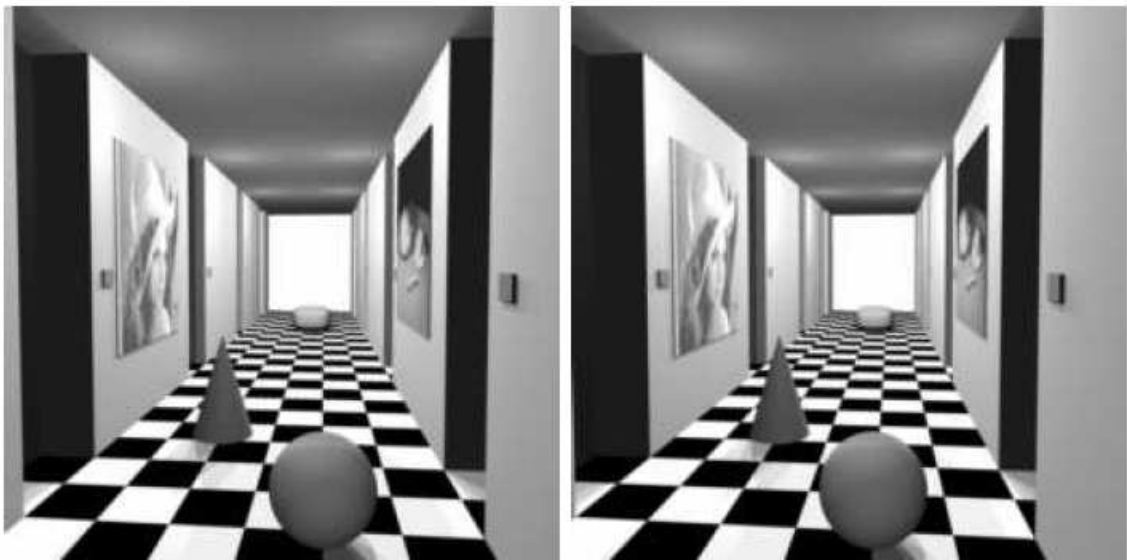


Figura B.2 – Corridor (256 x 256).



Figura B.3 – Man (384 x 384).



Figura B.4 – Saxo (700 x 576).

## Apêndice C

### Imagens Estéreo Reconstruídas

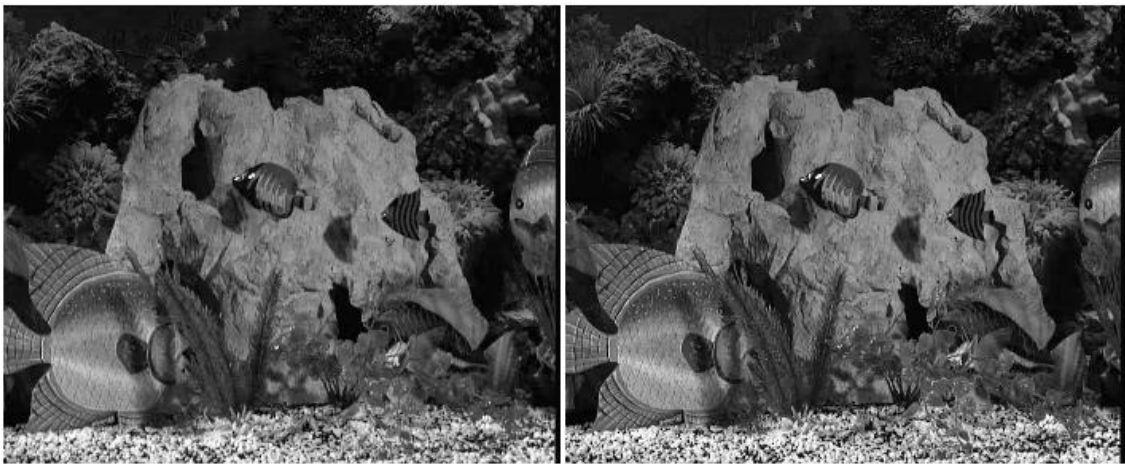


Figura C.1 – Par estéreo Aqua reconstruído. Imagem esquerda ( $\lambda = 32$ ): PSNR = 34.17; taxa = 1,67. Imagem direita ( $\lambda = 32$ ): PSNR = 32,86; taxa = 1,55.

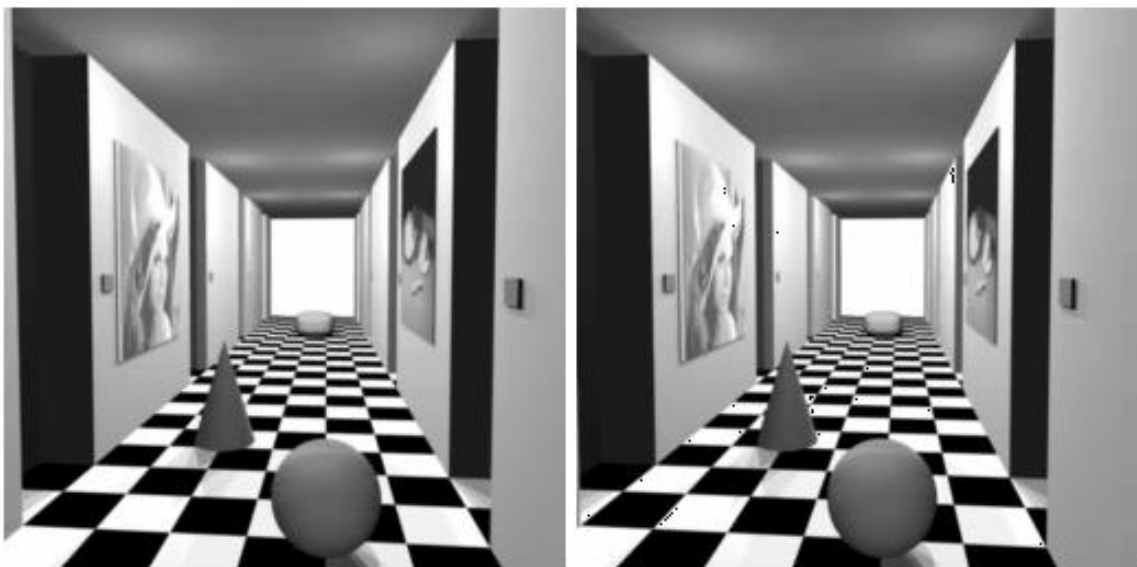


Figura C.2 – Par estéreo Corridor reconstruído. Imagem esquerda ( $\lambda = 8$ ): PSNR = 43,93; taxa = 0,68. Imagem direita ( $\lambda = 8$ ): PSNR = 43,30; taxa = 0,37.



Figura C.3 – Par estéreo Man reconstruído. Imagem esquerda ( $\lambda = 64$ ): PSNR = 40,98; taxa = 0,069. Imagem direita ( $\lambda = 64$ ): PSNR = 39,56; taxa = 0,066.



Figura C.4 – Par estéreo Saxo reconstruído. Imagem esquerda ( $\lambda = 16$ ): PSNR = 38,55; taxa = 1,30. Imagem direita ( $\lambda = 16$ ): PSNR = 38,33; taxa = 0,27.