

CODIFICAÇÃO DE IMAGENS UTILIZANDO MALHAS TRIANGULARES

Cesar da Silva Barreto

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

Prof. Gelson Vieira Mendonça, Ph.D.

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

Prof. Abraham Alcaim, Ph.D.

Prof. Weiler Alves Finamore, Ph.D.

Prof. Hélio Côrtes Vieira Lopes, Ph.D.

Prof. Murilo Bresciani de Carvalho, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

DEZEMBRO DE 2003

BARRETO, CESAR DA SILVA

Codificação de Imagens Utilizando
Malhas Triangulares [Rio de Janeiro]
2003

XII, 83 p., 29,7 cm (COPPE/UFRJ,
D.Sc., Engenharia Elétrica, 2003)

Tese - Universidade Federal do Rio de
Janeiro, COPPE

1. Codificação de imagens 2. Malhas
triangulares 3. Taxa-distorção

I.COPPE/UFRJ II.Título (série)

Agradeço aos meus orientadores, Professores Gelson Vieira Mendonça e Eduardo Antônio Barros da Silva, ao colega Sérgio Rodrigues Neves, ao Programa de Engenharia Elétrica da COPPE/UFRJ e ao Grupo de Sistemas Digitais do IPqM.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

CODIFICAÇÃO DE IMAGENS UTILIZANDO MALHAS TRIANGULARES

Cesar da Silva Barreto

Dezembro/2003

Orientadores: Gelson Vieira Mendonça

Eduardo Antônio Barros da Silva

Programa de Engenharia Elétrica

Malhas poligonais são largamente empregadas na representação de forma discreta de superfícies e volumes. Este trabalho investiga a utilização de malhas triangulares hierárquicas, constituídas de triângulos isósceles retângulos, na codificação de imagens. Inicialmente, a imagem é aproximada por uma malha fina que, posteriormente, vai sendo simplificada, produzindo uma malha hierárquica. Os nós da malha possuem posições fixas, de acordo com seu nível na hierarquia, evitando assim a necessidade de codificar suas coordenadas. Os arcos da malha podem ser permutados de forma a melhor aproximar as arestas ou bordas da imagem original. Todas as decisões se baseiam em uma otimização em termos de taxa-distorção. Concluída a aproximação da imagem, a malha é codificada, por meio de codificação por entropia das amplitudes dos seus nós, direções dos seus arcos e estrutura da árvore descritiva da sua hierarquia. Os resultados da codificação, principalmente em baixas taxas de bits, são encorajadores, considerando-se que a codificação se processa diretamente sobre a imagem, sem nenhuma transformação preliminar.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

IMAGE CODING USING TRIANGULAR MESHES

Cesar da Silva Barreto

December/2003

Advisors: Gelson Vieira Mendonça

Eduardo Antônio Barros da Silva

Department: Electrical Engineering

Polygonal meshes are largely employed for discrete form representation of surfaces and volumes. This work investigates the use of hierarchical triangular meshes, composed of isosceles rectangular triangles, to encode images. Initially, the image is approximated by a thin mesh and after some processing it is simplified, producing a hierarchical mesh. The mesh nodes have fixed positions, according to their hierarchical level, so avoiding the need of coding their coordinates. The mesh arcs may be swapped in order to best approximate the edges of the original image. All the decisions are based on a rate-distortion optimization criterium. After approximating the image, the mesh is encoded, by entropy coding their node amplitudes, their arc directions and the structure of the tree that describes its hierarchy. Considering that the coding is done directly in to the image, without any prior transformation, the proposed method leads to encouraging results at low bit rates.

Índice

1	Introdução	1
2	Representação de Imagens por Malhas	4
2.1	Malhas em imagens naturais	5
2.2	Malhas em imagens sintéticas e naturais	8
2.3	Malhas em imagens sintéticas	8
2.4	Definição da malha básica usada neste trabalho	9
2.5	Principais vantagens da malha escolhida	10
3	Codificação com Malha Triangular Fixa	14
3.1	Descrição da malha	14
3.2	Aproximação da imagem	15
3.2.1	Amplitudes iniciais dos nós	17
3.2.2	Interpolação para obtenção dos pixels	17
3.2.3	Permuta dos arcos	18
3.2.4	Otimização das amplitudes dos nós	21
3.3	Quantização e codificação	22
3.3.1	Quantização das amplitudes dos nós	23
3.3.2	Interpolação dos nós	24
3.3.3	Limiar para permuta dos arcos	25
3.4	Resultados experimentais	26
3.5	Conclusões	27
4	Codificação com Malha Hierárquica	35
4.1	Refinamento da malha	35
4.1.1	Problema da divisão em cascata	35

4.2	Seqüência da divisão hierárquica da malha	37
4.3	Definição da malha inicial	37
4.3.1	Posição dos nós em relação aos pixels	39
4.3.2	Amplitudes iniciais dos nós	39
4.3.3	Direções dos arcos	39
4.3.4	Ajuste e quantização das amplitudes dos nós	40
4.4	Simplificação da malha	40
4.4.1	Condições para a simplificação	41
4.4.2	Análise da variação na distorção	42
4.4.3	Repetição da Simplificação	43
4.4.4	Malha multinível	43
4.4.5	Direção dos arcos	44
4.5	Codificação da estrutura da malha	45
4.6	Resultados experimentais	46
4.6.1	Direções dos arcos na malha 2×1	49
4.7	Conclusões	51
5	Codificação com critério taxa-distorção	56
5.1	Critério taxa-distorção	56
5.1.1	Restrição pelo tamanho do código	57
5.1.2	Dependência entre elementos da malha	57
5.1.3	Otimização Lagrangeana	58
5.2	Codificação da malha	58
5.2.1	Custo dos arcos ortogonais no nível mais baixo	59
5.2.2	Custo dos arcos diagonais	59
5.2.3	Custo das amplitudes dos nós	60
5.2.4	Custo da árvore da malha	60
5.3	Resultados experimentais	65
5.4	Conclusões	67
6	Conclusões	79
	Bibliografia	81

Lista de Figuras

2.1	Exemplo de malha triangular que aproxima a imagem Lena 512×512 .	5
2.2	Malha em (a) refinada em (b), por um passo de biseções.	9
2.3	Esboço da malha usada.	10
2.4	Malha triangular fixa com triângulos de 16×8 pixels para uma imagem de 32×32 pixels (a). (b) A mesma malha com os triângulos subdivididos em 4 menores, de 8×4 pixels.	11
2.5	Possíveis direções de arcos que partem de um mesmo nó.	12
3.1	Posicionamento dos nós da malha em uma imagem de 32×32 pixels e separação dos conjuntos de pixels que são associados a cada face triangular da superfície gerada.	15
3.2	Malha triangular fixa com triângulos de 16×8 pixels para uma imagem de 32×32 pixels.	16
3.3	Losangos contendo os pixels que são usados no cálculo da média para a estimativa inicial das amplitudes dos nós na malha com triângulos de 16×8 pixels.	17
3.4	Triângulo ABC com ponto P interno, definindo três áreas, w_A , w_B e w_C	18
3.5	Permuta de arcos ortogonais (H/V).	19
3.6	Permuta de dois arcos ortogonais horizontais para verticais.	20
3.7	Permuta de vários arcos diagonais.	20
3.8	Permuta de dois arcos diagonais, um para a direção mais vertical e outro para a direção mais horizontal.	21
3.9	Malha com indicação de três nós que não são codificados, mas sim interpolados a partir de cada par de nós do triângulo maior que os contém.	25

3.10	Resultados em taxa×distorção da codificação da imagem Lena com malhas triangulares e com o codificador SPIHT (curva superior).	28
3.11	Resultados em taxa×distorção da codificação da imagem Mandrill com malhas triangulares e com o codificador SPIHT (curva superior).	29
3.12	(a) Imagem Lena 512×512 aproximada por malhas triangulares, usando quantização com 5 passos e taxa de 0.0511 bpp, com PSNR de 25.52 dB; (b) a mesma imagem superposta por malhas.	30
3.13	(a) Imagem Mandrill 512×512 aproximada por malhas triangulares, usando quantização com 5 passos e taxa de 0.0516 bpp, com PSNR de 20.09 dB; (b) a mesma imagem superposta por malhas.	31
3.14	Imagem Lena codificada a 0.0311 bpp com (a) malhas triangulares (PSNR = 25.02 dB), e (b) SPIHT (PSNR = 25.96 dB).	32
3.15	Imagem Mandrill codificada a 0.0323 bpp com (a) malhas triangulares (PSNR = 19.87 dB), e (b) SPIHT (PSNR = 20.00 dB).	33
4.1	Dois passos de refinamento de um triângulo da malha.	36
4.2	Divisão de triângulos vizinhos. Os pequenos círculos indicam nós interpolados. Em (a), a divisão somente do triângulo central é sem efeito e não cria nenhum novo nó ou arco; em (b), a divisão dos dois triângulos vizinhos cria um novo nó e 6 novos arcos.	36
4.3	Nós e pixels na malha mínima (2×1 pixels) para uma imagem de 16×16 pixels.	38
4.4	Divisão de um par de triângulos por biseção, dobrando o número de triângulos resultante.	39
4.5	Possíveis simplificações em um losango: (a) até 2 triângulos maiores horizontais (superior e inferior); (b) até 2 triângulos maiores verticais (esquerdo e direito).	42
4.6	Resultados em taxa×distorção da codificação da imagem Lena 512×512 com malha hierárquica e com o codificador SPIHT.	49
4.7	Resultados em taxa×distorção da codificação da imagem Mandrill 512×512 com malha hierárquica e com o codificador SPIHT.	50
4.8	Resultados em taxa×distorção da codificação da imagem Lena 256×256 com malha hierárquica e com o codificador SPIHT.	51

4.9	(a) Imagem Lena 512×512 codificada a 0.4290 bpp com malha hierárquica (PSNR = 33.63 dB). (b) A mesma imagem superposta pela malha.	53
4.10	(a) Imagem Mandrill 512×512 codificada a 0.5482 bpp com malha hierárquica (PSNR = 24.05 dB). (b) A mesma imagem superposta pela malha.	54
4.11	(a) Imagem Lena 256×256 codificada a 0.3710 bpp com malha hierárquica (PSNR = 28.73 dB). (b) A mesma imagem superposta pela malha.	55
5.1	(a) Simplificação da malha com a união de um grupo de 4 triângulos; e (b) identificação dos nós e arcos das duas malhas.	62
5.2	Identificação dos nós e arcos candidatos a serem utilizados no cálculo das taxas de bits; (a) na malha do grupo de 4 triângulos; e (b) na malha do triângulo maior.	62
5.3	Simplificação de triângulo com (a) 1 triângulo acima e à esquerda já simplificado; (b) 2 triângulos acima, à esquerda e à direita, simplificados; e (c) 2 triângulos simplificados, um acima e à esquerda e outro abaixo.	64
5.4	Resultados da codificação da Lena 512×512	68
5.5	Resultados da codificação do Mandrill 512×512	68
5.6	Resultados da codificação da Lena 256×256	69
5.7	Resultados da codificação da Bridge 512×512	69
5.8	Resultados da codificação da Aerial 512×512	70
5.9	Resultados da codificação da PP1205 512×512	70
5.10	Resultados da codificação da PP1209 512×512	71
5.11	Imagem Lena 512×512 codificada com taxa-distorção, $\lambda = 13$, que determinou 5 passos de quantização dos nós, triângulos menores de 4×2 pixels, taxa de 0.366 bpp e PSNR de 33.32 dB.	72
5.12	Imagem Mandrill 512×512 codificada com taxa-distorção, $\lambda = 180$, que determinou 4 passos de quantização dos nós, triângulos menores de 4×2 pixels, taxa de 0.442 bpp e PSNR de 23.59 dB.	73
5.13	Imagem Lena 256×256 codificada com taxa-distorção, $\lambda = 11$, que determinou 4 passos de quantização dos nós, triângulos menores de 2×1 pixels, taxa de 0.811 bpp e PSNR de 31.86 dB.	74

5.14 Imagem Bridge 512×512 codificada com taxa-distorção, $\lambda = 100$, que determinou 4 passos de quantização dos nós, triângulos menores de 4×2 pixels, taxa de 0.485 bpp e PSNR de 25.88 dB.	75
5.15 Imagem Aerial 512×512 codificada com taxa-distorção, $\lambda = 100$, que determinou 4 passos de quantização dos nós, triângulos menores de 4×2 pixels, taxa de 0.443 bpp e PSNR de 26.54 dB.	76
5.16 Imagem PP1205 512×512 codificada com taxa-distorção, $\lambda = 50$, que determinou 4 passos de quantização dos nós, triângulos menores de 2×1 pixels, taxa de 0.760 bpp e PSNR de 24.75 dB.	77
5.17 Imagem PP1209 512×512 codificada com taxa-distorção, $\lambda = 50$, que determinou 4 passos de quantização dos nós, triângulos menores de 2×1 pixels, taxa de 0.811 bpp e PSNR de 27.21 dB.	78

Lista de Tabelas

3.1	Resultados da codificação da imagem Lena com malhas triangulares e comparação com o codificador SPIHT.	28
3.2	Resultados da codificação da imagem Mandrill com malhas triangulares e comparação com o codificador SPIHT.	29
4.1	Resultados da codificação da imagem Lena 512×512 com malha hierárquica e comparação com o codificador SPIHT.	47
4.2	Resultados da codificação da imagem Mandrill 512×512 com malha hierárquica e comparação com o codificador SPIHT.	47
4.3	Resultados da codificação da imagem Lena 256×256 com malha hierárquica e comparação com o codificador SPIHT.	48
5.1	Resultados da codificação da imagem Lena 512×512 com malha hierárquica e critério taxa-distorção para diferentes valores de λ	66
5.2	Resultados da codificação da imagem Mandrill 512×512 com malha hierárquica e critério taxa-distorção para diferentes valores de λ	66
5.3	Resultados da codificação da imagem Lena 256×256 com malha hierárquica e critério taxa-distorção para diferentes valores de λ	67

Capítulo 1

Introdução

Com a crescente difusão da comunicação de dados a distância, como na Internet, em comunicações móveis, em TV digital e DVD, a compressão de dados tem se tornado cada vez mais necessária. Nesse contexto, destacam-se a transmissão e o armazenamento de imagens, estáticas e em seqüências (vídeo), como sendo as aplicações que manipulam maior quantidade de dados e, portanto, demandam processos de compressão mais eficientes. Dependendo da aplicação, a compressão de imagens pode ser sem perda ou com perda. A compressão de imagens sem perda normalmente se restringe a aplicações médicas e às imagens que requerem posterior processamento para sua análise, como as obtidas por meio de satélites, radares e sonares. No entanto, na grande maioria das aplicações, são toleradas perdas que não prejudicam a qualidade da imagem codificada, mas que, por outro lado, reduzem muito a quantidade de dados necessários para representá-la. Atualmente, os codificadores de imagens com perda mais eficientes e mais usados utilizam a transformada DCT (transformada cosseno discreta) ou a transformada DWT (transformada wavelet discreta). Os coeficientes obtidos nessa transformação passam por um processo de quantização, que gera símbolos que, finalmente, são codificados por um codificador por entropia, como o Huffman ou o codificador aritmético.

Malhas poligonais são largamente utilizadas como estrutura de imagens sintéticas geradas em computador. Essas imagens normalmente procuram aproximar cenários do mundo real ou de ficção. Isso indica que pode ser interessante usar malhas na codificação de imagens naturais, onde se busca uma representação eficiente de imagens do mundo real. As malhas também podem ser usadas para represen-

tar imagens-distância (*range-images*) produzidas a partir de sensores direcionais de distância, como radares usados no mapeamento de terrenos a partir de satélites e de algumas câmeras fotográficas que registram também a informação de distância de regiões da cena fotografada.

Uma vantagem das malhas em relação à divisão em blocos usada na DCT é a continuidade dos seus elementos. Nas malhas, os pixels das fronteiras (arestas) entre pares de polígonos são comuns a ambos os polígonos e, assim, indesejáveis efeitos de blocos não ocorrem. A codificação por malhas é diretamente aplicada a imagens, sem transformação de domínio, e isto pode ser uma vantagem na codificação de vídeo, porque a mesma representação pode ser usada tanto nas imagens “intra” (quadros da seqüência codificados como imagens estáticas) quanto nas imagens “diferença” (diferença entre um quadro e sua previsão). Além disso, em seqüências de vídeo, a estimação de movimento pode ser obtida das malhas, em substituição aos blocos dos esquemas convencionais, e os vetores-movimento serem aplicados aos nós (vértices) das malhas. Estruturas de representação de imagens naturais por malhas foram recentemente estudadas para aplicação em codificação de imagens e vídeo com resultados que incentivam a continuidade da pesquisa nesta área.

Nesta tese, nos propomos a obter representações de imagens naturais por meio de malhas triangulares hierárquicas, e esquemas eficientes para quantização e codificação dessa informação, atendendo a critérios de taxa de compressão versus distorção das imagens. Inicialmente, a imagem é aproximada por uma malha fina, com triângulos bem pequenos. Os nós da malha ou vértices dos triângulos têm posições fixas e independentes da imagem codificada. As amplitudes desses nós são aproximações das intensidades dos pixels da imagem naquelas regiões. Os arcos da malha ou arestas dos triângulos podem ter suas direções originais alteradas objetivando melhor aproximar as bordas ou arestas da imagem. Uma vez otimizada a aproximação da imagem, a malha é simplificada gradualmente, de forma a economizar a quantidade de nós e arcos, produzindo uma malha triangular hierárquica. Finalmente, as amplitudes quantizadas dos nós, as direções dos arcos e a árvore descritiva da hierarquia são codificados por um codificador aritmético.

No capítulo 2, apresentamos uma introdução à representação e codificação de imagens com malhas poligonais e alguns trabalhos relevantes desenvolvidos nessa

área.

O capítulo 3 mostra o algoritmo de codificação por malhas fixas desenvolvido no início desta tese, que produziu resultados que motivaram a continuidade do desenvolvimento deste trabalho.

No capítulo 4, desenvolvemos uma estrutura hierárquica para as malhas triangulares, num algoritmo que procura simplificar a malha, substituindo conjuntos de triângulos pequenos por outros maiores, em regiões com menor variação nas intensidades dos pixels da imagem.

O critério taxa-distorção é apresentado no capítulo 5. Com a utilização deste critério, o codificador procura otimizar a alocação de bits aos vários itens que descrevem a malha, a partir de um parâmetro que relaciona a redução na distorção da imagem à quantidade de bits de código.

O capítulo 6 conclui este trabalho, comentando os resultados obtidos e apresentando pontos que poderão ser investigados em trabalhos futuros sobre o tema.

Capítulo 2

Representação de Imagens por Malhas

As malhas poligonais são normalmente usadas como uma forma de discretização de superfícies ou volumes. As superfícies são representadas, na maioria das vezes, por malhas de triângulos, quadriláteros ou hexágonos, e os volumes, por conjuntos de tetraedros ou hexaedros. Sua principal aplicação é no método de elementos finitos usado pela indústria em projetos e análises.

A grande maioria das imagens sintéticas, geradas por computador, é composta por objetos tridimensionais, e se utiliza a representação dos volumes desses objetos por malhas.

No caso das imagens naturais, que são as tratadas neste trabalho, considera-se a intensidade dos seus pixels como sendo a terceira dimensão (perpendicular ao plano definido pela imagem) na representação tridimensional de uma superfície. Na figura 2.1, apresentamos uma malha triangular que aproxima a imagem Lena, de 512×512 pixels, e intensidade de pixels de 0 a 255.

Quanto aos trabalhos já desenvolvidos sobre o tema, a grande maioria trata da geração e simplificação de malhas usadas para aproximar sólidos sintetizados em computador e suas imagens. Há um número bem menor de trabalhos sobre codificação de imagens naturais com malhas poligonais. Apresentaremos a seguir os trabalhos considerados relevantes para o desenvolvimento desta tese, [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12].

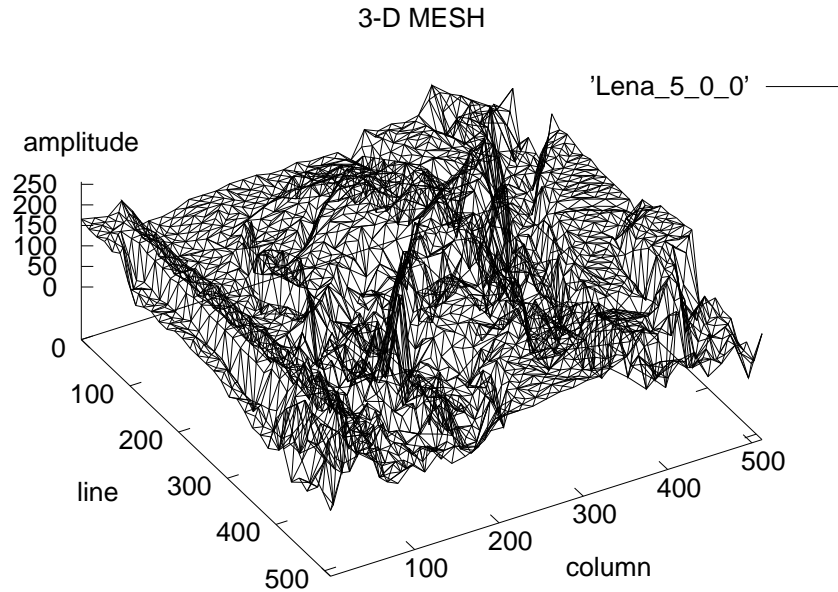


Figura 2.1: Exemplo de malha triangular que aproxima a imagem Lena 512×512 .

2.1 Malhas em imagens naturais

Os principais trabalhos encontrados que tratam da codificação de imagens naturais utilizando malhas são resumidos a seguir.

Em [1], LECHAT *et. al* propõem um codificador de imagens que usa como modelo de representação das imagens uma malha triangular definida no espaço bidimensional, na qual as amplitudes dos nós, que procuram aproximar as intensidades dos pixels da imagem nas suas regiões, constitui a terceira dimensão. A malha assim obtida forma uma superfície constituída por faces planas triangulares (*patches*) contíguas, unidas umas às outras por arestas comuns.

Nesse codificador, a função-objetivo a ser minimizada é o volume formado ao se superpor a malha do modelo à superfície gerada pela imagem bidimensional original, considerando a amplitude dos pixels da imagem original como sendo a terceira dimensão. O código gerado por esse codificador contém informação das posições e amplitudes de todos os nós da malha, após um processo de simplificação denominado *split-and-merge*.

Os resultados apresentados são relativos à aplicação do algoritmo a uma imagem não identificada, que não é usada como referência na grande maioria dos

trabalhos da comunidade de codificação de imagens. Além disso, nenhum resultado numérico é apresentado, apenas imagens.

Em [2], LECHAT *et. al* apresentam um codificador de imagens que usa também uma malha triangular como modelo de representação. Esta malha possui uma hierarquia entre os triângulos que a compõem, representando com triângulos menores as regiões da imagem que contêm maiores detalhes e com triângulos maiores, as regiões com menores variações nas intensidades dos seus pixels.

O codificador produz uma seqüência de bits progressiva, ou seja, gera um código denominado escalável, com ordenação dos bits por importância na redução da distorção na imagem reconstituída pelo decodificador. A otimização da malha se dá pelo ajuste das posições e amplitudes dos seus nós e pela escolha das direções das suas arestas.

Pelos resultados apresentados, fica claro que a flexibilização das posições dos nós da malha, de modo a saírem da grade fixa inicialmente definida, incute um custo muito alto em termos de taxa de bits, sem considerável retorno em redução da distorção. As imagens utilizadas nos testes também não são comumente usadas na avaliação de codificadores. Tratam-se de uma versão de 210×210 pixels do Mandrill, colorida, e de um quadro da seqüência Susie, em formato CIF (352×288 pixels), também colorida, com codificação independente dos componentes da luminância, Y, e das crominâncias, U e V. Na codificação do quadro da Susie, com 24Kbits de código, um codificador MPEG-4, com wavelets, obtém PSNR de 32.96dB, e, o codificador proposto, 31.90dB.

Em [3], LAURENT e ROBERT apresentam um codificador de vídeo também baseado numa malha triangular, para baixas taxas de bits. Um estimador de movimento é usado para vetores de movimento aplicados aos nós da malha. Os deslocamentos dos pixels internos a cada triângulo são interpolados a partir dos vetores de movimento dos seus três vértices ou nós.

Os resultados apresentados comparam o codificador proposto ao MPEG-4 (DCT) na codificação da seqüência Susie, no formato CIF, a 25 quadros/s e taxa de bits de 80Kb/s para o proposto e 160Kb/s para o MPEG-4. Esses resultados consistem de 1 quadro “intra” e 4 quadros obtidos por estimação e compensação de movimento, com a afirmação de que a codificação com malha apresenta menos

artefatos indesejáveis. Como não há comparação numérica, o tamanho das imagens é reduzido e a qualidade da representação das imagens no artigo obtido é bem baixa, a avaliação se tornou impossível.

Em [4], DEMARET e ROBERT propõem um codificador de imagens com malha triangular hierárquica combinado com DCT para taxas de bits muito baixas. A DCT é usada em *patches* triangulares da malha onde a interpolação linear e a divisão em triângulos menores não conseguem aproximar bem e eficientemente a imagem original naquelas regiões.

Resultados experimentais são obtidos com a codificação de um quadro da Susie, no formato CIF, usando duas versões do codificador proposto, uma normal e outra escalável, e três taxas de bits distintas. É mostrada uma comparação visual das imagens com resultados obtidos pela codificação com o algoritmo SPIHT [13] em duas taxas de bits. Novamente, o tamanho é muito reduzido e a qualidade das imagens no artigo obtido não permitiu qualquer análise.

Em [5], LAURENT *et. al* apresentam um algoritmo para evitar a superposição de triângulos na estimação de movimento baseada em malhas, com o uso de funções Lagrangianas aumentadas.

Resultados visuais da codificação da seqüência Horse com e sem o uso das funções propostas, bem como as imagens-erro são mostrados.

Em [6] e [7], JANSEN *et. al* propõem uma representação de funções bidimensionais *piecewise smooth*, como imagens naturais, por malhas triangulares multiresolução pela sua vantagem na aproximação das descontinuidades entre regiões com pouca variação de intensidade (*smooth*), em relação, por exemplo, às wavelets. Uma forma de subdivisão da malha denominada normal (ortogonal) é apresentada como uma triangulação eficiente.

Esse trabalho analisa também um problema clássico da triangularização que é a representação de bordas ou descontinuidades da imagem por triângulos verticais. Tratam-se de triângulos cuja projeção no plano da imagem é uma linha reta que contém os seus três vértices.

O resultado da aproximação da imagem Lena com o algoritmo proposto mostra a ótima representação das descontinuidades e das áreas com pequena variação de intensidade. Por outro lado, texturas com mais detalhes são muito mal represen-

tadas pela malha.

2.2 Malhas em imagens sintéticas e naturais

Alguns algoritmos foram testados tanto em imagens sintéticas como em imagens naturais, como o descrito em [8]. Neste trabalho, LU *et. al* apresentam um algoritmo de aproximação de imagens naturais, imagens-distância e imagens sintéticas por uma malha triangular adaptativa que usa o algoritmo *split-and-merge* para dividir e simplificar a malha original. O codificador da malha faz uma compressão denominada geométrica, com predição das coordenadas dos vértices baseada nas coordenadas de vértices vizinhos. Resultados são apresentados para 4 imagens distintas, uma gerada por computador, uma imagem-distância mais simples, uma imagem-distância mais complexa obtida por satélite e uma imagem natural.

2.3 Malhas em imagens sintéticas

Como dissemos, há uma infinidade de trabalhos que tratam de malhas poligonais e desses, um número também considerável analisa métodos de simplificação das malhas, como, por exemplo, em [9], [10] e [11].

Em [12], VELHO e ZORIN introduzem um esquema de subdivisão com triângulos isósceles retângulos denominado subdivisão 4-8. Esses valores, 4 e 8, correspondem às chamadas valências dos vértices da malha, que vêm a ser o número de arestas e faces que partem de cada vértice. No caso desta subdivisão, as valências se alternam entre os valores 4 e 8 em vértices subseqüentes da malha (ver figura 2.2).

O refinamento da malha nesse esquema é feito por divisões das faces (*patches*) em duas por meio da introdução de um novo nó no ponto central da aresta (hipotenusa) oposta ao ângulo reto da face triangular. A este processo, dá-se o nome de biseção. Este refinamento é mais gradual do que o refinamento mais comumente usado que é o de se dividir as faces em 4 menores, introduzindo um novo nó no ponto central de cada uma das três arestas da face original. Com a biseção, há um maior controle na resolução da malha obtida, pois a cada passo do refinamento o número de faces duplica ao invés de quadruplicar.

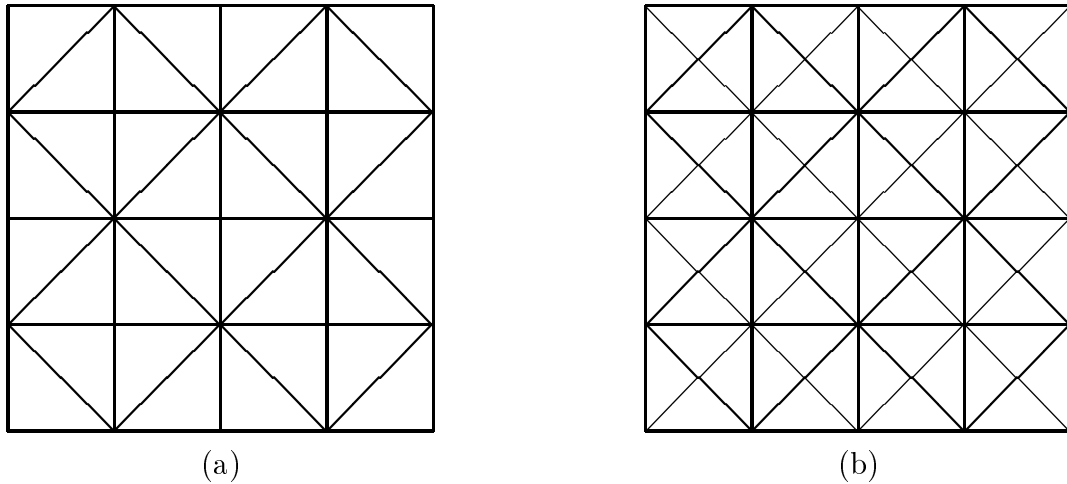


Figura 2.2: Malha em (a) refinada em (b), por um passo de biseções.

Em [14], VIEIRA *et. al.*, e em [15] e [16], VIEIRA e LOPES, introduzem uma simplificação de malha triangular denominada simplificação *Stellar*, uma implementação rápida do algoritmo de simplificação de conjuntos de 4 faces adjacentes que formam uma face triangular maior, na qual um método probabilístico substitui a fila de prioridades do algoritmo original. Também é adotada uma estrutura de dados denominada (*Corner-Table*, que se adequa bem à simplificação *Stellar* e à codificação das malhas. Um novo esquema de codificação e decodificação permite a transmissão e a compressão progressivas das malhas.

2.4 Definição da malha básica usada neste trabalho

Para formar uma malha necessitamos, inicialmente, escolher o tipo de polígono. Em computação gráfica, triângulos, quadriláteros e hexágonos são os mais comuns. Como o triângulo é o polígono mais simples e o mais flexível num espaço tridimensional (podemos mudar a posição de apenas um vértice e ele ainda continua a ser um triângulo, ou seja, continua definindo um plano onde está contida a face triangular, num espaço de três dimensões), ele foi o escolhido para este trabalho.

Escolhido o polígono, outra decisão deve ser tomada: devemos escolher entre adaptar a malha triangular à imagem original ou adaptar a imagem a uma malha triangular fixa predefinida. Em outras palavras, a malha pode mudar a posição

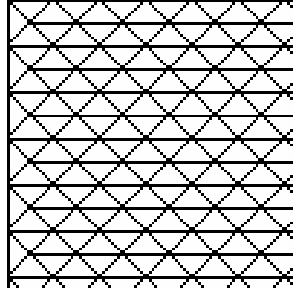


Figura 2.3: Esboço da malha usada.

dos seus vértices ou nós e arestas ou arcos para aproximar as bordas da imagem, ou a imagem tem suas bordas aproximadas pela malha fixa. A segunda opção foi escolhida neste trabalho: as imagens serão adaptadas a uma malha triangular com posição fixa. O motivo dessa escolha é apresentado na seção 2.5, a seguir.

O próximo passo é a definição do formato dos triângulos. Pelos motivos apresentados também na seção 2.5, a seguir, escolhemos os triângulos isósceles retângulos (com tamanho da base igual ao dobro da altura), posicionados de forma a gerar uma malha com triângulos idênticos, como esboçado na figura 2.3. Ressaltamos que denominamos triângulo da malha, a projeção no plano da imagem do triângulo ou face triangular da superfície da malha. Portanto, as formas e dimensões dos triângulos estarão sempre referenciadas a estas projeções e não às faces triangulares correspondentes.

A figura 2.4 mostra a malha usada neste trabalho e um passo de refinamento no qual cada triângulo é dividido por 4. A malha da figura 2.4(b) difere da malha da figura 2.2(b) apenas pelas direções de parte dos arcos denominados ortogonais (horizontais e verticais). Como esses arcos na malha usada por nós podem ser permutados, a malha da figura 2.2(b) corresponde a um caso particular da nossa malha.

2.5 Principais vantagens da malha escolhida

A malha triangular mais utilizada na representação de superfícies é a malha triangular regular composta por triângulos equiláteros, como em [2]. A malha usada neste trabalho, no entanto, é semelhante à descrita em [12]. Da mesma forma que

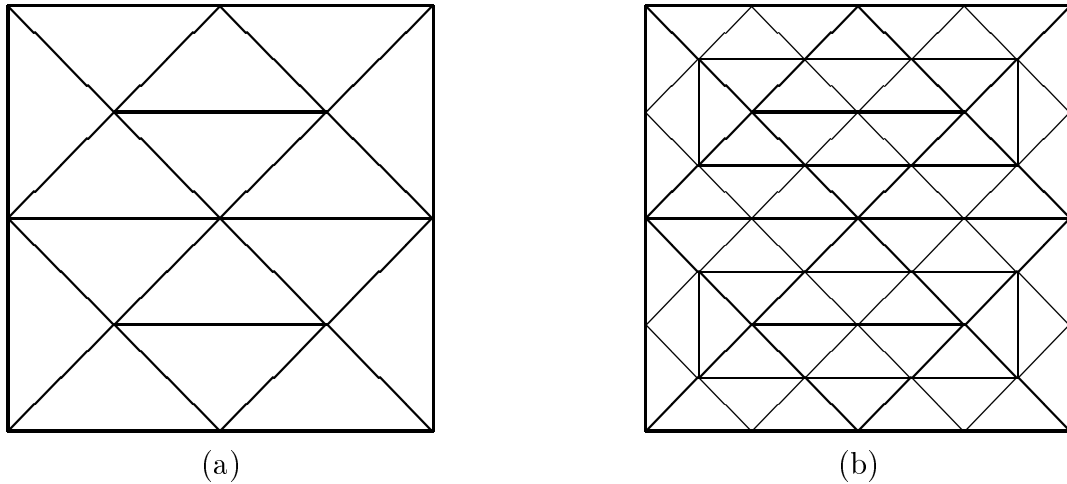


Figura 2.4: Malha triangular fixa com triângulos de 16×8 pixels para uma imagem de 32×32 pixels (a). (b) A mesma malha com os triângulos subdivididos em 4 menores, de 8×4 pixels.

ela, é composta por triângulos isósceles retângulos. Os motivos que nos levaram a escolher este tipo de malha, no lugar da malha regular com triângulos equiláteros, foram os seguintes:

- definição precisa da posição da malha e seus elementos em relação aos pixels da imagem
- grande valor de valência máxima dos seus nós
- possibilidade futura de se implementar a hierarquia por biseção

Com os triângulos isósceles retângulos, se usarmos valores inteiros para o tamanho das suas hipotenusas, poderemos, por exemplo, manter a mesma quantidade de pixels contidos em triângulo de um mesmo tamanho, o que não seria possível com os triângulos equiláteros. Isso implica em uma homogeneidade nas avaliações das variações de distorções provocadas por mudanças nos parâmetros da malha, tais como nas amplitudes dos nós e nas direções dos arcos. Todos os triângulos da malha, independente das direções dos seus arcos, envolverão a mesma quantidade de pixels.

Nas malhas regulares com triângulos equiláteros e possibilidade de permuta dos arcos, seus nós possuem valência máxima igual a 12. Nas malhas com triângulos isósceles, nas mesmas condições, a valência máxima é de 16. Isso significa que os

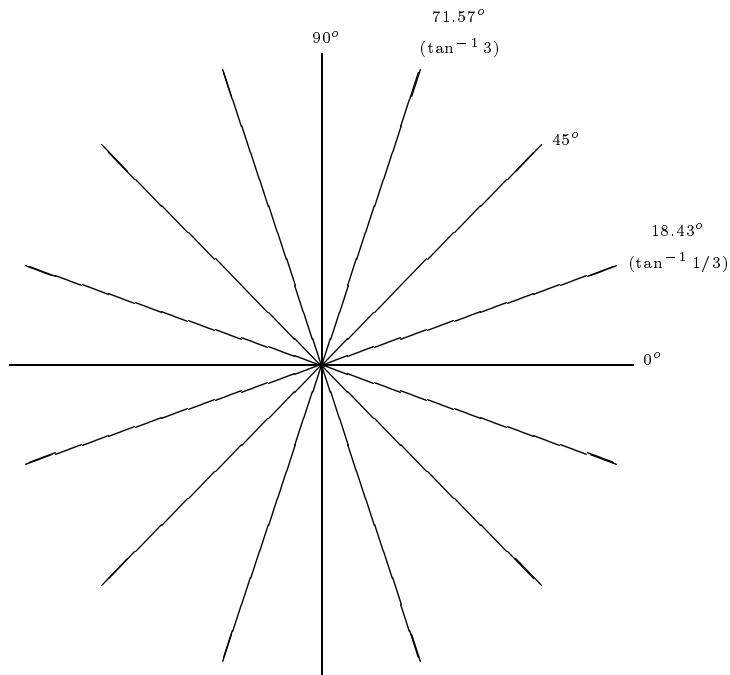


Figura 2.5: Possíveis direções de arcos que partem de um mesmo nó.

seus arcos podem assumir 16 diferentes direções nos 360° em torno de um nó. A desvantagem, no entanto, é que essas direções não são uniformemente distribuídas, como ocorre com os triângulos equiláteros. A figura 2.5 mostra as possíveis direções que os arcos da malha usada podem assumir.

Conforme [12], uma grande vantagem da malha escolhida é a da possibilidade de biseção, que, apesar de não ser usada no presente trabalho, poderá ser analisada em trabalho futuro, aproveitando-se grande parte dos conceitos aqui desenvolvidos e da implementação realizada.

Uma diferença, porém, entre as duas malhas, diz respeito à sua orientação em relação à imagem que ela representa. Na malha descrita em [12], inicialmente, as maiores arestas (hipotenusas) assumem as direções $\pm 45^\circ$ com a horizontal ou vertical e, como efeito da biseção, mudam essas direções a cada nível do refinamento. Na nossa malha, as hipotenusas assumem sempre as direções horizontal e vertical, independente do nível na hierarquia da malha. Dessa forma, na malha escolhida, pares de triângulos opostos com hipotenusas comuns formam quadrados inclinados de 45° ou losangos, como pode ser observado na figura 2.4. A escolha dessa orientação não teve nenhuma motivação especial, mas foi apenas uma opção de implementação.

Outra diferença entre a malha citada e a utilizada encontra-se na forma de refinamento da malha. No lugar da divisão por biseção, optamos pela divisão dos triângulos por quatro. O motivo para isso deveu-se ao sentimento de que esta opção proporcionaria melhor resultado em termos de taxa-distorção. Se usássemos a biseção, a árvore descritiva da hierarquia da malha possuiria, em média, o dobro do número de níveis em relação à divisão por quatro, para se chegar a erros médios quadráticos próximos na representação, o que indica que o custo da sua codificação seria consideravelmente aumentado. Por outro lado, com a biseção, teríamos um controle maior desse refinamento, evitando em alguns casos divisões que não trazem consideráveis melhorias em termos de taxa-distorção. No entanto, a divisão por quatro, como veremos no capítulo 4, não implica necessariamente em codificar mais três nós. Se um ou dois dos seus três triângulos vizinhos do mesmo nível hierárquico não forem divididos, teremos, respectivamente, dois ou apenas um novo nó. Quanto aos arcos (arestas dos triângulos da malha), sim, teremos que codificar as direções de todos os novos arcos criados, mas o custo de codificação da direção de um arco é, em geral, bem menor do que o da codificação de um nó.

Como já dissemos, optamos também por uma malha fixa, ou seja, seus nós não têm grande liberdade de se posicionar, a não ser nos pontos de uma grade predefinida. Apesar de dificultar a representação com muita fidelidade de detalhes como texturas com alta freqüência, a malha fixa permite que se mantenha a taxa de bits dentro de limites mais razoáveis do que as malhas flexíveis, como podemos observar nos resultados de [2] e [8]. Além disso, as operações de permuta de arcos são simplificadas, facilitando a detecção de situações proibidas como, por exemplo, as que produzem cruzamento entre arcos. O problema de triângulos verticais, citado na seção 2.1, também não ocorre na malha fixa.

No próximo capítulo descrevemos a implementação de um codificador de imagens com a malha descrita e tamanho de triângulos único. Os testes realizados com imagens a taxas de bits muito baixas mostraram resultados bem próximos aos de codificadores que usam transformada wavelet.

Capítulo 3

Codificação com Malha Triangular

Fixa

Para dar início à avaliação da codificação de imagens com a malha proposta, desenvolvemos um codificador simplificado, com uma malha composta por triângulos de tamanho único, sem a possibilidade de dividi-los ou agrupá-los. Com a malha utilizada, conseguimos apenas uma aproximação mais grosseira das imagens, possibilitando a codificação a baixas taxas. Nessas condições, os resultados em termos de erro quadrático foram muito bons, bem próximos aos obtidos por codificadores que utilizam transformada wavelet. Um resumo dos trabalhos descritos neste capítulo encontra-se em [17] e [18].

3.1 Descrição da malha

A malha usada nesta parte do trabalho é a definida na seção 2.4. Para esses testes iniciais, escolhemos triângulos com 16 pixels de base e 8 pixels de altura, por entendermos que já produziriam uma representação razoável de imagens de dimensões 512×512 pixels, por exemplo, e facilitariam posteriores subdivisões quando da implementação de hierarquia na malha. A figura 3.1 mostra detalhadamente o posicionamento dos nós dessa malha em relação aos pixels de uma imagem hipotética de 32×32 pixels, indicando que conjuntos de pixels pertencem a cada triângulo. É importante notar que os nós da malha, ou vértices dos triângulos, nunca coincidem com pixels da imagem.

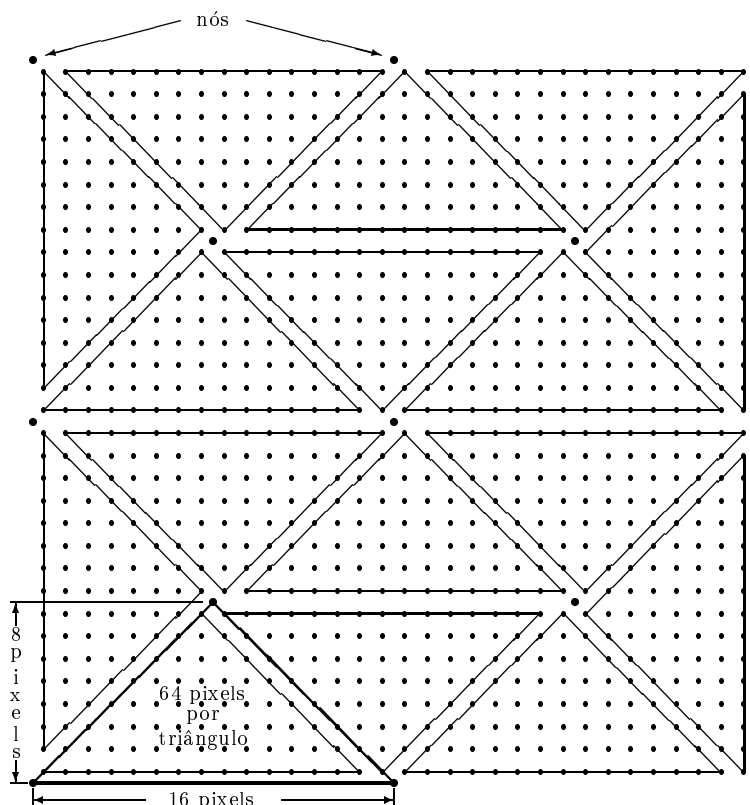


Figura 3.1: Posicionamento dos nós da malha em uma imagem de 32×32 pixels e separação dos conjuntos de pixels que são associados a cada face triangular da superfície gerada.

A figura 3.2 apresenta uma malha ampliada para uma imagem hipotética de 32×32 pixels, identificando os seus principais elementos.

3.2 Aproximação da imagem

O objetivo da codificação por malhas é, a princípio, aproximar imagens por uma malha, codificando apenas as amplitudes dos seus nós e estimando as amplitudes dos pixels da imagem a partir desses nós.

A malha original é definida como se constituindo de triângulos posicionados horizontalmente, ou seja, com sua base ou lado maior (hipotenusa) na horizontal e o terceiro vértice, alternadamente, acima ou abaixo da base. As exceções são os triângulos das extremidades esquerda e direita da malha, que são verticais, isto é, suas bases coincidem com as bordas das extremidades da imagem.

A mudança nas direções originais dos arcos das malhas melhora consideravelmente essa aproximação sem aumentar muito o custo de codificação [2] e, portanto,

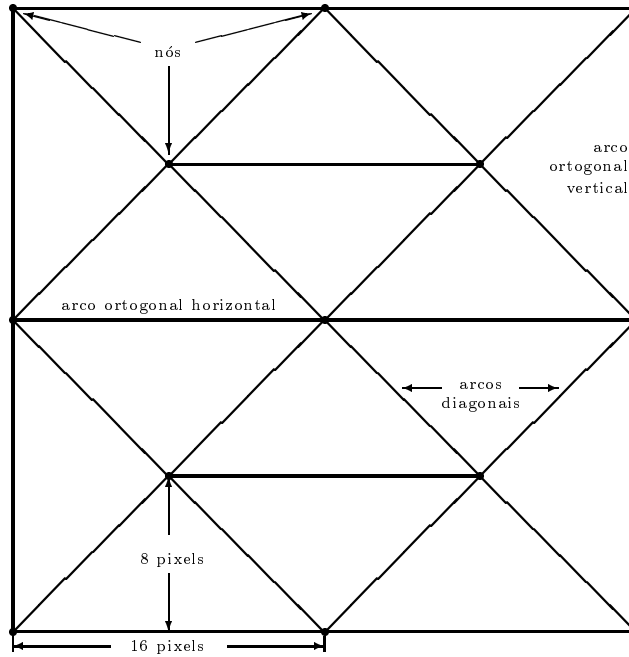


Figura 3.2: Malha triangular fixa com triângulos de 16×8 pixels para uma imagem de 32×32 pixels.

ela é analisada em cada arco e a sua informação adicionada ao código.

O processo de aproximação se inicia com uma estimativa da amplitude de cada nó. A seguir, os arcos dos triângulos são reposicionados de forma a otimizar a aproximação da imagem. Existe uma dependência muito grande entre esses dois parâmetros, ou seja, as direções ótimas dos arcos dependem das amplitudes dos nós e, vice-versa, as amplitudes ótimas dos nós dependem das direções dos arcos. Devido a isso, após o reposicionamento dos arcos, é necessário reajustar as amplitudes dos nós e, após esse reajuste, a repetição desse processo por algumas vezes melhora o resultado da aproximação. Isto deve ser feito até que a redução do erro seja bem pequena. Em nossos testes isto ocorreu após a terceira iteração.

Passos do processo de aproximação da imagem pela malha triangular:

- Estimativa inicial das amplitudes dos nós
- Executa 3 vezes:
 - Escolha das direções dos arcos ortogonais
 - Escolha das direções dos arcos diagonais
 - Otimização das amplitudes dos nós

3.2.1 Amplitudes iniciais dos nós

As amplitudes iniciais dos nós são estimadas a partir das amplitudes dos pixels na vizinhança de cada nó. A região considerada é um losango com diagonais horizontal e vertical do tamanho da altura do triângulo da malha. No caso da malha usada, com triângulos de 16×8 pixels, os pixels cuja média será usada na estimativa inicial da amplitude de cada nó estão contidos num losango com diagonais de 8 pixels de tamanho, como mostra a figura 3.3.

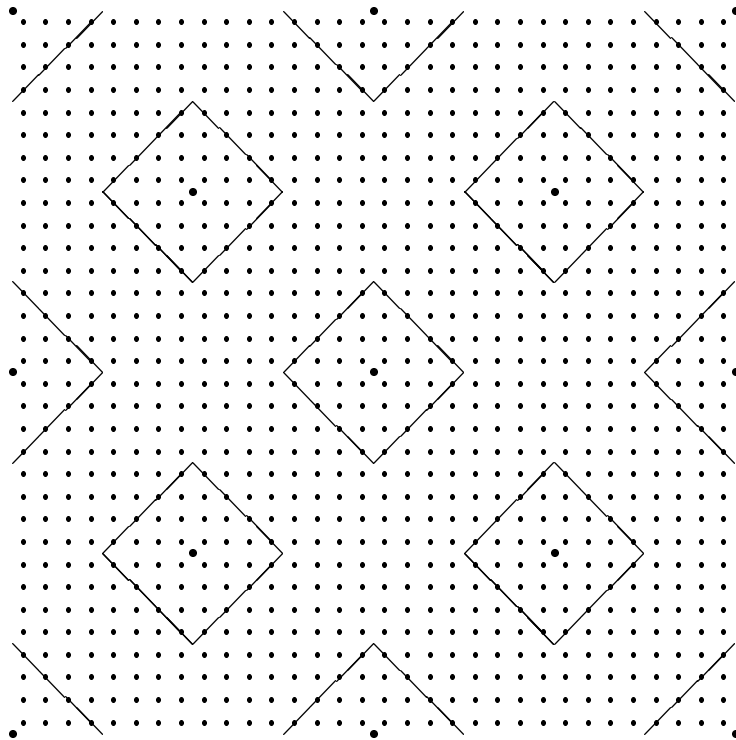


Figura 3.3: Losangos contendo os pixels que são usados no cálculo da média para a estimativa inicial das amplitudes dos nós na malha com triângulos de 16×8 pixels.

3.2.2 Interpolação para obtenção dos pixels

As amplitudes dos pixels internos a um triângulo da malha correspondem à terceira coordenada dos pontos da face plana triangular. Assim, para obtermos os seus valores, temos que obter esta coordenada (z) dos pontos (x, y, z) sobre a face triangular, onde x e y são as coordenadas dos pixels no plano da imagem.

Para isto, calculamos as coordenadas baricênticas de cada ponto a ser interpolado. Essas coordenadas são usadas como pesos dos valores das amplitudes nos

vértices do triângulo para interpolação dentro dele. Para um triângulo com vértices A, B e C e um ponto de interpolação P dentro dele, os pesos w_A , w_B e w_C são iguais às áreas dos triângulos PCB, APC e ABP, respectivamente, divididas pela área do triângulo ABC propriamente dito (ver figura 3.4). Esses pesos são usados na seguinte interpolação linear:

$$z(P) = z(A) * w_A + z(B) * w_B + z(C) * w_C$$

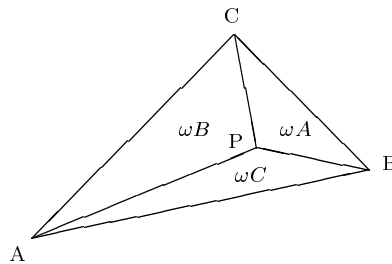


Figura 3.4: Triângulo ABC com ponto P interno, definindo três áreas, w_A , w_B e w_C .

Isto corresponde ao plano que passa pelos 3 vértices de um triângulo. A superfície interpolada resultante é plana dentro dos triângulos e nas fronteiras dos triângulos (arestas das faces) ela é contínua, mas não “suave” - o gradiente é descontínuo.

Nos testes com essa malha, as imagens resultantes têm aparência embossada e com defeitos em forma de pontas. Isto ocorre porque os triângulos são grandes e em posições fixas.

3.2.3 Permuta dos arcos

As limitações apresentadas nessa aproximação podem ser reduzidas com a permuta de triângulos da posição horizontal original para a posição vertical, como esboçado na figura 3.5.

O estado do arco ortogonal define se o triângulo está na posição horizontal ou vertical. Os chamados arcos ortogonais são as arestas das faces triangulares da malha cujas projeções no plano da imagem podem assumir as direções horizontal e vertical, ou seja, tratam-se das hipotenusas dos triângulos. Inicialmente, elas são

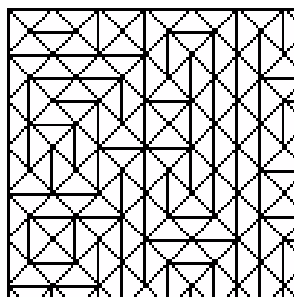


Figura 3.5: Permuta de arcos ortogonais (H/V).

posicionadas de forma a se obter a melhor aproximação da imagem original pela malha. A figura 3.6 mostra com detalhes a permuta de arcos ortogonais sobre os pixels de uma imagem de 32×32 pixels.

Esta permuta é realizada sempre que produzir uma redução no erro. Embora isto implique em *overhead* para informação de quais triângulos foram permutados, pode haver uma melhora global em termos de taxa-distorção.

Outra forma de melhorar a qualidade da imagem obtida é através da permuta de arcos diagonais, como esboçado na figura 3.7. Os chamados arcos diagonais são as arestas das faces triangulares da malha cujas projeções no plano da imagem podem assumir, inicialmente, as direções com ± 45 graus de inclinação em relação aos arcos ortogonais. Tratam-se dos catetos dos triângulos e, ao serem permutados, podem assumir as direções com $\pm 18, 43$ graus em relação à horizontal ou à vertical, conforme mostra a figura 2.5. Denominamos essas novas direções dos arcos diagonais como “mais horizontal” ou “mais vertical.”

A figura 3.8 mostra com detalhes a permuta de arcos diagonais sobre os pixels em uma imagem de 32×32 pixels.

Novamente, isto é feito nos casos em que há redução no erro. Da mesma forma, é necessário informar quais das possíveis permutas foram realizadas.

Dependendo das posições dos arcos ortogonais, permutas de diagonais vizinhas que possibilitariam considerável redução no erro podem se tornar proibidas. Por exemplo, se apenas uma das duas permutas ortogonais mostradas na figura 3.6, fosse efetuada, uma das permutas diagonais mostradas em 3.8, a do arco mais acima e à esquerda, seria proibida. Portanto, nesses casos a permuta torna-se permitida, através do que chamamos de flexibilização, e a direção do arco ortogonal que a

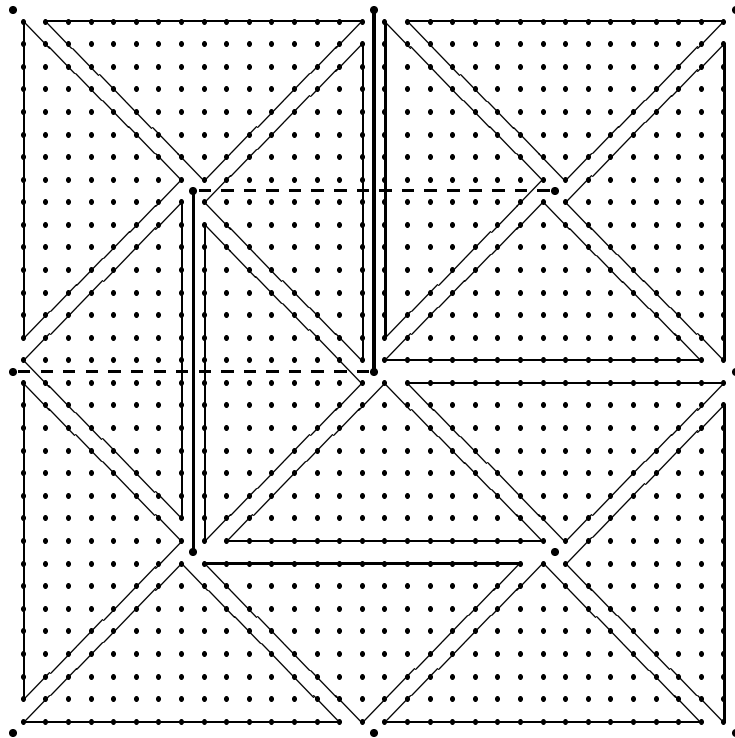


Figura 3.6: Permuta de dois arcos ortogonais horizontais para verticais.

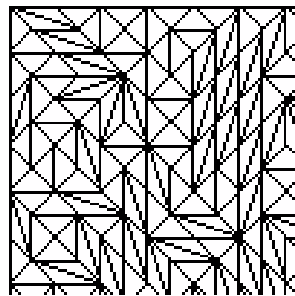


Figura 3.7: Permuta de vários arcos diagonais.

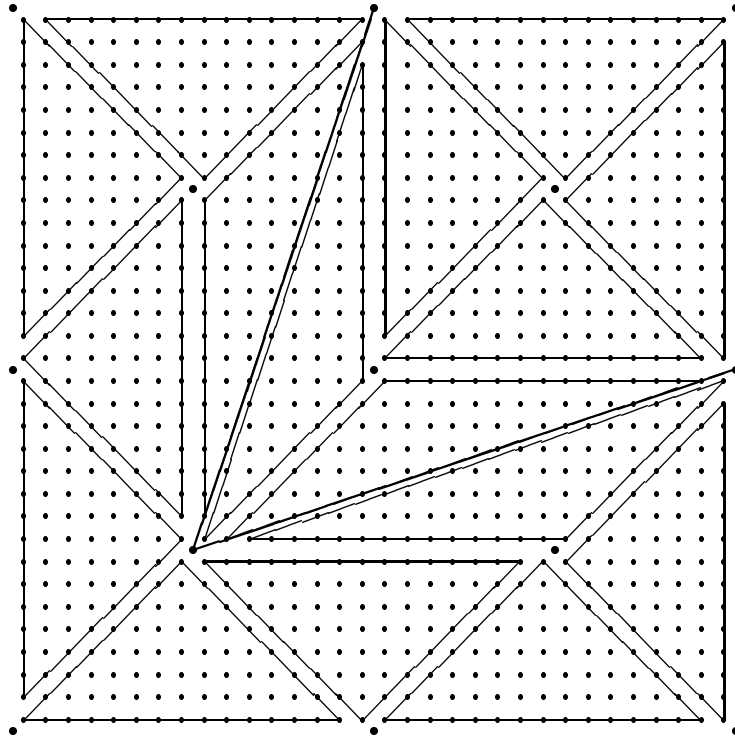


Figura 3.8: Permuta de dois arcos diagonais, um para a direção mais vertical e outro para a direção mais horizontal.

proibida é trocada.

Uma vez que a permuta de uma diagonal é tomada, as permutas de algumas diagonais vizinhas tornam-se proibidas. Portanto, é importante analisar primeiro o efeito de cada permuta de arco e, em seguida, executá-las por ordem decrescente de redução do erro quadrático.

Com as possibilidades de permutas de arcos apresentadas, um nó da malha poderá ter uma valência máxima de 16, ou seja, 16 arcos/triângulos que se originam nele, como pode ser observado na figura 2.5.

3.2.4 Otimização das amplitudes dos nós

A amplitude de cada nó é inicialmente escolhida como o valor médio dos pixels da sua vizinhança, como apresentado em 3.2.1. Após as permutas dos arcos, esses valores são ajustados para minimizar o erro quadrático na região afetada por cada permuta. Isto é feito por tentativa e erro, aumentando e diminuindo, separadamente, a amplitude inicial de cada nó. Como a mudança na amplitude de um nó interfere

com os valores ótimos dos nós vizinhos, esse processo deve ser feito através de algumas iterações. Testes mostraram que três iterações são suficientes para se obter convergência.

A seqüência de varreduras dos nós no processo de otimização é *raster-scan*, ou seja, da esquerda para a direita, linha a linha, de cima para baixo. O processo de ajuste da amplitude de cada nó se dá da seguinte forma:

- Soma 1 à amplitude do nó
- Calcula o erro quadrático da região afetada pelo nó e compara com o erro quadrático anterior
- Se não houver aumento no erro e enquanto isto acontecer:
 - Salva novo erro quadrático
 - Repete os passos anteriores, até um número máximo especificado. (Nos nossos testes, o número máximo usado foi 6.)
- Se houver aumento no erro no primeiro teste acima (na primeira vez que for somado 1 à amplitude do nó):
 - Repete todo o processo anterior, substituindo a “soma” do primeiro passo por “subtração”
- Se houver aumento no erro no primeiro teste com subtração (na primeira vez que for subtraído 1 da amplitude do nó):
 - Termina, sem mudar a amplitude do nó
- Em caso contrário, termina ajuste do nó

3.3 Quantização e codificação

Os valores das amplitudes dos nós e os estados dos arcos, permutados ou não, estão agora prontos para serem quantizados e codificados. No caso das amplitudes, foi usada quantização por aproximação sucessiva [19], muito semelhante à chamada codificação por planos de bits, seguida por codificação aritmética.

Os estados dos arcos são codificados diretamente por um codificador aritmético. Os estados dos arcos H/V são codificados primeiro, seguidos pelos estados dos arcos diagonais (levando-se em conta que eles não podem ser permutados em todas as posições). Dois modelos (histogramas) são usados em cada um dos dois tipos de arcos, visando explorar tendências dos arcos vizinhos e, conseqüentemente, melhorar a taxa de compressão.

3.3.1 Quantização das amplitudes dos nós

A quantização aplicada às amplitudes dos nós da malha é praticamente idêntica à usada por SHAPIRO em [19], para os coeficientes da transformada wavelet de imagens e denominada quantização por aproximação sucessiva.

Um limiar inicial T_0 é definido como sendo igual à metade da maior das amplitudes e é usado na comparação de todos os nós para determinar sua significância. Os nós com amplitudes maiores que T_0 são considerados significantes e geram um símbolo “1” e os demais, um símbolo “0”. Os nós significantes têm suas amplitudes subtraídas do valor T_0 . Em seguida, é usado um limiar $T_1 = T_0/2$, e assim por diante, até o último passo (N) de quantização determinado, com limiar T_{N-1} .

A partir do segundo passo, além da comparação para determinação da significância daqueles que ainda são insignificantes, todos os nós já considerados significantes em algum passo anterior são agora comparados com um valor igual à metade do limiar de significância para o passo, ou seja, $T_i/2$, gerando um símbolo de refinamento “1”, os que forem maiores que $T_i/2$, e “0”, os demais. Os nós cujo símbolo de refinamento for igual a “1” têm suas amplitudes subtraídas de $T_i/2$.

Essa seqüência de símbolos de significância e refinamento é codificada por um codificador aritmético e os valores do limiar inicial, T_0 , e do número de passos de quantização, N , acrescentados ao cabeçalho do código gerado.

Na decodificação, a mesma seqüência de limiares T_0, \dots, T_{N-1} é usada e as amplitudes quantizadas dos nós, reconstruídas. Todas as amplitudes dos nós são iniciadas com o valor 0, e eles considerados insignificantes. Quando um símbolo “1” é decodificado para um nó ainda insignificante, o valor do nó é acrescido de $1.5T_i$, ou seja, ele é reconstruído no centro do intervalo de incerteza. Em caso contrário,

símbolo “0” e nó insignificante, seu valor continua 0. Quando um nó é refinado, ou seja, já foi considerado significativo em algum passo anterior, um símbolo “1” faz com que seu valor seja acrescido de $3/4T_i$, ou seja, novamente no centro do intervalo de incerteza. Em caso contrário, símbolo “0” e nó significativo, seu valor é subtraído de $T_i/4$.

Modelos ou histogramas diferentes são usados para a codificação dos símbolos de significância e dos símbolos de refinamento.

No caso dos símbolos de significância, dois histogramas diferentes foram usados e escolhidos conforme as direções dos arcos ortogonais acima e à esquerda do nó a ser codificado e as significâncias dos nós acima e à esquerda. Se ambos os arcos são verticais e o nó superior é significativo ou se ambos os arcos são horizontais e o nó à esquerda é significativo, um histograma diferente é usado.

Para os símbolos de refinamento, apenas um histograma foi usado.

3.3.2 Interpolação dos nós

Ao invés de se codificar as amplitudes de todos os nós, podemos substituir algumas delas pelo valor médio das amplitudes dos pares de nós vizinhos, desde que o erro introduzido seja menor que um limiar predefinido. Isso, no entanto, acrescentaria um grande custo na sinalização de cada nó assim interpolado. Para reduzir esse custo, fazemos uma interpolação desse tipo em trios de nós, conforme mostrado na figura 3.9, somente quando o erro total introduzido for inferior a um limiar. Essa simplificação pode parecer, a princípio, tratar-se do início de uma hierarquização da malha, criando-se um nível acima do inicial. Se não pudesse haver permuta dos arcos em relação às suas direções originais, isso seria verdadeiro. No entanto, com a possibilidade dessas permutas, não podemos afirmar que essa simplificação garanta que o triângulo maior, traçado com linha mais grossa na figura 3.9, exista, ou seja, que a superfície por ele delimitada esteja toda num mesmo plano.

Nesses casos, indicamos os nós que não são codificados através de um mapa de nós que é codificado posteriormente. Para explorar os casos freqüentes em que dois ou mais triângulos adjacentes têm seus nós interpolados, usamos dois histogramas para codificar este mapa. Se pelo menos um triângulo adjacente e já “varrido” tiver sido interpolado, o histograma usado é diferente do usado no caso contrário.

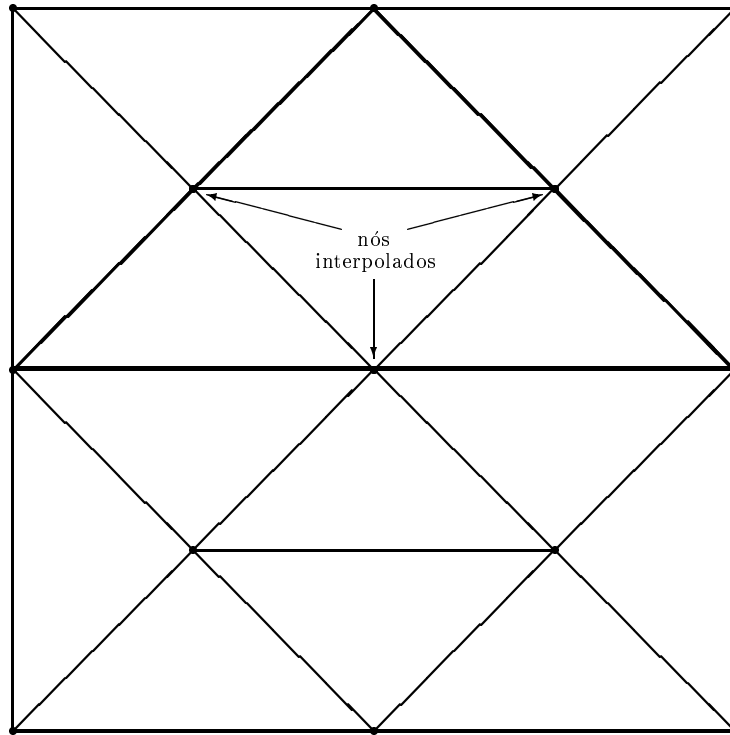


Figura 3.9: Malha com indicação de três nós que não são codificados, mas sim interpolados a partir de cada par de nós do triângulo maior que os contém.

Quando o erro é maior que o limiar, a amplitude do nó é quantizada e codificada normalmente, como descrito em 3.3.1.

3.3.3 Limiar para permuta dos arcos

Após a interpolação dos nós, os símbolos usados para indicar permutas de arcos passam a produzir grande *overhead* em relação ao código total gerado (*bit stream*). Em muitos casos a permuta de um arco não melhora consideravelmente a qualidade da imagem, mas contribui para aumentar a taxa de bits. Nesses casos, a redução no erro não compensa os bits gastos.

Para reduzir o problema, em termos de taxa-distorção, testamos alguns valores de limiares de permuta de arcos e realizamos somente as permutas que reduziam o erro quadrático de pelo menos estes limiares. Ao usarmos limiares para permutar ou não um arco, é importante escolher bem qual será a direção original, considerada não permutada. Esta direção original deve ser a que ocorre com maior freqüência na imagem toda, pois assim, o número de permutas será minimizado, reduzindo a

entropia dessa informação, com um número consideravelmente maior de casos sem permuta.

Para obter a direção ortogonal original, efetuamos um passo de otimização com permuta de arcos ortogonais com limiar igual a zero e contamos o número de arcos horizontais e verticais. Aquela que for mais freqüente passará a ser a direção original até o final da codificação. No caso dos arcos diagonais assumimos que as direções a 45 graus, ou seja, da malha original, são as suas direções originais.

No capítulo 5 descrevemos como essa decisão passa a ser baseada na redução do custo, visando a otimização em termos de taxa-distorção.

3.4 Resultados experimentais

Inicialmente, aproximamos as imagens Lena e Mandrill 512×512 com o uso da malha triangular fixa de 16×8 pixels. Em seguida, quantizamos as amplitudes dos nós usando 5 ou 4 passos e as codificamos com um codificador aritmético. Feito isto, testamos combinações de limiares para interpolação de nós e permuta de arcos ortogonais e diagonais. Os resultados obtidos são apresentados nas tabelas 3.1 e 3.2, cujas linhas correspondem aos seguintes testes:

1. quantização de nós com 5 passos, sem limiares de interpolação de nós e permuta de arcos;
2. quantização de nós com 4 passos, sem limiares de interpolação de nós e permuta de arcos;
3. quantização de nós com 4 passos, sem limiar de interpolação de nós, e com limiar de permuta de arcos correspondente à redução do erro quadrático em 8000;
4. quantização de nós com 4 passos, com limiar de interpolação de nós correspondente ao aumento no erro quadrático em 8000, e com limiar de permuta de arcos correspondente à redução do erro quadrático em 8000;
5. quantização de nós com 4 passos, com limiar de interpolação de nós correspondente ao aumento no erro quadrático em 15000, e com limiar de permuta de arcos correspondente à redução do erro quadrático em 10000;

6. quantização de nós com 4 passos, com limiar de interpolação de nós correspondente ao aumento no erro quadrático em 20000, e com limiar de permuta de arcos correspondente à redução do erro quadrático em 20000;

Os gráficos com os resultados de taxa×distorção obtidos na codificação das duas imagens nos 6 testes são apresentados nas figuras 3.10 e 3.11.

As imagens resultantes do primeiro teste, com quantização com 5 passos e com limiares de interpolação de nós e permutas de arcos iguais a 0, apresentam qualidade visual razoável, levando-se em conta as dimensões dos triângulos da malha.

No teste 2, com 4 passos de quantização e também sem limiares de nós e permutas de arcos, a taxa de bits já se reduz bastante sem tanta redução na relação sinal/ruído. Isto continua acontecendo nos testes subseqüentes, após a introdução de limiares nas amplitudes dos nós e/ou nas permutas de arcos, até a aproximação com os resultados de codificadores de alto desempenho, como o SPIHT. Isso indica que nessas condições a taxa-distorção é melhor. A qualidade visual das imagens resultantes, no entanto, se reduz consideravelmente.

As imagens mostradas nas figuras 3.12 e 3.13, juntamente com as estruturas em malha usadas nas suas aproximações, foram obtidas nos testes 1 das tabelas 3.1 e 3.2, respectivamente. Os resultados numéricos das suas codificações estão na primeira linha dessas tabelas.

As imagens nas partes (a) das figuras 3.14 e 3.15 foram codificadas com malhas e correspondem ao teste 5, penúltima linha das tabelas 3.1 e 3.2, respectivamente. As imagens na parte (b) das mesmas figuras resultaram da codificação com o codificador de imagens SPIHT [13], que usa transformada *wavelet*, com as mesmas taxas de codificação, e são mostradas para efeito de comparação.

3.5 Conclusões

Embora as imagens codificadas possuam defeitos pronunciados, principalmente as das figuras 3.14 e 3.15, deve ser levado em conta que as taxas de compressão usadas foram bem altas, da ordem de 256:1. Com interpolação de nós e limiar para permuta de arcos, as relações sinal/ruído (PSNR) obtidas são bem próximas das resultantes de codificadores baseados em *wavelet* [13] para as mesmas taxas de com-

Tabela 3.1: Resultados da codificação da imagem Lena com malhas triangulares e comparação com o codificador SPIHT.

Teste Num.	Q. Nós (passos)	Limiar Nós	Limiar Arcos	Código (bytes)	Taxa (bpp)	PSNR (dB)	SPIHT (dB)
1	5	0	0	1676	0.0511	25.52	27.61
2	4	0	0	1416	0.0432	25.37	27.06
3	4	0	8000	1178	0.0359	25.20	26.47
4	4	8000	8000	1080	0.0330	25.16	26.21
5	4	15000	10000	1021	0.0311	25.02	25.96
6	4	20000	20000	914	0.0279	24.78	25.60

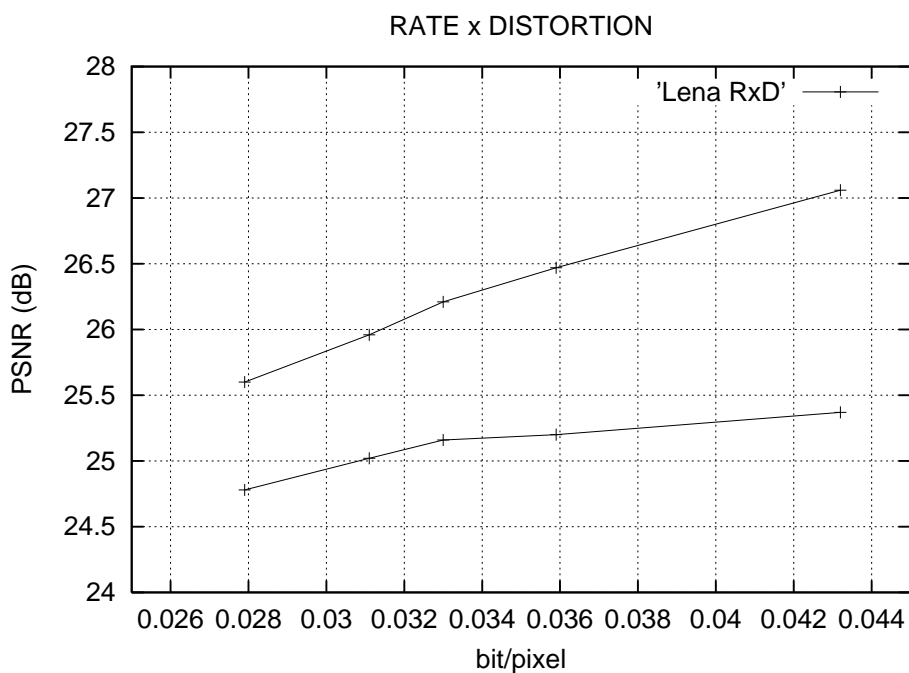


Figura 3.10: Resultados em taxa×distorção da codificação da imagem Lena com malhas triangulares e com o codificador SPIHT (curva superior).

Tabela 3.2: Resultados da codificação da imagem Mandrill com malhas triangulares e comparação com o codificador SPIHT.

Teste Num.	Q. Nós (passos)	Limiar Nós	Limiar Arcos	Código (bytes)	Taxa (bpp)	PSNR (dB)	SPIHT (dB)
1	5	0	0	1692	0.0516	20.09	20.53
2	4	0	0	1424	0.0435	20.06	20.22
3	4	0	8000	1227	0.0374	19.97	20.10
4	4	8000	8000	1182	0.0361	19.94	20.08
5	4	15000	10000	1070	0.0323	19.87	20.00
6	4	20000	20000	874	0.0263	19.74	19.86

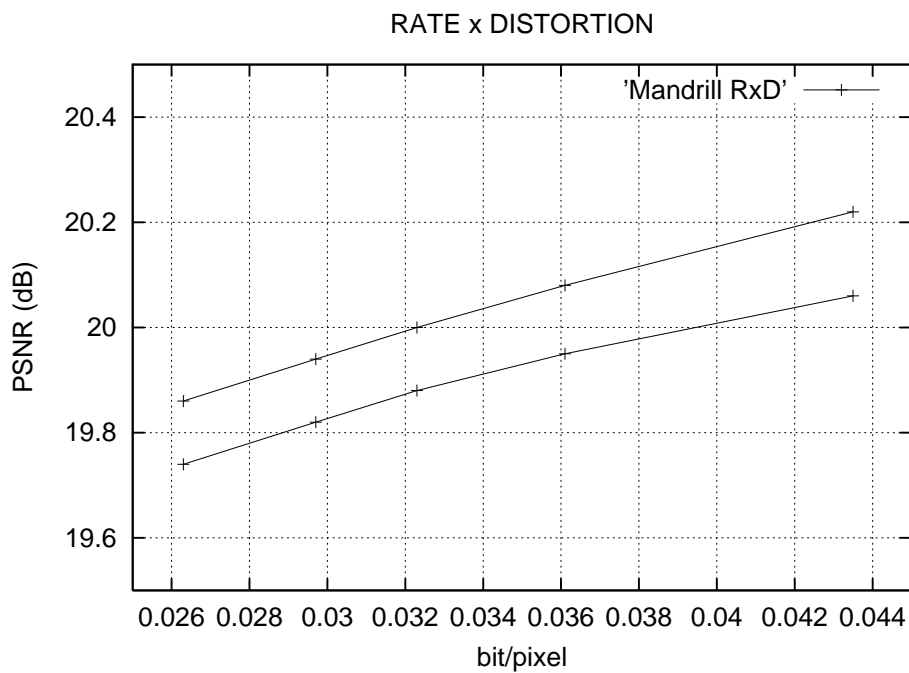


Figura 3.11: Resultados em taxa \times distorção da codificação da imagem Mandrill com malhas triangulares e com o codificador SPIHT (curva superior).



(a)

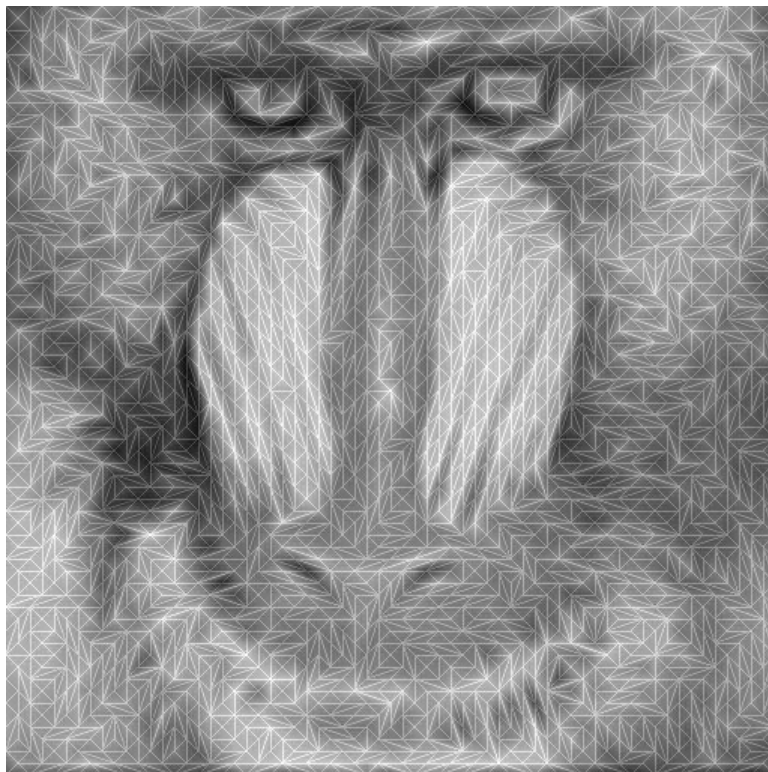


(b)

Figura 3.12: (a) Imagem Lena 512×512 aproximada por malhas triangulares, usando quantização com 5 passos e taxa de 0.0511 bpp, com PSNR de 25.52 dB; (b) a mesma imagem superposta por malhas.

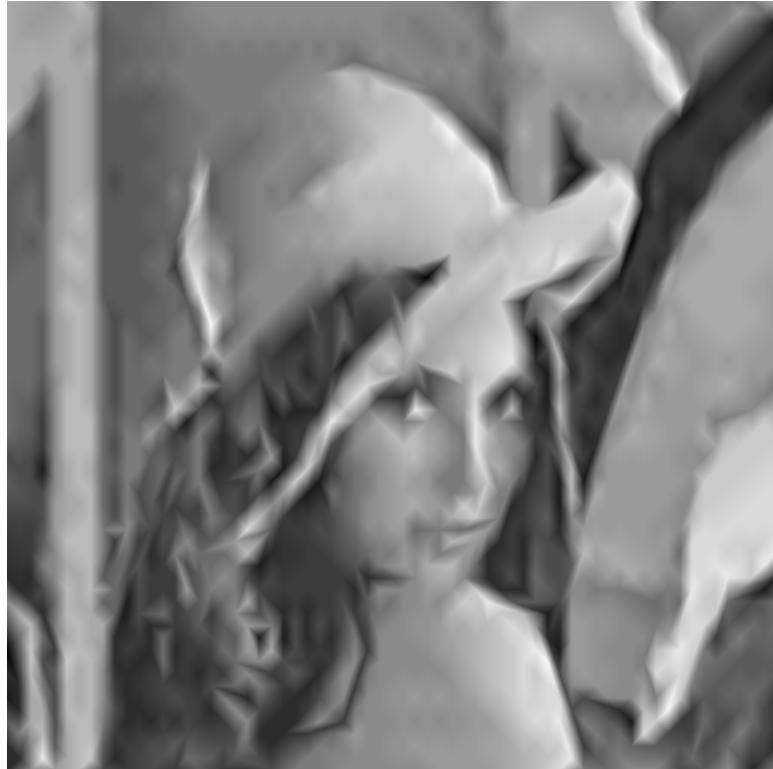


(a)



(b)

Figura 3.13: (a) Imagem Mandrill 512×512 aproximada por malhas triangulares, usando quantização com 5 passos e taxa de 0.0516 bpp, com PSNR de 20.09 dB; (b) a mesma imagem superposta por malhas.

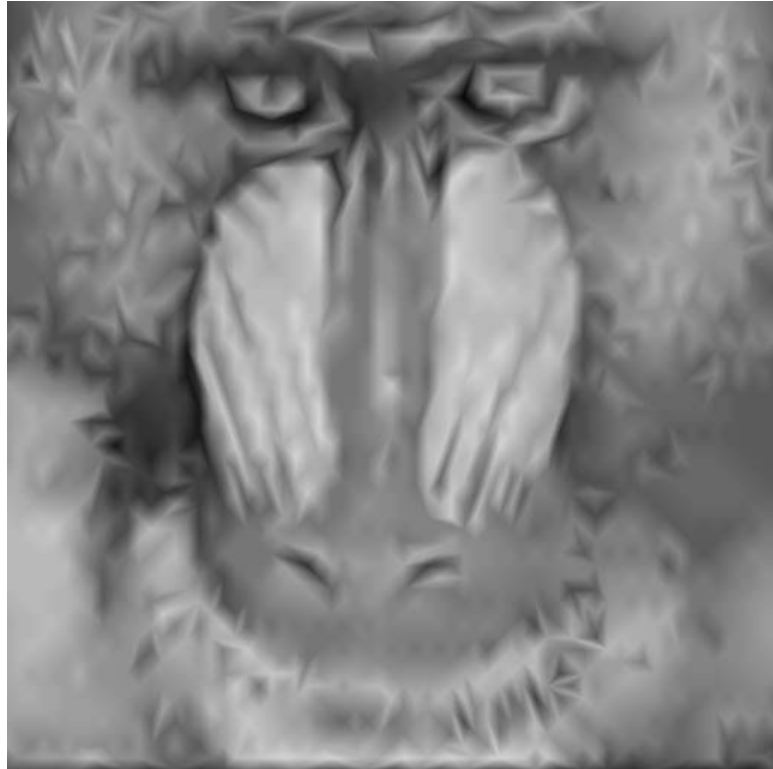


(a)



(b)

Figura 3.14: Imagem Lena codificada a 0.0311 bpp com (a) malhas triangulares (PSNR = 25.02 dB), e (b) SPIHT (PSNR = 25.96 dB).



(a)



(b)

Figura 3.15: Imagem Mandrill codificada a 0.0323 bpp com (a) malhas triangulares (PSNR = 19.87 dB), e (b) SPIHT (PSNR = 20.00 dB).

pressão, e a qualidade visual é comparável. Defeitos produzidos pelas malhas muitas vezes são menos incômodos do que defeitos do tipo *ringing* das wavelets na codificação com baixíssimas taxas de bits. No entanto, como a malha empregada possui triângulos muito grandes, detalhes de menor tamanho, como por exemplo, os olhos nas imagens testadas, ficaram muito distorcidos.

Capítulo 4

Codificação com Malha Hierárquica

A malha usada no capítulo 3 é de tamanho fixo, formada por triângulos de 16 pixels de base por 8 pixels de altura. Neste capítulo apresentaremos um algoritmo de divisão hierárquica da malha, de forma a alocar triângulos menores a áreas com maior variação nas intensidades dos pixels próximos (alta frequência) e triângulos maiores onde essa variação for menor (baixa frequência). O critério usado para essa escolha será o do erro médio quadrático da imagem aproximada em relação à mesma região na imagem original.

4.1 Refinamento da malha

O refinamento da malha é obtido pela divisão de cada triângulo que a compõe por 4, obtendo-se assim, 4 triângulos similares menores no lugar do original. Nesta divisão, um novo nó é inserido no ponto central de cada um dos três arcos do triângulo original, e as três linhas que os unem entre si, formam três novos arcos. Assim, a forma dos triângulos da malha se mantém a mesma. Esta divisão é mostrada na figura 4.1.

4.1.1 Problema da divisão em cascata

Ao se imaginar a divisão de um triângulo em 4 triângulos menores, surge uma questão. A vantagem buscada nesta divisão é, naturalmente, a de criarmos novos

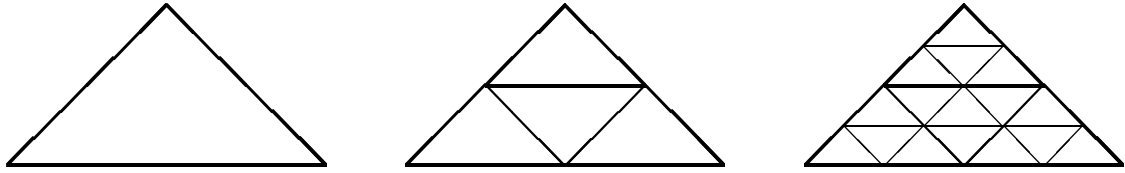


Figura 4.1: Dois passos de refinamento de um triângulo da malha.

nós e arcos para podermos adaptar melhor a intensidade dos pixels nas suas regiões. No entanto, se os três triângulos vizinhos¹, que possuem arcos comuns ao que está sendo dividido não forem também divididos por 4 menores, eles se manterão cada um em um plano e, portanto, os novos três nós criados pela divisão terão que ter amplitudes iguais às médias das amplitudes dos pares de nós adjacentes, fazendo com que o triângulo que está sendo dividido também permaneça em um plano, e isso corresponde a não dividi-lo. A figura 4.2(a) ilustra esta situação.

Isso pode nos levar a pensar que, ao dividirmos um triângulo, automaticamente, teríamos que dividir os três outros triângulos vizinhos, que teriam que ter os seus demais vizinhos também divididos, e assim por diante, de forma a termos toda a malha com triângulos menores.

Mas isso não ocorre. O fato de um triângulo ser dividido em 4 triângulos menores implica apenas que pelo menos um dos seus três vizinhos tenha que ser também dividido, como mostra a figura 4.2(b).

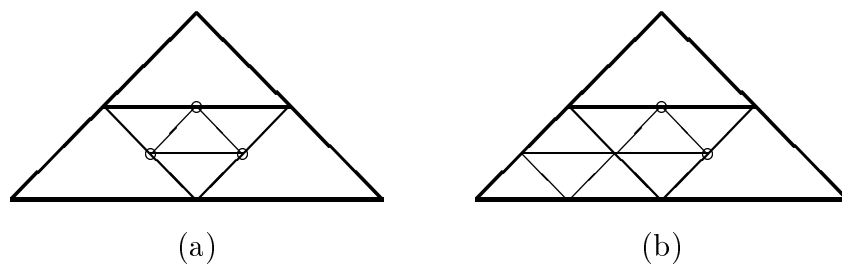


Figura 4.2: Divisão de triângulos vizinhos. Os pequenos círculos indicam nós interpolados. Em (a), a divisão somente do triângulo central é sem efeito e não cria nenhum novo nó ou arco; em (b), a divisão dos dois triângulos vizinhos cria um novo nó e 6 novos arcos.

¹triângulos com um lado comum

4.2 Seqüência da divisão hierárquica da malha

A hierarquia da malha pode ser obtida, a princípio, de duas maneiras distintas. De cima para baixo, iniciando-se com uma malha mais grossa e dividindo os triângulos conforme a necessidade de melhor aproximação, ou de baixo para cima, iniciando-se com a malha mais fina e simplificando-a onde isso não provocar uma redução expressiva na qualidade.

De forma a evitar que pequenas áreas triangulares fiquem com qualidade inferior à desejada, apesar de fazer parte de uma área triangular maior cujo erro médio é aceitável, ou seja, abaixo do limiar estabelecido, a opção escolhida foi a de se iniciar pela malha mais fina.

Outra vantagem desta opção será quando da implementação do critério de taxa-distorção na escolha dos tamanhos dos triângulos em cada região da imagem. Ao fazermos o teste de taxa-distorção num processo de refinamento de cima para baixo, podemos deixar de dividir triângulos cuja redução na distorção em função do gasto em bits de código está aquém do valor especificado, apesar de que, num passo futuro, a redução poderia ser tão grande que compensasse a divisão anterior, ou seja, atenderia globalmente o critério de taxa-distorção adotado. Naturalmente, poderíamos fazer os testes em mais de um nível antes de tomar a decisão sobre dividir ou não um triângulo, mas isso complicaria o algoritmo.

4.3 Definição da malha inicial

Inicialmente, aproximamos a imagem original por uma malha com triângulos pequenos. O tamanho desses triângulos é escolhido após testes com cada imagem e diferentes limiares de distorção. Mantendo a forma escolhida para os triângulos, se quiséssemos ter nós em todos os pixels da imagem, seria necessário que o menor triângulo da malha tivesse 1 pixel de base por 0,5 pixel de altura. No entanto, a codificação de malha tão fina, com quantidade tão elevada de nós a serem codificados, mais do que o número de pixels da imagem, tornaria muito ineficiente o código gerado. Por esse motivo, foi arbitrado que o menor triângulo da malha será de 2 pixels de base por 1 pixel de altura. Esta decisão incute um limite máximo na qualidade da representação das imagens pela malha. Mesmo assim, o número de

nós ainda é muito grande, aproximadamente igual à metade do número de pixels da imagem. Portanto, essa malha mais fina só deve ser usada quando se desejar uma qualidade de reconstrução bem alta e, ainda assim, em poucas regiões da imagem, sob pena de se elevar demais a taxa de bits da sua codificação. Quando se desejar codificar a baixas taxas de bits, ou seja, tolerando-se distorções maiores, a malha inicial deverá se constituir de triângulos maiores, como de 4×2 pixels ou 8×4 pixels.

A figura 4.3 mostra a distribuição de nós e pixels numa malha 2×1 .

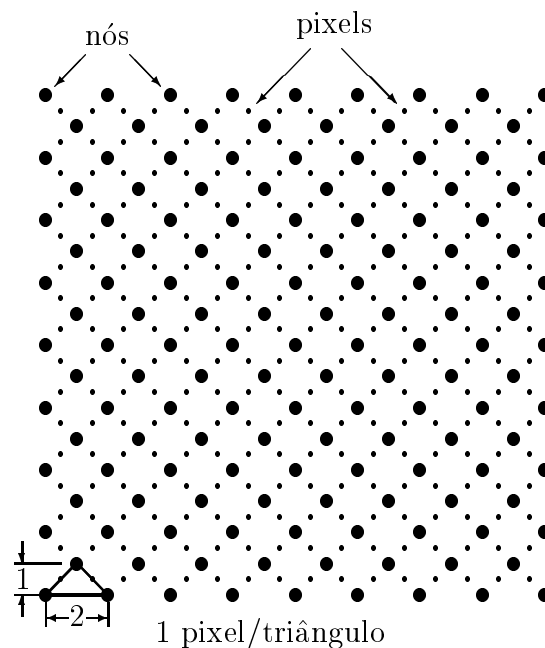


Figura 4.3: Nós e pixels na malha mínima (2×1 pixels) para uma imagem de 16×16 pixels.

Uma alternativa intermediária para a solução do problema de limite na qualidade determinado pelo tamanho mínimo de 2×1 pixels poderia ser a aplicação da divisão biseção [12] à malha mais fina. Na biseção, os triângulos são divididos por dois, através da inserção de um novo nó no ponto central das suas hipotenusas. Conseqüentemente, cada nó inserido produz a divisão dos dois triângulos opostos por esta hipotenusa comum (ver figura 4.4). No entanto, devido ao aumento na taxa de bits provocado pelo acréscimo de um novo nível na hierarquia da malha, além dos novos nós, essa divisão também não foi adotada.

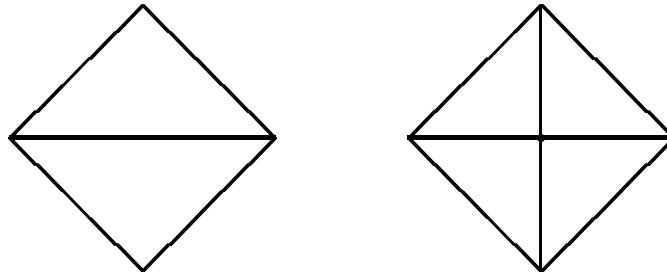


Figura 4.4: Divisão de um par de triângulos por biseção, dobrando o número de triângulos resultante.

4.3.1 Posição dos nós em relação aos pixels

A definição da malha foi feita de forma que os seus vértices ou nós não coincidam com a posição dos pixels, como pode ser observado na figura 4.3. O objetivo dessa escolha é o de fazer com que cada triângulo da malha tenha um conjunto bem definido de pixels a ele associados, ou seja, pertencentes ao plano da face definida por esse triângulo. Além disso, a quantidade de pixels associados a cada face triangular é sempre a mesma em um mesmo nível da hierarquia da malha, e igual ao valor da sua área.

4.3.2 Amplitudes iniciais dos nós

A amplitude inicial dos vértices ou nós é estimada a partir das amplitudes dos pixels da região em torno dele. Essa vizinhança é de tamanho proporcional ao tamanho dos triângulos da malha inicial. A amplitude estimada é a média dos pixels dessa região considerada.

4.3.3 Direções dos arcos

Na aproximação inicial da imagem pela malha, escolhamos as melhores direções de cada um dos arcos ortogonais (horizontais ou verticais) e as melhores direções de cada um dos arcos diagonais, em termos de erro quadrático. No caso dos arcos diagonais, nem todos são habilitados a se permutarem, dependendo das direções dos arcos ortogonais mais próximos. Esses arcos ortogonais podem ser permutados de forma a permitir as permutas diagonais, nos casos em que o erro final é reduzido.

4.3.4 Ajuste e quantização das amplitudes dos nós

Feito isto, as amplitudes dos nós são ajustadas de forma a reduzir o erro médio quadrático da aproximação em relação à imagem original. Nesse ajuste é feita uma busca ao valor da amplitude de cada nó que minimize o o erro na região por ele afetada. Os nós são ajustados na seqüência de varredura da malha, da esquerda para a direita, de cima para baixo. Exatamente como em 3.2.4, após o ajuste de todos os nós, o processo é repetido por três vezes, devido ao fato de o valor ótimo² de um nó depender dos valores dos nós vizinhos e também das direções dos arcos vizinhos.

Como no capítulo 3, essa seqüência de permuta de arcos e otimização de nós é repetida por três vezes, para se obter uma melhor aproximação. Obtida a malha otimizada, os nós são quantizados, novamente otimizados e a seqüência de de permuta de arcos e otimização de nós é executada uma última vez. Neste ponto, temos uma aproximação da imagem pela malha inicial, com todos os triângulos do mesmo tamanho, exatamente como foi feito no capítulo 3. A diferença é que agora os triângulos são menores.

4.4 Simplificação da malha

A malha mínima definida, uma vez otimizada para uma determinada imagem, proporciona uma aproximação com boa fidelidade em relação à imagem original. No entanto, a quantidade de elementos da malha que têm que ser codificados é extremamente grande, produzindo taxas-distorções não competitivas. Com o objetivo de reduzir essa grande quantidade de elementos, procuramos simplificar a malha nas regiões da imagem onde essa simplificação não resulta em aumento muito grande na sua distorção.

No capítulo 3, os elementos que descreviam a malha eram as amplitudes dos nós, as direções dos arcos ortogonais e diagonais. No entanto, nem todos os nós eram codificados. Já fazíamos uma simplificação para não termos que codificar todos eles. Em substituição às amplitudes desses nós, tínhamos que codificar as suas posições

²notar que, devido à interdependência entre os elementos da malha, os valores obtidos nas otimizações realizadas neste trabalho são sempre sub-ótimos

(suas amplitudes eram interpoladas a partir da amplitude de outros nós que eram codificados).

Na malha hierárquica, usamos um limiar de erro quadrático para determinar quais triângulos devem ser simplificados, ou seja, que conjuntos de 4 triângulos que, unidos, podem formar um triângulo maior, do nível 1 da malha, respeitando o limite de erro permitido. Com essa simplificação da malha, estaremos trocando a codificação de grupos de nós e arcos pela informação das suas posições.

Essas posições serão informadas através de uma árvore quaternária (*quad-tree*), na qual cada conjunto de 4 filhos de um nó correspondem aos 4 triângulos nos quais um triângulo maior pode ser dividido. A simplificação da malha é executada para todos os triângulos de um mesmo nível, antes de se passar para o nível subsequente.

4.4.1 Condições para a simplificação

O nível inicial da hierarquia, com a malha mais fina e os menores triângulos, é denominado nível 0. Conforme ocorre a simplificação, aumentando o tamanho dos triângulos e subindo na árvore, passamos aos níveis $1, 2, \dots, N - 1$.

No nível 0, não há nenhuma restrição para que se possa substituir um conjunto de 4 triângulos por um único triângulo do nível 1, cuja área seja a união das 4 áreas dos triângulos menores. Cada novo triângulo maior será composto por um quadrilátero (losango formado por um par de triângulos opostos pela hipotenusa comum) menor e mais dois triângulos menores. As direções do arco ortogonal interno ao losango e dos dois únicos arcos diagonais internos ao triângulo maior não interferem no processo de simplificação. Os demais arcos devem ser analisados para saber se suas direções impedem a obtenção de um triângulo maior, no nível 1 da malha. Caso isso ocorra, suas direções devem ser permutadas para que se possa obter o triângulo maior.

A partir do nível 1 da malha, para que possa haver a simplificação para o nível 2, é necessário naturalmente, que os 4 triângulos existam no nível 1, ou seja, que já tenha havido simplificações no nível 0 que produziram esses triângulos maiores no nível 1. Dependendo das direções dos arcos, haverá necessidade de que mais triângulos existam no nível 1 para viabilizar as eventuais permutas necessárias

previamente à simplificação.

4.4.2 Análise da variação na distorção

Para se tomar a decisão sobre a simplificação ou não de grupos de triângulos, esses grupos são analisados losango a losango (ou quadrilátero a quadrilátero). Em cada losango, desde que atendidas as condições descritas em 4.4.1, é testado o efeito da simplificação para um possível triângulo maior superior, inferior, esquerdo ou direito, isto é, apontando, respectivamente, para cima, para baixo, para a esquerda ou para a direita. A figura 4.5 ilustra essas simplificações. A variação na distorção é analisada em toda a área influenciada por cada possível simplificação, inclusive pelo efeito das permutas de arcos necessárias à sua viabilização.

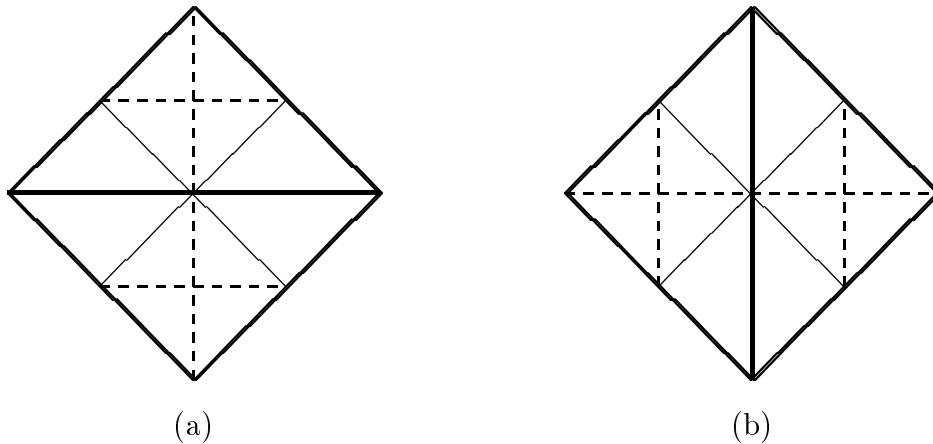


Figura 4.5: Possíveis simplificações em um losango: (a) até 2 triângulos maiores horizontais (superior e inferior); (b) até 2 triângulos maiores verticais (esquerdo e direito).

A princípio, esperava-se que a variação na distorção fosse sempre não negativa. No entanto, devido ao fato das amplitudes dos nós já estarem quantizadas, a simplificação permite que nós interpolados assumam valores que seriam impossíveis sem ela e que, em alguns casos, são mais próximos dos valores ótimos. Nesses casos, a simplificação da malha produz uma redução na distorção.

Para a determinação de que triângulos do losango serão simplificados, os seguintes critérios são usados:

- quando mais triângulos horizontais podem ser simplificados do que verticais,

ou vice-versa, eles são escolhidos independente da variação na distorção (desde que inferior ao limiar pré-estabelecido, naturalmente);

- se o número de triângulos horizontais e verticais que podem ser simplificados for igual, as variações totais na distorção em cada caso são comparadas e os triângulos da direção de menor variação são simplificados (levando-se em conta o limiar correspondente aos arcos ortogonais)

A preferência pela direção que viabilizar mais simplificações é justificada pela maior redução na taxa de bits. No capítulo 5 esta decisão se baseará na otimização em termos de taxa-distorção.

Todos os nós da malha que forem internos a triângulos obtidos pela simplificação de outros 4 menores deixam de ser codificados. O valor da amplitude de cada um desses nós é obtido pela média das amplitudes dos dois nós que são ligados pelo arco que tem este nó no seu ponto central. Essa interpolação é idêntica à descrita na seção 3.3.2 e mostrada na figura 3.9.

4.4.3 Repetição da Simplificação

A variação na distorção em função da simplificação de um grupo de triângulos não ocorre somente na região interna ao triângulo maior, mas também na região em torno dele, porque uma simplificação altera valores de nós e arcos que estão nessa área externa. Portanto, uma vez efetuada uma simplificação, grupos de triângulos já testados e não simplificados podem passar a atender ao critério de distorção máxima exigido para tal. Então, após concluída a aplicação do algoritmo de simplificação da malha em um nível, repetimos mais duas vezes sua aplicação em todos os triângulos ainda não simplificados usando o mesmo limiar de distorção. Finalmente, com o objetivo de simplificar triângulos divididos isolados, como o mostrado na figura 4.2(a), aplicamos mais uma vez o algoritmo, agora com limiar igual a 0.

4.4.4 Malha multinível

Como opção de implementação, são mantidas estruturas de dados para malhas completas, cobrindo toda a imagem, em todos os N níveis possíveis na hierarquia. Assim, quando há uma simplificação na malha, os triângulos menores, do

nível que foi deixado para trás, são alinhados em conformidade com o novo triângulo maior. Devido ao aumento na complexidade do programa com a implementação da malha hierárquica, isto foi usado durante os testes e depurações e mantido na versão definitiva para uma espécie de detecção de erros do programa. Tanto nas simplificações quanto nas permutas de arcos diagonais dos níveis acima do nível 0, as distorções são calculadas no nível i no qual se está analisando e também no nível $i - 1$ imediatamente abaixo, para verificação de consistência. Naturalmente, isso produz um *overhead* no tempo de execução do programa de codificação. A gerência dessas estruturas de dados também foi mantida pensando numa possível futura implementação, na qual uma imagem possa ser toda aproximada simultaneamente por malhas de diferentes resoluções que seriam analisadas numa decisão final sobre que resolução usar em cada região da imagem. As estruturas de dados usadas se consistem de vetores multidimensionais. Não são usadas listas encadeadas, a não ser no processo de ordenação de arcos diagonais, onde se usa árvore binária.

4.4.5 Direção dos arcos

Para permitir a simplificação de um triângulo maior, os arcos dos seus triângulos menores que agora formarão o contorno desse triângulo maior precisam ser alinhados. Os dois arcos ortogonais devem assumir a direção do triângulo maior, horizontal ou vertical, e os 4 arcos diagonais, as suas posições originais sem permutas, ou seja, a ± 45 graus com a horizontal ou vertical. Assim, após a simplificação de triângulos de um determinado nível i , todos os novos triângulos do nível $i + 1$ encontram-se nas suas formas originais, sem permutas diagonais.

As direções dos arcos ortogonais dos novos níveis já foram escolhidas no processo de análise da variação da distorção em 4.4.2. Os arcos diagonais, no entanto, estão todos nas direções originais, sem permuta. Num processo semelhante ao usado nas permutas diagonais do nível 0, os arcos diagonais dos níveis superiores também são permutados, imediatamente após a simplificação de triângulos em cada nível. Sempre que houver permuta em arcos de nível maior que 0, os arcos dos níveis inferiores são também alinhados, recursivamente, para manter a consistência da malha multinível. Além disso, nós posicionados no centro de arcos diagonais permutados em níveis maiores que 0 são interpolados a partir do novo par de nós dos extremos

do arco que passa por ele.

A seqüência de simplificação de triângulos e permutas diagonais é repetida, nível a nível, até que não seja possível simplificar mais ou que se atinja um nível máximo predefinido. Nos nossos testes, limitamos esse número de níveis a $N = 5$.

4.5 Codificação da estrutura da malha

A malha assim criada possui uma estrutura em árvore, na qual cada nó possui 4 “filhos” ou nenhum. A codificação dessa malha é feita a partir do nível mais alto (nível $N - 1$, onde N é o número de níveis da malha), até o nível mais baixo (nível 0). Embutidos na codificação da hierarquia, vão os símbolos que informam as direções dos arcos ortogonais em cada um dos pares de triângulos (losangos) de cada nível da malha. Em qualquer nível, cada losango pode possuir de 0 a 2 triângulos simplificados (não divididos). Caso pelo menos um triângulo do losango exista no nível, é necessário que codifiquemos sua direção, horizontal (superior e inferior) ou vertical (esquerdo ou direito). A seqüência de codificação da hierarquia a partir de cada losango do nível $N - 1$ mais alto é a seguinte:

- Primeiro triângulo do par (superior ou esquerdo):
 - testar o primeiro triângulo do par e codificar seu estado, dividido ou não
 - se não dividido, codificar sua direção, horizontal (se for superior) ou vertical (se for esquerdo)
 - se dividido, codificar os 4 triângulos menores que o compõem
- Segundo triângulo do par (inferior ou direito):
 - testar o segundo triângulo do par e codificar seu estado, dividido ou não
 - se não dividido e o primeiro tiver sido dividido, codificar a direção do segundo triângulo, horizontal (se for inferior) ou vertical (se for direito)
 - se dividido, codificar os 4 triângulos menores que o compõem

A codificação por entropia desses símbolos é feita através de um codificador aritmético que usa dois modelos para a informação de divisão ou não dos triângulos

e um modelo para a codificação das direções (nos losangos em que pelo menos um triângulo existe), para cada nível da árvore, totalizando $3N$ modelos diferentes.

A condição para a escolha do modelo alternativo na codificação da informação de divisão ou não dos primeiros triângulos de cada par (losango) é a de que o segundo triângulo do losango imediatamente acima e à esquerda tenha sido simplificado (não dividido). A condição para a escolha do modelo alternativo na codificação da informação de divisão ou não dos segundos triângulos de cada par (losango) é a de que o primeiro triângulo que faz par com ele no losango tenha sido simplificado. Este condicionamento reduz o código da hierarquia em mais de 15%.

Os símbolos que informam as direções dos arcos diagonais em todos os níveis da malha, bem como as amplitudes quantizadas dos nós, são codificados separadamente. Os condicionamentos usados na codificação aritmética são os mesmos do codificador do capítulo 3.

4.6 Resultados experimentais

Testamos o codificador com malha hierárquica em algumas imagens de teste e apresentamos os resultados a seguir. Para codificar uma imagem necessitamos informar ao codificador os seguintes parâmetros:

1. tamanho da malha mínima (2, 4, 8, ...)
2. número de passos do quantizador (de 1 a 8)
3. limiar para permuta de arcos ortogonais (em erro quadrático)
4. limiar para permuta de arcos diagonais (em erro quadrático)
5. limiar para simplificação de grupos de 4 triângulos (em erro quadrático)

Iniciamos os testes com os mesmos parâmetros para todas as imagens testadas. O número máximo de níveis da malha usado foi 5 para todos os testes. Analisando as variações de tamanho do código gerado e qualidade da imagem, passamos a ajustar os parâmetros do codificador separadamente para cada imagem. A seguir apresentamos os resultados desses testes. O codificador proposto, com malha

hierárquica, é identificado por *mesh-h*. Os resultados da codificação com os algoritmos SPIHT [13], MMP [20](para as imagens testadas neste trabalho) e JPEG são também apresentados para efeito de comparação.

A tabela 4.1 e o gráfico da figura 4.6 mostram os resultados obtidos com a imagem Lena 512×512 pixels. A imagem codificada correspondente ao teste num. 7 encontra-se na figura 4.9.

A tabela 4.2 e o gráfico da figura 4.7 mostram os resultados obtidos com a imagem Mandrill 512×512 pixels. A imagem codificada correspondente ao teste num. 1 encontra-se na figura 4.10.

A tabela 4.1 e o gráfico da figura 4.8 mostram os resultados obtidos com a imagem Lena 256×256 pixels. A imagem codificada correspondente ao teste num. 1 encontra-se na figura 4.11.

Tabela 4.1: Resultados da codificação da imagem Lena 512×512 com malha hierárquica e comparação com o codificador SPIHT.

Teste Num.	Tam. Tri.	Pass. Qntz.	Lim. Simpl.	Lim. Ortog.	Lim. Diag.	Árvore (bytes)	Diag. (bytes)	Nós (bytes)	Código (bytes)	Taxa (bpp)	PSNR (dB)	SPIHT (dB)
1	4	4	1500	100	100	2473	1280	4229	7982	0.2436	31.79	33.98
2	4	5	900	60	60	2664	1470	6181	10315	0.3148	32.93	35.15
3	2	5	900	60	60	3963	1783	8064	13810	0.4214	32.76	36.47
4	2	5	450	30	30	5118	2821	11756	19695	0.6010	34.39	38.02
5	4	5	450	30	30	3178	1825	7597	12600	0.3845	33.43	36.07
6	2	6	450	30	30	5146	2812	14717	22675	0.6920	34.73	38.67
7	4	6	450	30	30	3062	1795	9202	14059	0.4290	33.63	36.55

Tabela 4.2: Resultados da codificação da imagem Mandrill 512×512 com malha hierárquica e comparação com o codificador SPIHT.

Teste Num.	Tam. Tri.	Pass. Qntz.	Lim. Simpl.	Lim. Ortog.	Lim. Diag.	Árvore (bytes)	Diag. (bytes)	Nós (bytes)	Código (bytes)	Taxa (bpp)	PSNR (dB)	SPIHT (dB)
1	4	4	1500	100	100	4302	3095	10565	17962	0.5482	24.05	26.08
2	4	5	900	60	60	4313	3415	14835	22563	0.6886	24.23	27.13
3	2	4	900	60	60	8082	8944	31109	48135	1.4690	27.66	31.99
4	8	4	3000	200	200	1158	876	2945	4979	0.1519	21.20	22.06
5	8	4	1500	100	100	1130	948	3157	5235	0.1598	21.23	22.16

Tabela 4.3: Resultados da codificação da imagem Lena 256×256 com malha hierárquica e comparação com o codificador SPIHT.

Teste Num.	Tam. Tri.	Pass. Qntz.	Lim. Simpl.	Lim. Ortog.	Lim. Diag.	Árvore (bytes)	Diag. (bytes)	Nós (bytes)	Código (bytes)	Taxa (bpp)	PSNR (dB)	SPIHT (dB)
1	4	4	1500	100	100	836	513	1690	3039	0.3710	28.73	31.06
2	4	5	900	60	60	869	567	2355	3791	0.4628	29.43	32.26
3	2	5	900	60	60	1502	962	4066	6530	0.7971	31.77	35.92
4	4	4	3000	200	200	738	424	1409	2571	0.3138	28.12	30.08
5	8	4	1500	100	100	267	231	806	1304	0.1592	25.27	27.11

Os testes estão apresentados nas tabelas na seqüência em que eles foram realizados. Para efeito de comparação, os três primeiros testes de todas as imagens foram realizados com os mesmos parâmetros de entrada para o codificador. A partir do quarto teste, baseados nos resultados dos anteriores, os parâmetros foram modificados de forma diferente para cada uma das imagens.

Os valores dos parâmetros de codificação para os testes iniciais foram escolhidos baseados em inúmeros testes realizados anteriormente. O fato dos valores dos limiares dos dois tipos de arcos serem idênticos se deve à observação de que eles gastam aproximadamente a mesma quantidade de bits, em média, de código, e a área afetada pelas suas permutas é idêntica. É importante notar que o limiar para arcos ortogonais só é usado no nível mais baixo da malha (nível 0), pois as suas direções nos níveis superiores é determinada pelo procedimento descrito em 4.4.2, no qual este limiar não é levado em conta.

Quanto ao limiar de simplificação dos grupos de triângulos, foi estimado e comprovado por alguns testes, que o valor 15 vezes maior que os limiares dos arcos, corresponde aproximadamente à relação entre a quantidade de bits economizada na simplificação de um grupo de 4 triângulos e a quantidade de bits gasta numa permuta de arcos.

Após inúmeros testes com diferentes imagens e parâmetros, identificamos, empiricamente, os valores desses parâmetros mais adequados para cada imagem, em determinadas situações de taxa ou distorção.

Em todas as imagens testadas, ficou aparente a ineficiência da codificação da malha mais fina, com triângulos 2×1 . Principalmente no terceiro teste (terceira

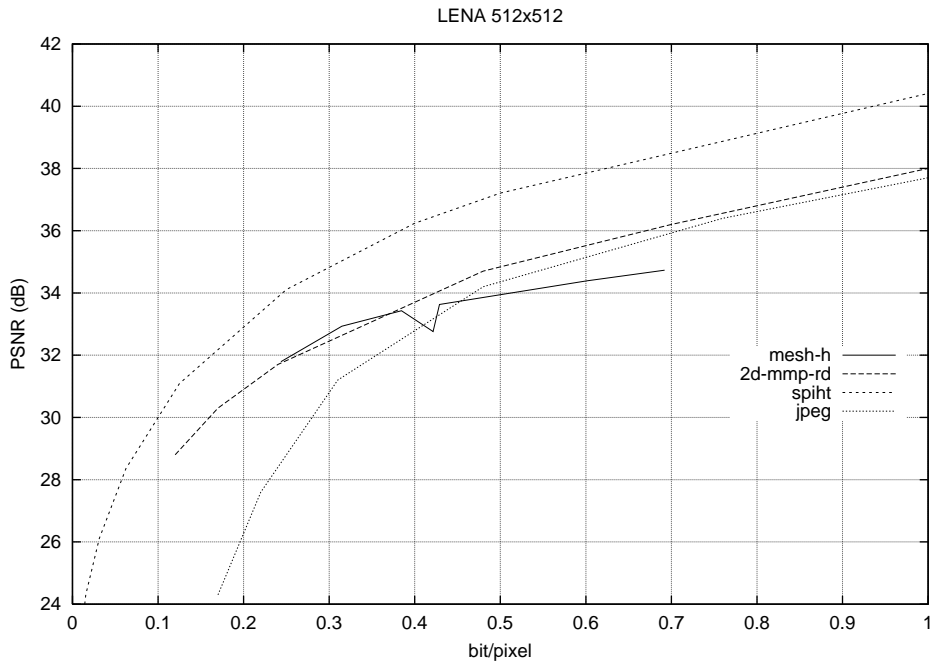


Figura 4.6: Resultados em taxa \times distorção da codificação da imagem Lena 512×512 com malha hierárquica e com o codificador SPIHT.

linha de tabela 4.1) com a Lena 512×512 , onde há uma brusca descontinuidade quando se passa de triângulos 4×2 para 2×1 , indicando que para aquela taxa de bits ainda não se deve descer na hierarquia ao nível da malha mínima.

Na imagem Mandrill, o uso desse nível mais baixo da malha faz o tamanho do código disparar, não só na parte do código usada para os nós e a árvore, mas também nos arcos diagonais, e a qualidade melhora muito pouco. Outra característica dessa imagem é a pequena melhora na qualidade com o refinamento das amplitudes dos nós (aumento do número de passos do quantizador). É importante notar também que na codificação do mandrill no teste num. 5, muito pouco mudou em relação ao teste num. 4, com o dobro do valor dos limiares.

Na Lena 256×256 com triângulos 2×1 , o código também aumenta muito, mas ainda há um aumento considerável na qualidade. Porém, pelo gráfico de qualidade da imagem em função da taxa de bits, vê-se nitidamente a queda na inclinação da curva, como nas demais imagens.

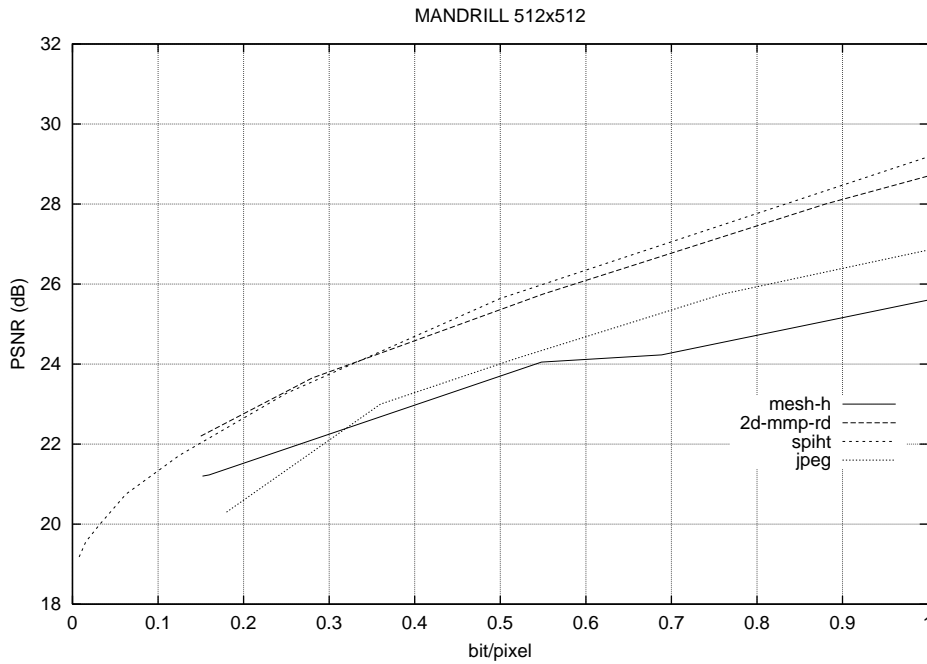


Figura 4.7: Resultados em taxa \times distorção da codificação da imagem Mandrill 512×512 com malha hierárquica e com o codificador SPIHT.

4.6.1 Direções dos arcos na malha 2×1

Na malha 2×1 , todos os pixels da imagem encontram-se no centro de arcos diagonais. Isso significa que cada pixel da imagem é obtido pela interpolação de apenas dois nós. Dessa forma, as direções dos arcos ortogonais são irrelevantes para a qualidade da aproximação de imagens com o uso desta malha. Da forma como está implementado o codificador, todos os arcos ortogonais assumem uma única direção, o que implica em permutas diagonais também apenas em uma direção. O código com a informação das direções dos arcos ortogonais é gerado desnecessariamente. No entanto, como as direções são todas idênticas e o codificador aritmético aproveita esta previsibilidade para reduzir o tamanho do código (podemos observar este efeito analisando o histograma), esse código é praticamente desprezível.

Os valores dos pixels dependem das direções dos arcos diagonais, e eles só podem permutar todos para uma mesma direção, ou mais horizontal ou mais vertical. Além da restrição quanto ao posicionamento dos arcos diagonais, com todos os arcos ortogonais alinhados numa mesma direção, a princípio, todos os arcos diagonais podem ser permutados (apesar de apenas numa direção), o que aumenta o código dessa informação. Naturalmente, arcos diagonais permutados impedem seus dois

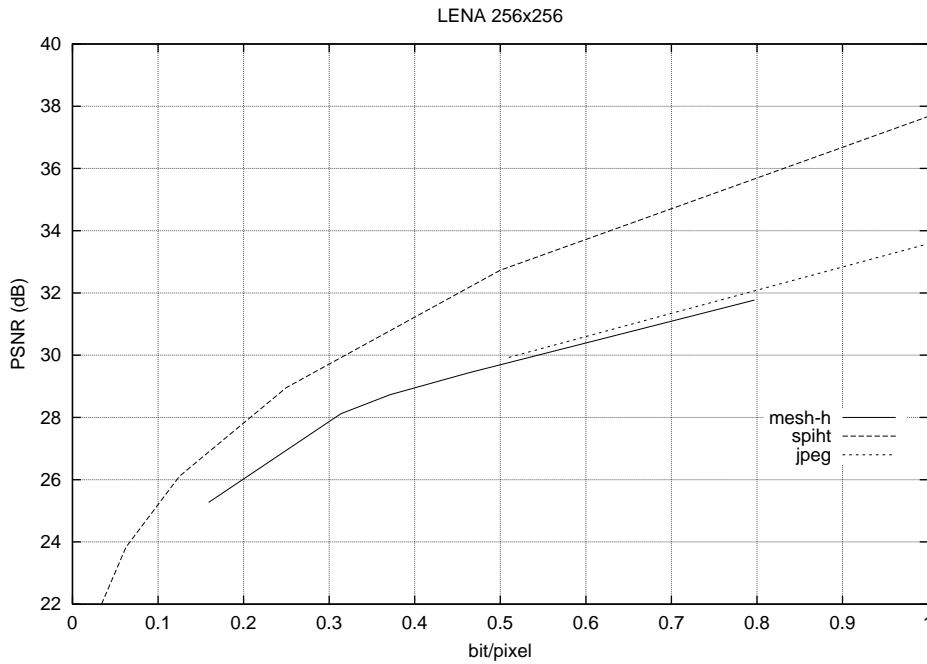


Figura 4.8: Resultados em taxa \times distorção da codificação da imagem Lena 256×256 com malha hierárquica e com o codificador SPIHT.

vizinhos de se permutarem.

A solução para este problema poderia ser implementar um novo algoritmo para permuta dos arcos diagonais, mais complexo que o atual, que permitisse que eles assumissem qualquer das duas direções, independente das direções de todos os arcos ortogonais à sua volta.

4.7 Conclusões

Os resultados da codificação de imagens com o codificador de malha hierárquica são muito dependentes dos parâmetros escolhidos para o codificador. Por mais que testemos, não é suficiente que obtenhamos combinações ótimas desses parâmetros para determinadas imagens, produzindo certos valores de qualidade da imagem e tamanho do código. Portanto, é necessário que se use um método de obter automaticamente os parâmetros o mais próximos possíveis dos ótimos, que produzam imagens com qualidade próxima da esperada, assim como tamanho desejado do código.

Entretanto, pelos resultados obtidos nos testes até aqui, não nos parece que

o codificador se aproxime muito mais de codificadores com transformada wavelet, como o SPIHT, usado nas comparações. Além disso, a representação/codificação da malha mínima parece estar inculindo uma limitação grande no uso do codificador em taxas de bits mais altas (qualidades de imagens melhores). No caso dos testes aqui realizados, esses limites estão pouco acima de 0.3 bpp na duas imagens Lena e próximo a 0.55 bpp na imagem Mandrill.

Há determinadas combinações de parâmetros que produzem resultados de taxa-distorção um pouco melhores do que os apresentados neste capítulo, mas não há como definir uma regra única para todas as imagens. No capítulo 5 descrevemos a otimização usando o critério taxa-distorção, que utiliza um único parâmetro de taxa-distorção para decidir o tamanho da malha mínima, as simplificações de triângulos, as permutas de arcos e o número de passos do quantizador das amplitudes dos nós.

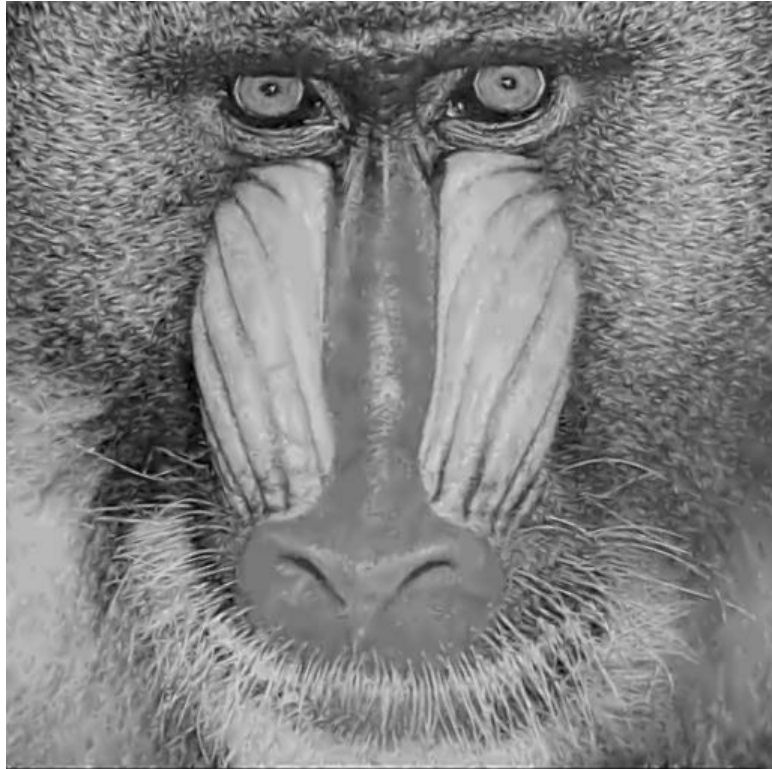


(a)

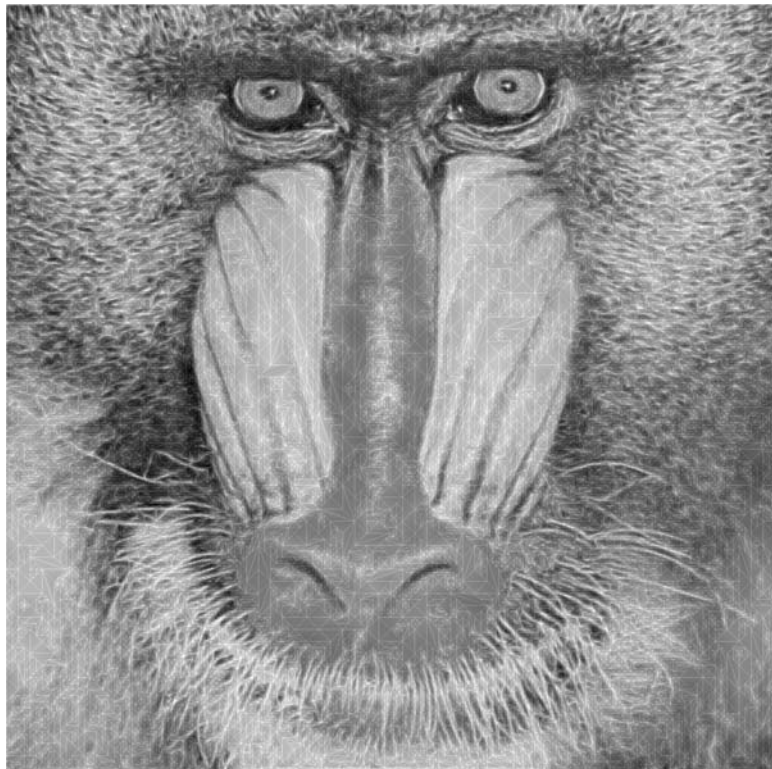


(b)

Figura 4.9: (a) Imagem Lena 512×512 codificada a 0.4290 bpp com malha hierárquica (PSNR = 33.63 dB). (b) A mesma imagem superposta pela malha.



(a)



(b)

Figura 4.10: (a) Imagem Mandrill 512×512 codificada a 0.5482 bpp com malha hierárquica (PSNR = 24.05 dB). (b) A mesma imagem superposta pela malha.



(a)



(b)

Figura 4.11: (a) Imagem Lena 256×256 codificada a 0.3710 bpp com malha hierárquica ($\text{PSNR} = 28.73 \text{ dB}$). (b) A mesma imagem superposta pela malha.

Capítulo 5

Codificação com critério taxa-distorção

O processo de codificação da malha que aproxima a imagem depende da tomada de várias decisões. Elas dizem respeito ao tamanho dos triângulos da malha mínima, ao número de passos de quantização das amplitudes dos nós e aos limiares de erro para a permuta de arcos e simplificação da malha. Até o capítulo 4, fizemos essas escolhas empiricamente, baseadas em testes preliminares com as imagens a serem codificadas. No entanto, para que o codificador possa funcionar automaticamente, é necessário que ele tome essas decisões de forma autônoma, baseado apenas em um único parâmetro, por exemplo, de qualidade ou tamanho do código gerado (taxa de bits).

Utilizaremos um parâmetro de taxa-distorção, ou seja, que especifica a relação custo/benefício associada aos bits do código produzido pela codificação de uma imagem em relação à qualidade da imagem que pode ser reconstruída a partir desse código.

5.1 Critério taxa-distorção

A utilização de um critério taxa-distorção global para a imagem resolve o problema da multiplicidade de parâmetros e, adicionalmente, procura otimizar a alocação dos bits do código de forma que eles representem o que reduzirá mais substancialmente a distorção da imagem codificada e, posteriormente, reconstruída,

em relação à imagem original, para uma dada taxa de bits. Estimamos que o impacto desta mudança seria pequeno em relação à complexidade da sua implementação.

A teoria taxa-distorção trata da tarefa de representar uma fonte ou um conjunto de dados de entrada, como um sinal, por exemplo, com o menor número possível de bits para uma dada qualidade de reprodução [21]. Este assunto vem sendo estudado há mais de meio século, desde o trabalho de SHANNON [22] [23], que foi responsável pela criação de um novo campo de estudo denominado “teoria da informação.”

5.1.1 Restrição pelo tamanho do código

Um caso típico de otimização com taxa-distorção é onde há restrição no número total de bits gastos. Este número disponível de bits deve ser distribuído para as unidades de código, minimizando a distorção global [21]. O problema é como encontrar o quantizador ótimo para cada unidade de código i tal que:

$$\sum_{i=1}^N r_i \leq R_T,$$

(onde r_i é o número de bits gasto para o código i e R_T , o número total de bits gastos) e uma medida de distorção $f(d_1, d_2, \dots, d_N)$ seja minimizada.

Esta medida poderia ser, por exemplo, a distorção média ou a distorção máxima permitida.

5.1.2 Dependência entre elementos da malha

Na representação por malha existe uma dependência muito grande entre os elementos que a caracterizam, quais sejam, as amplitudes de nós, os tamanhos de triângulos e as direções de arcos. Para uma otimização levando-se em conta esta dependência, seria necessário o uso de programação dinâmica [21], na qual todas as combinações possíveis de resultados são analisadas. Esta solução, no entanto, raramente é adotada porque consome um tempo de processamento muito grande.

A codificação aritmética adaptativa adotada para todos os elementos da malha também gera uma dependência grande entre os itens codificados em seqüência. Apesar disso, de forma a simplificar a otimização, consideraremos cada elemento da

malha a ser codificado como sendo independente dos demais. Em muitos casos, esta simplificação não piora de forma expressiva os resultados da otimização [21].

5.1.3 Otimização Lagrangeana

Para resolver o problema de alocação de bits descrito em 5.1.1, utilizaremos a otimização Lagrangeana. Esta técnica consiste na introdução de um multiplicador Lagrangeano $\lambda \geq 0$, um número real não negativo. O custo Lagrangeano $J(\lambda)$ para uma distorção D obtida pela codificação a uma taxa de bits R é dado por [21]:

$$J(\lambda) = D + \lambda R \quad (5.1)$$

Precisamos especificar uma medida para a taxa na qual o sinal é codificado e outra para a distorção do sinal representado nesse código em relação ao original. Usaremos a taxa em bits/pixel e a distorção em erro médio quadrático.

Na comparação de dois custos para escolher que decisão tomar, o multiplicador Lagrangeano determina uma razão:

$$\lambda = -\Delta D / \Delta R ,$$

que dá a relação entre a redução na distorção e o aumento na taxa de bits ou, vice-versa, a relação entre o aumento na distorção e a redução na taxa de bits.

5.2 Codificação da malha

No processo de representação da imagem pela malha hierárquica apresentada no capítulo 4, escolhemos empiricamente o tamanho dos triângulos na malha inicial mais fina. Com o codificador com taxa-distorção, iniciamos sempre com a malha mais fina possível, ou seja, com triângulos de 2×1 pixels.

Após o procedimento de simplificação ou agrupamento de triângulos que produz a malha hierárquica, fazemos uma avaliação a partir do nível mais baixo, com triângulos de 2×1 pixels, do custo em termos de taxa-distorção relativo à codificação do nível. Caso este custo da manutenção do último nível esteja acima do limite associado ao parâmetro λ escolhido, todo o nível é eliminado da malha.

A seguir, descrevemos como fazemos o cálculo das taxas de bits necessárias para a codificação de cada elemento da malha e dos correspondentes custos em termos de taxa-distorção.

5.2.1 Custo dos arcos ortogonais no nível mais baixo

No capítulo 4, as permutas de arcos ortogonais eram decididas baseadas apenas nas variações de distorção provocadas por elas. Com o critério taxa-distorção, levamos em conta também a taxa necessária para codificar cada um dos dois estados possíveis de cada arco. Para isso, simulamos a codificação aritmética da direção de cada arco ortogonal e medimos o número de bits gastos para cada opção de código. Calculamos, através da expressão 5.1, o custo para cada caso, usando o parâmetro de otimização constante λ escolhido, a distorção e taxa de bits obtidas também para cada caso, e escolhemos a direção que gerar o menor custo.

5.2.2 Custo dos arcos diagonais

A estimação da taxa de bits gasta na codificação dos arcos diagonais é mais complicada porque o processo de permuta dos arcos diagonais é composto de três fases. Inicialmente, avaliamos a variação na distorção provocada pela permuta de cada arco; em seguida, ordenamos esses arcos por ordem decrescente de redução da distorção; ao final, executamos as permutas propriamente ditas e as codificamos. Entretanto, como uma permuta diagonal impede que seus dois arcos vizinhos se permutem, nem todos os arcos cujas permutas reduziriam a distorção poderão ser permutados.

Como a simulação da codificação aritmética dos estados dos arcos tem que ser feita durante a primeira fase deste processo, a de avaliação das distorções, e nem todas as permutas ocorrerão efetivamente, os valores das taxas de bits não serão idênticos aos que serão obtidos na codificação real, mas apenas aproximações deles.

As reais taxas de bits da codificação dos arcos diagonais poderão ser obtidas se, ao identificarmos um arco como candidato a permuta por ter atendido ao critério de custo, analisarmos se, por acaso, o arco diagonal anterior, seu vizinho à esquerda ou acima, dependendo do tipo do arco, respectivamente, permutado para mais horizontal ou para mais vertical, também é candidato a permuta. Em caso

afirmativo, os custos deles deverão ser comparados e apenas o de menor custo deverá permanecer na lista de candidatos a permuta. Além disso, o arco que não mais integrar a lista de candidatos deverá ter sua simulação de codificação desfeita, de modo a corrigir o histograma do codificador aritmético. O cálculo dessas taxas reais não foi implementado neste trabalho.

5.2.3 Custo das amplitudes dos nós

A quantização e codificação por aproximações sucessivas das amplitudes dos nós é realizada em uma seqüência *raster-scan* de varredura dos nós da malha, sem nenhuma ordem de prioridade escolhida, por planos de bits, dos mais significativos para os menos significativos. Dessa forma, o critério taxa-distorção somente pode ser usado na definição do número de passos ou planos de bits do quantizador, por meio do cálculo do custo global da codificação de cada plano de bits e da variação na distorção associada.

Se quiséssemos seguir uma seqüência que escolhesse os nós pelo seu custo em termos de taxa-distorção, teríamos que informar a posição de cada nó codificado, produzindo um aumento significativo na taxa de bits. Uma solução intermediária poderia ser a de se utilizar alguma informação embutida na estrutura da malha que pudesse estar associada a maiores ou menores custos de codificação dos nós.

Como exemplo da utilização da estrutura da malha para determinar a ordem de codificação dos nós, em [2] os nós são codificados por ordem de nível na hierarquia da malha, a partir do nível mais alto. A cada nível subsequente, os novos nós são estimados a partir dos nós do nível imediatamente acima, já codificados, e as diferenças são quantizadas e codificadas. Esse procedimento faz parte de um esquema de codificação progressiva da malha.

5.2.4 Custo da árvore da malha

A hierarquia da malha triangular usada neste trabalho constitui-se de uma árvore com cada nó representando um triângulo da malha e possuindo 4 filhos ou nenhum, dependendo, respectivamente, se o seu triângulo correspondente é dividido em outros 4 ou não. Quando montamos esta árvore baseados nas variações de distorção, determinamos simultaneamente as direções dos arcos ortogonais em todos

os níveis, com excessão do nível mais baixo, pois esses arcos já foram escolhidos na otimização da malha mínima inicial.

Como descrito no capítulo 4, montamos a árvore de baixo para cima, a partir da malha mais fina. Analisamos cada grupo de 4 triângulos cuja união poderia formar um triângulo do nível seguinte acima, quanto às distorções no caso de permanecerem separados e de se juntarem, formando um triângulo maior.

Para calcularmos as taxas de bits correspondentes, temos que levar em conta, para cada grupo de 4 triângulos e nas duas possibilidades, as entidades que teriam que ser codificadas. Na descrição a seguir, consideraremos o caso de um triângulo maior do tipo superior, ou seja, horizontal e com o vértice oposto à hipotenusa acima dela. As mesmas considerações são aplicáveis aos três demais casos de triângulos maiores, inferior, esquerdo e direito. A figura 5.1 mostra os nós e arcos nas duas situações e, a seguir, listamos os itens a serem codificados em cada caso:

1. Codificação dos 4 triângulos separados:

- 6 nós
- 3 arcos ortogonais
- 6 arcos diagonais
- estado dos 4 triângulos, divididos ou não (com excessão do nível mais baixo, onde nenhum é dividido)

2. Codificação do triângulo maior obtido pelo agrupamento:

- 3 nós
- 1 arco ortogonal
- 2 arcos diagonais

5.2.4.1 Variações nas taxas e distorções

Para simplificar o cálculo das taxas de bits e dos custos, calcularemos a variação nas distorções entre os dois casos, e a diferença entre as taxas dos elementos que não ocorrem simultaneamente nos dois casos. As taxas dos três nós dos vértices do triângulo maior, por exemplo, não são calculadas porque eles ocorrem nos dois

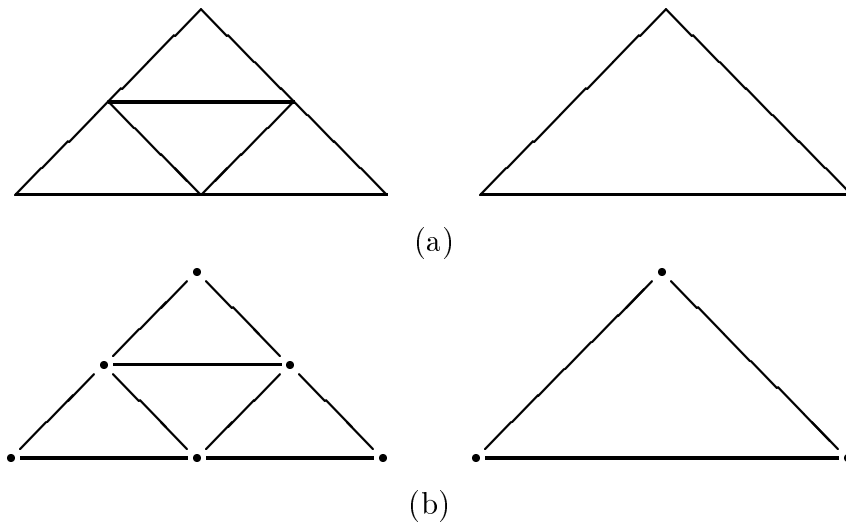


Figura 5.1: (a) Simplificação da malha com a união de um grupo de 4 triângulos; e (b) identificação dos nós e arcos das duas malhas.

casos e as taxas devem variar muito pouco entre as duas situações. Esta pequena variação é consequência do codificador aritmético ser adaptativo.

No cálculo das taxas em cada situação, serão considerados apenas os nós e arcos numerados na figura 5.2. No caso de divisão ou não agrupamento (figura 5.2(a)), eles são 3 nós, 3 arcos ortogonais e 6 arcos diagonais, além de 4 símbolos, exceto no nível mais baixo, indicando para cada um dos 4 triângulos menores, se ele será ou não dividido. Caso ocorra o agrupamento de triângulos (figura 5.2(b)), 1 arco ortogonal e 2 arcos diagonais, do nível imediatamente acima na árvore, devem ter suas taxas de bits calculadas.

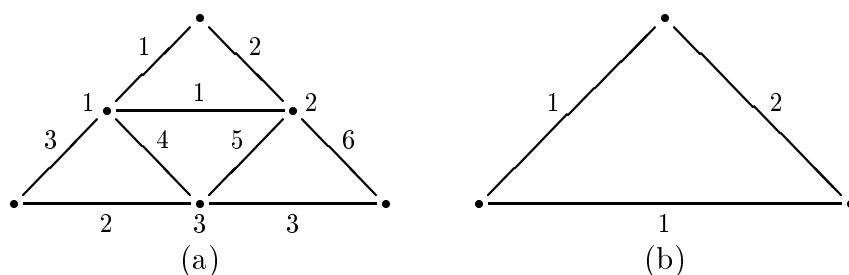


Figura 5.2: Identificação dos nós e arcos candidatos a serem utilizados no cálculo das taxas de bits; (a) na malha do grupo de 4 triângulos; e (b) na malha do triângulo maior.

5.2.4.2 Casos de triângulos vizinhos já simplificados

Devemos analisar também a existência de eventuais triângulos vizinhos já simplificados. Como procedemos aos agrupamentos ou simplificações de triângulos, nível a nível, transversalmente, na direção *raster-scan*, a princípio, somente o triângulo acima e à esquerda do triângulo analisado, poderia estar simplificado. No entanto, esse processo de simplificação é repetido iterativamente e, então, devemos considerar todas as possibilidades de vizinhos já simplificados.

Por exemplo, se o único vizinho simplificado for o triângulo acima e à esquerda, teremos a situação apresentada na figura 5.3(a). No caso de divisão ou não agrupamento, deverão ser calculadas as taxas de bits de 2 nós, 3 arcos ortogonais e 4 arcos diagonais, além de 4 símbolos, exceto no nível mais baixo, indicando, para cada um dos 4 triângulos menores, se ele será ou não dividido. Caso ocorra o agrupamento de triângulos, 1 arco ortogonal e dois arcos diagonais, do nível imediatamente acima na árvore, devem ter suas taxas de bits calculadas.

Com os 2 vizinhos simplificados, acima, um à esquerda e o outro à direita, figura 5.3(b), teremos, para o caso de triângulos divididos, 1 nó, 3 arcos ortogonais, 2 arcos diagonais, e quando não for o nível mais baixo, os 4 símbolos dos triângulos-filhos.

Finalmente, com os 2 vizinhos simplificados, um acima e à esquerda e outro abaixo, figura 5.3(c), teremos, para o caso de triângulos divididos, 1 nó, 1 arco ortogonal, 4 arcos diagonais, e, quando em nível acima do mais baixo, os 4 símbolos dos triângulos-filhos.

Se todos os três vizinhos já tiverem sido simplificados, então o triângulo maior em questão já existe, ou seja, já está totalmente contido no plano que passa pelos seus três vértices, não havendo necessidade da codificação dos arcos menores no seu interior. Neste caso, o custo sempre cai com a simplificação, pois a distorção não varia e a taxa de bits é reduzida.

5.2.4.3 Eliminação de níveis mais baixos da malha

Para decidir cada agrupamento de 4 triângulos, avaliamos a diferença nos custos dos dois casos, mantendo os 4 triângulos ou agrupando-os, optando pelo de menor custo. No entanto, na codificação de determinadas imagens com determinadas

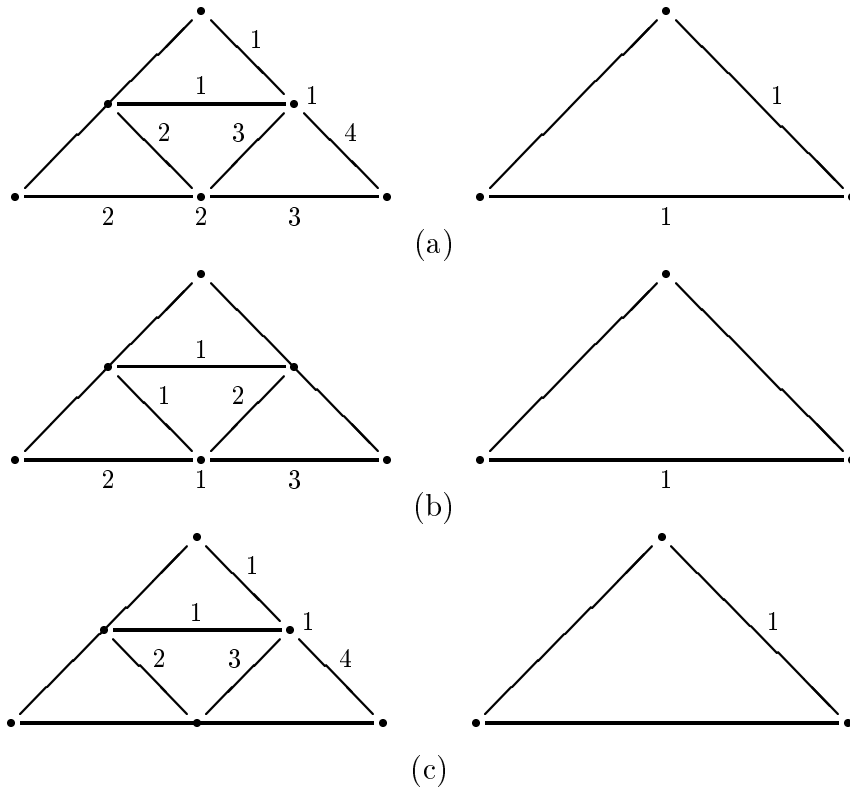


Figura 5.3: Simplificação de triângulo com (a) 1 triângulo acima e à esquerda já simplificado; (b) 2 triângulos acima, à esquerda e à direita, simplificados; e (c) 2 triângulos simplificados, um acima e à esquerda e outro abaixo.

taxas de bits, haverá situações onde o número de triângulos no último nível da malha será pequeno e a eliminação de todo o nível da malha melhorará sensivelmente a taxa-distorção global da imagem codificada. Isto ocorre porque, ao eliminarmos este último nível, eliminamos também um número elevado de códigos que informam a divisão ou não de triângulos do penúltimo nível.

Então, após montada toda a malha hierárquica, fazemos uma avaliação a partir do nível mais baixo, com triângulos de 2×1 pixels, da taxa-distorção relativa à codificação do nível, ou seja, se o custo da manutenção do último nível está acima do limite $J(\lambda)$ associado ao parâmetro λ escolhido. Neste caso, eliminamos este último nível da malha e repetimos o teste para o nível imediatamente acima, e assim por diante.

A eliminação do nível mais baixo da malha implica na necessidade de se repetir o processo de simplificação daquele nível, que desta vez ocorrerá independente dos custos, apenas para a determinação das direções dos novos arcos ortogonais

daquele que agora será o último nível da malha. Em seguida, os novos arcos diagonais do nível também são analisados para se determinar os que serão permutados.

No algoritmo, a simplificação e a permuta dos arcos diagonais do novo último nível são repetidas por completo, em seqüência, para que se obtenha os valores corretos das taxas de bits necessárias à codificação de cada item.

5.3 Resultados experimentais

A seguir são apresentados resultados da codificação de algumas imagens de teste com o uso da malha hierárquica e otimização com critério taxa-distorção. Os resultados da codificação com os algoritmos SPIHT [13], MMP [20] e JPEG são também apresentados para efeito de comparação.

As tabelas 5.1, 5.2 e 5.3 mostram os resultados da codificação, respectivamente, das imagens Lena 512×512 , Mandrill 512×512 e Lena 256×256 . Nelas relacionamos os valores de λ usados e os tamanhos dos triângulos menores da malha e números de passos de quantização por eles determinados. As quantidades de bytes gastos na codificação da árvore hierárquica (que inclui os arcos ortogonais), dos arcos diagonais e das amplitudes dos nós também são mostradas separadamente. Ao final, vêm os totais de bytes de código gastos, a taxa em bits por pixel e a relação sinal/ruído da imagem codificada.

Em todas as imagens testadas, o codificador apresenta resultados melhores nas taxas muito baixas de bits. Na maioria das imagens, com excessão para a PP1205 e PP1209, os resultados com o codificador proposto acompanham de perto os do codificador MMP, que também não usa transformação, até 0.4 ou 0.5 bit por pixel, dependendo da imagem. A partir deste ponto, o aumento na relação sinal/ruído (PSNR) em relação ao aumento na taxa de bits se reduz consideravelmente. Neste ponto, a malha mínima de 4×2 passa a não representar bem as imagens. É necessário o uso da malha de 2×1 , mas esta malha, com já dissemos, não proporciona uma representação muito eficiente em termos de taxa-distorção, e a inclinação da curva taxa \times PSNR continua menor.

As figuras de 5.4 a 5.10 mostram os gráficos com os resultados da relação sinal/ruído em função da taxa de bits, respectivamente, para as imagens Lena

Tabela 5.1: Resultados da codificação da imagem Lena 512×512 com malha hierárquica e critério taxa-distorção para diferentes valores de λ .

λ	Tam. Tri.	Pass. Qntz.	Árvore (bytes)	Diag. (bytes)	Nós (bytes)	Código (bytes)	Taxa (bpp)	PSNR (dB)
900	8	4	670	287	1055	2012	0.0614	26.55
600	8	4	737	350	1265	2352	0.0718	27.33
200	8	5	841	476	2087	3404	0.1039	28.46
100	8	5	910	562	2411	3883	0.1185	28.74
50	4	5	2060	1041	4324	7425	0.2266	31.56
30	4	5	2462	1300	5378	9140	0.2789	32.37
13	4	5	3048	1748	7193	11989	0.3659	33.32
11	4	6	3136	1908	9650	14694	0.4484	33.72
9	2	5	5121	2100	11155	18376	0.5608	34.02
4	2	5	6715	3825	15954	26494	0.8085	35.76

Tabela 5.2: Resultados da codificação da imagem Mandrill 512×512 com malha hierárquica e critério taxa-distorção para diferentes valores de λ .

λ	Tam. Tri.	Pass. Qntz.	Árvore (bytes)	Diag. (bytes)	Nós (bytes)	Código (bytes)	Taxa (bpp)	PSNR (dB)
2000	8	3	593	185	698	1476	0.0450	19.78
700	8	3	1034	568	1473	3075	0.0938	20.69
400	4	3	2824	1744	4604	9172	0.2799	22.43
180	4	4	3342	2623	1473	14499	0.4425	23.59
45	2	4	8034	5652	1473	39102	1.1933	26.38

512×512 , Mandrill 512×512 , Lena 256×256 , Aerial 512×512 , PP1205 512×512 e PP1209 512×512 , com o codificador proposto (identificado pelo nome *mesh-rd*), o 2d-mmp-rd (apresentado em [20]¹), o spiht [13] e o JPEG.

As imagens obtidas pela codificação das mesmas imagens de teste com malha hierárquica e taxa-distorção são apresentadas nas figuras de 5.11 a 5.17, juntamente com os valores de λ usados, os números de passos de quantização e os tamanhos de triângulos menores obtidos a partir de λ e usados na codificação, bem como as taxas (em bits por pixel) e distorções (PSNR em dB) das imagens codificadas.

¹com excessão da imagem Lena 256×256

Tabela 5.3: Resultados da codificação da imagem Lena 256×256 com malha hierárquica e critério taxa-distorção para diferentes valores de λ .

λ	Tam. Tri.	Pass. Qntz.	Árvore (bytes)	Diag. (bytes)	Nós (bytes)	Código (bytes)	Taxa (bpp)	PSNR (dB)
1000	8	3	190	85	251	526	0.0642	22.58
900	8	4	231	110	455	796	0.0972	24.25
100	4	4	673	375	1225	2273	0.2775	27.55
55	4	5	794	465	1907	3166	0.3865	28.75
11	2	4	1785	1147	3710	6642	0.8108	31.86

5.4 Conclusões

Como comentado na seção 4.7, o codificador apresenta bons resultados apenas em baixas taxas de bits. Em taxas de bits mais altas, no entanto, a eficiência do codificador vai baixando gradativamente, notadamente quando é necessário o uso da malha mais fina usada no algoritmo em muitas regiões da imagem. Esta malha mais fina possui um triângulo para cada pixel da imagem e um nó para cada dois pixels, aumentando assim a quantidade de itens na descrição da malha. Apesar disto, esta malha mais fina não consegue aproximar com fidelidade muito alta regiões de alta frequência na imagem, inculindo uma limitação na qualidade máxima possível nesta representação.

Outro motivo para a degradação do codificador em taxas mais altas é a necessidade de maior número de passos na quantização das amplitudes dos nós. Uma quantização diferencial dos nós pode aumentar a eficiência do quantizador. Esta quantização se inicia pelos nós da malha do nível mais alto. As amplitudes dos nós da malha do nível imediatamente abaixo são obtidas pela média das amplitudes dos nós do extremo do arco que passa por ele, mais o valor da diferença quantizada.

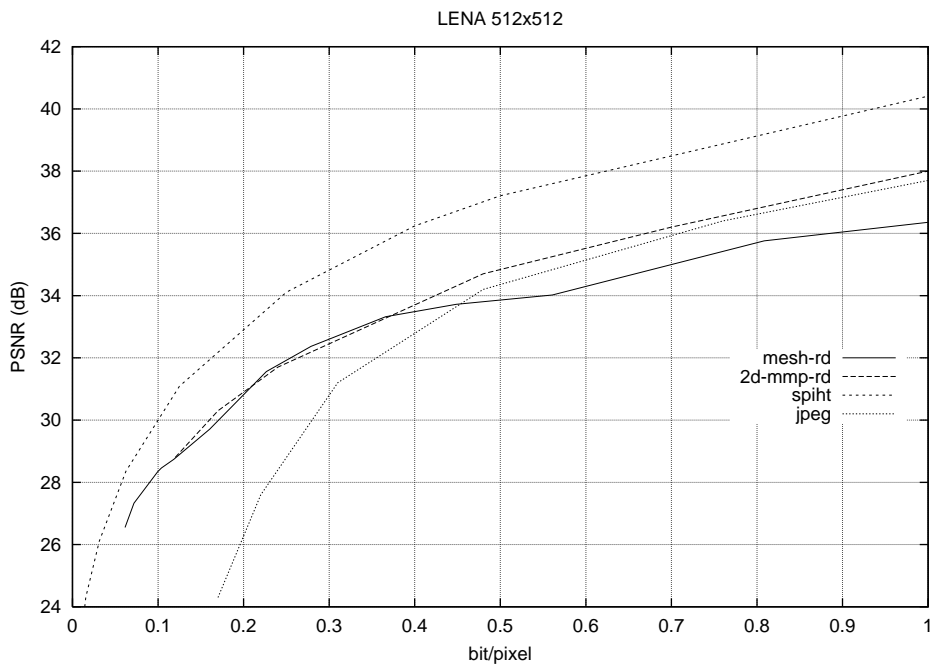


Figura 5.4: Resultados da codificação da Lena 512×512.

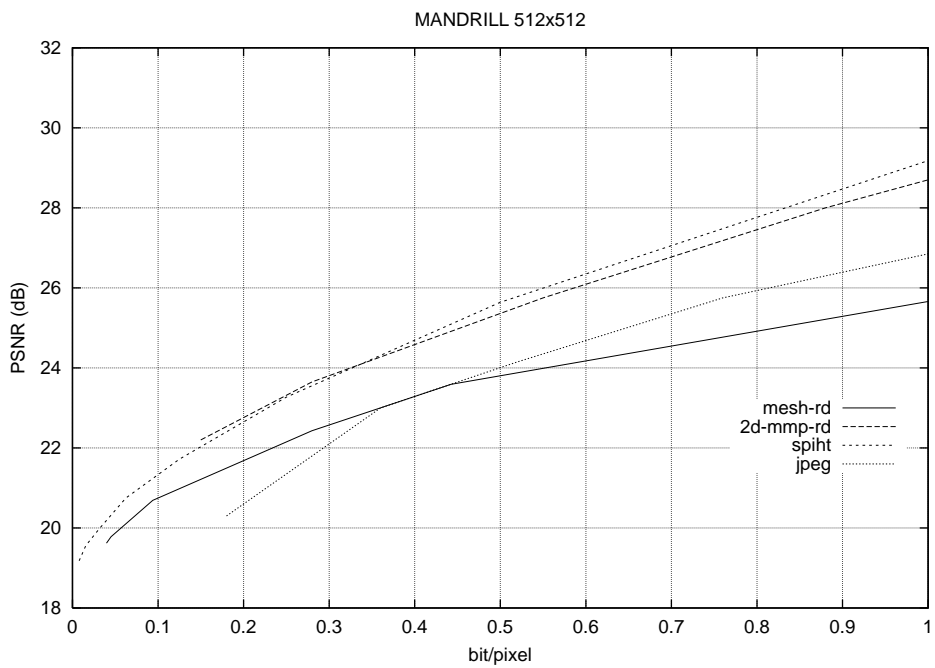


Figura 5.5: Resultados da codificação do Mandrill 512×512.

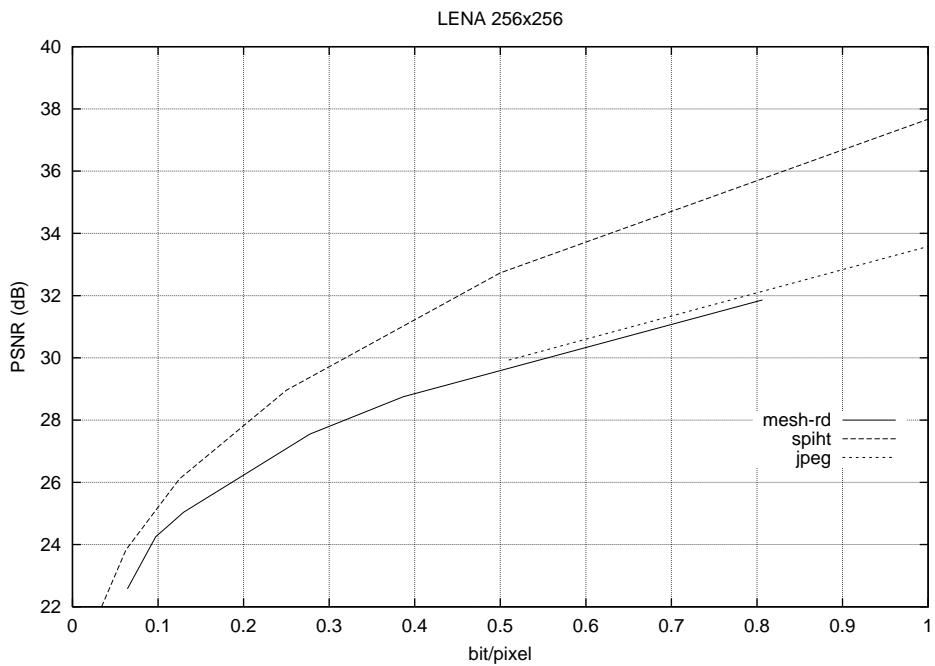


Figura 5.6: Resultados da codificação da Lena 256×256.

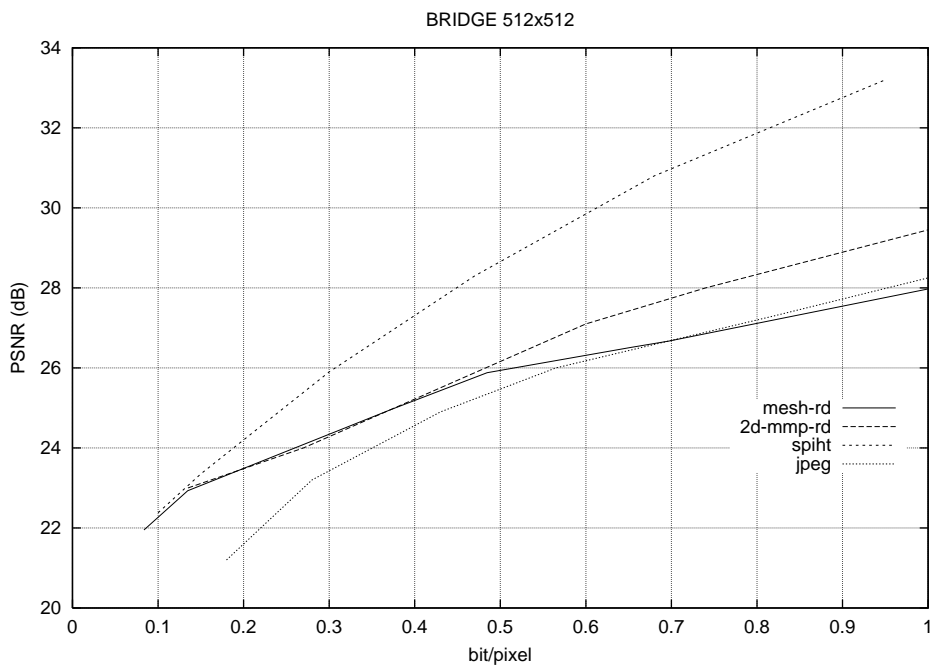


Figura 5.7: Resultados da codificação da Bridge 512×512.

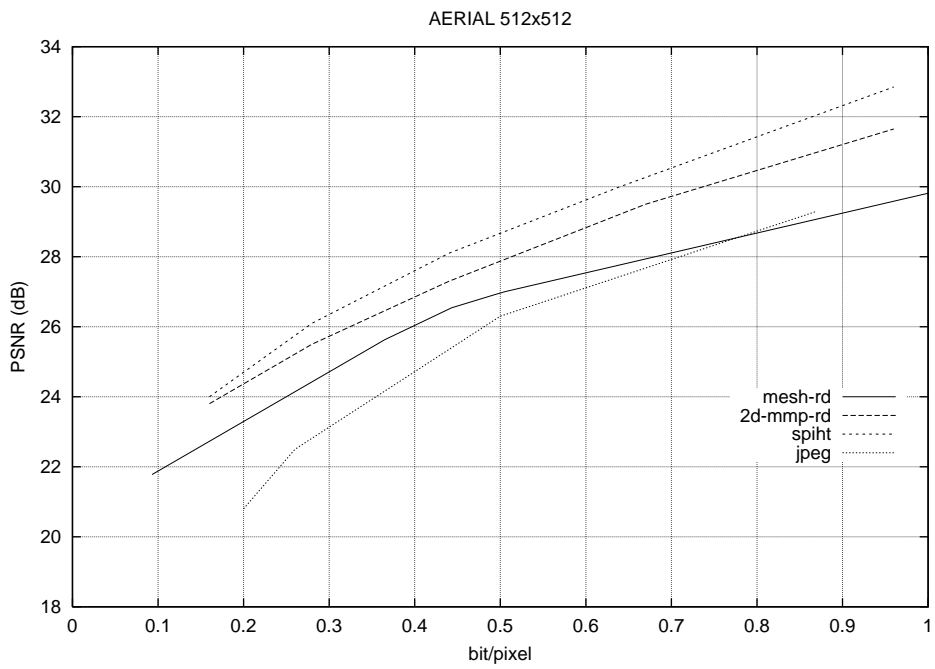


Figura 5.8: Resultados da codificação da Aerial 512×512.

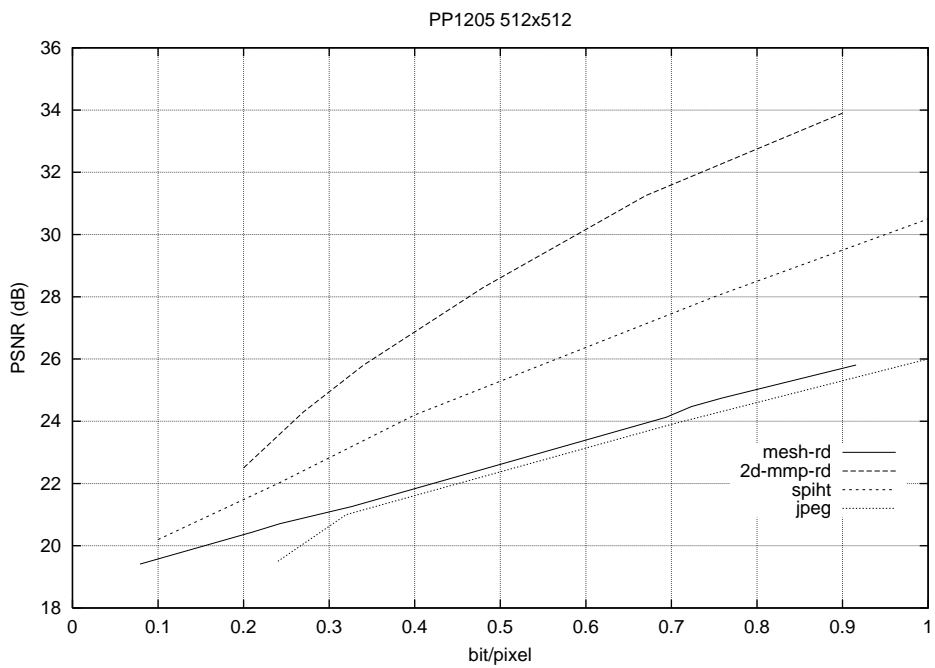


Figura 5.9: Resultados da codificação da PP1205 512×512.

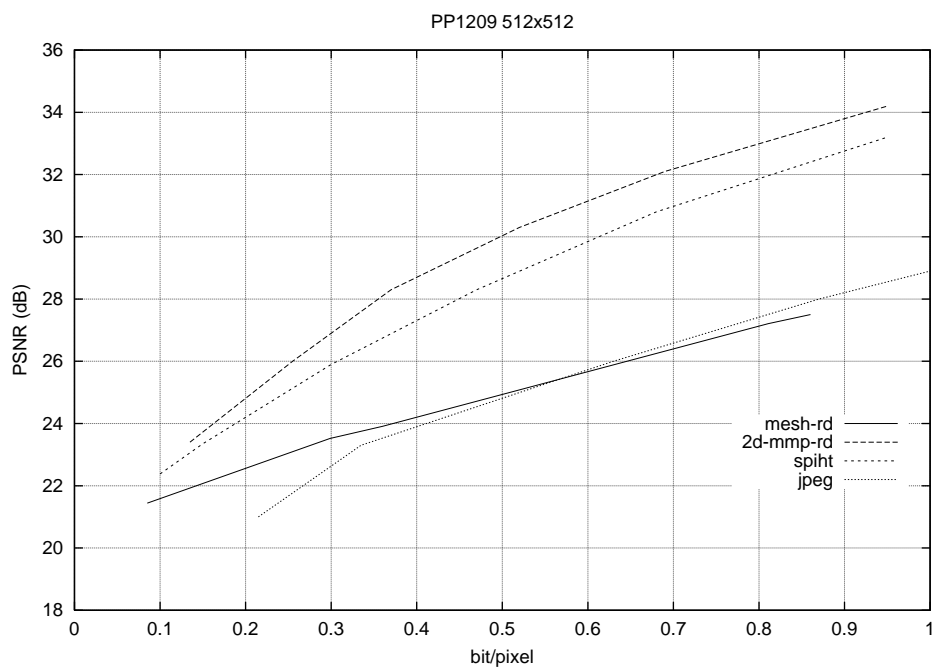


Figura 5.10: Resultados da codificação da PP1209 512×512.



Figura 5.11: Imagem Lena 512×512 codificada com taxa-distorção, $\lambda = 13$, que determinou 5 passos de quantização dos nós, triângulos menores de 4×2 pixels, taxa de 0.366 bpp e PSNR de 33.32 dB.

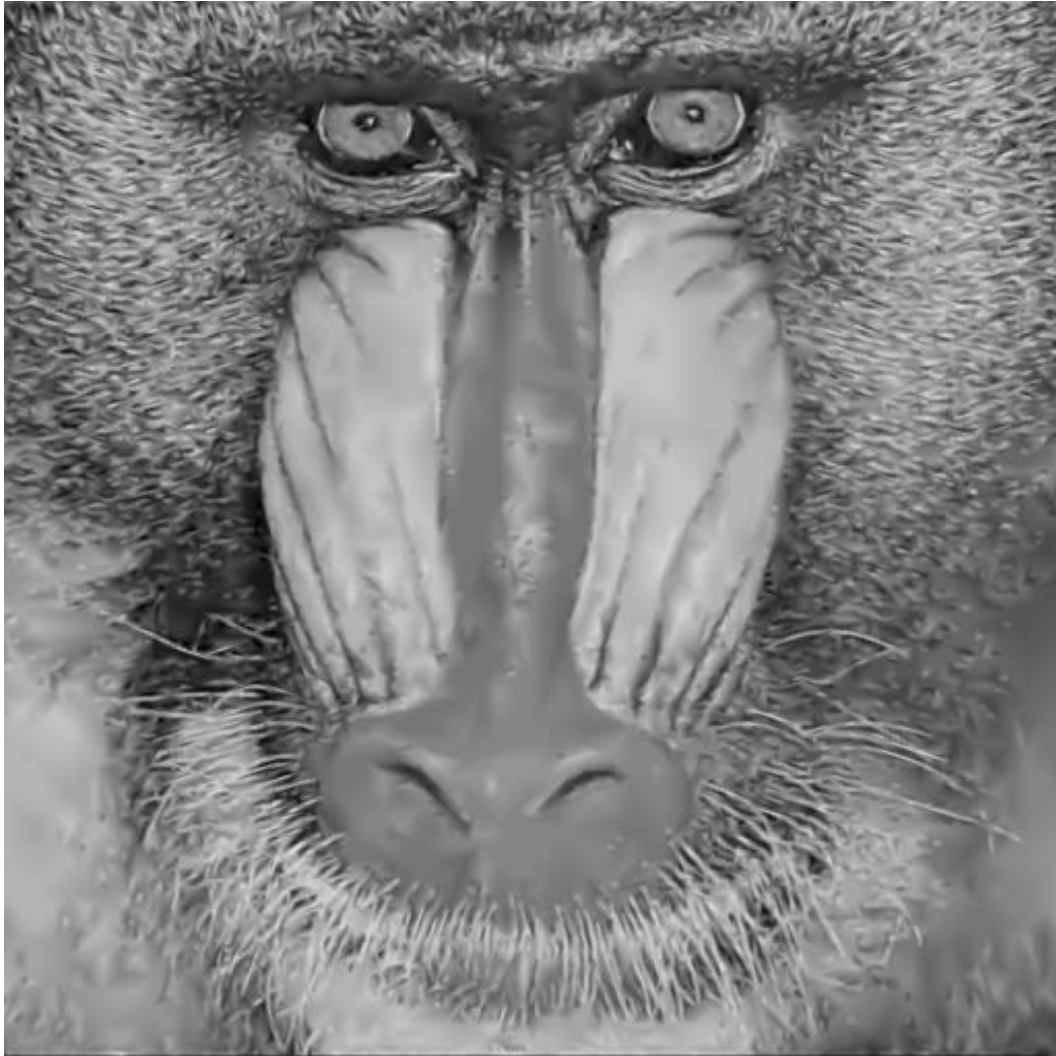


Figura 5.12: Imagem Mandrill 512×512 codificada com taxa-distorção, $\lambda = 180$, que determinou 4 passos de quantização dos nós, triângulos menores de 4×2 pixels, taxa de 0.442 bpp e PSNR de 23.59 dB.



Figura 5.13: Imagem Lena 256×256 codificada com taxa-distorção, $\lambda = 11$, que determinou 4 passos de quantização dos nós, triângulos menores de 2×1 pixels, taxa de 0.811 bpp e PSNR de 31.86 dB.



Figura 5.14: Imagem Bridge 512×512 codificada com taxa-distorção, $\lambda = 100$, que determinou 4 passos de quantização dos nós, triângulos menores de 4×2 pixels, taxa de 0.485 bpp e PSNR de 25.88 dB.



Figura 5.15: Imagem Aerial 512×512 codificada com taxa-distorção, $\lambda = 100$, que determinou 4 passos de quantização dos nós, triângulos menores de 4×2 pixels, taxa de 0.443 bpp e PSNR de 26.54 dB.

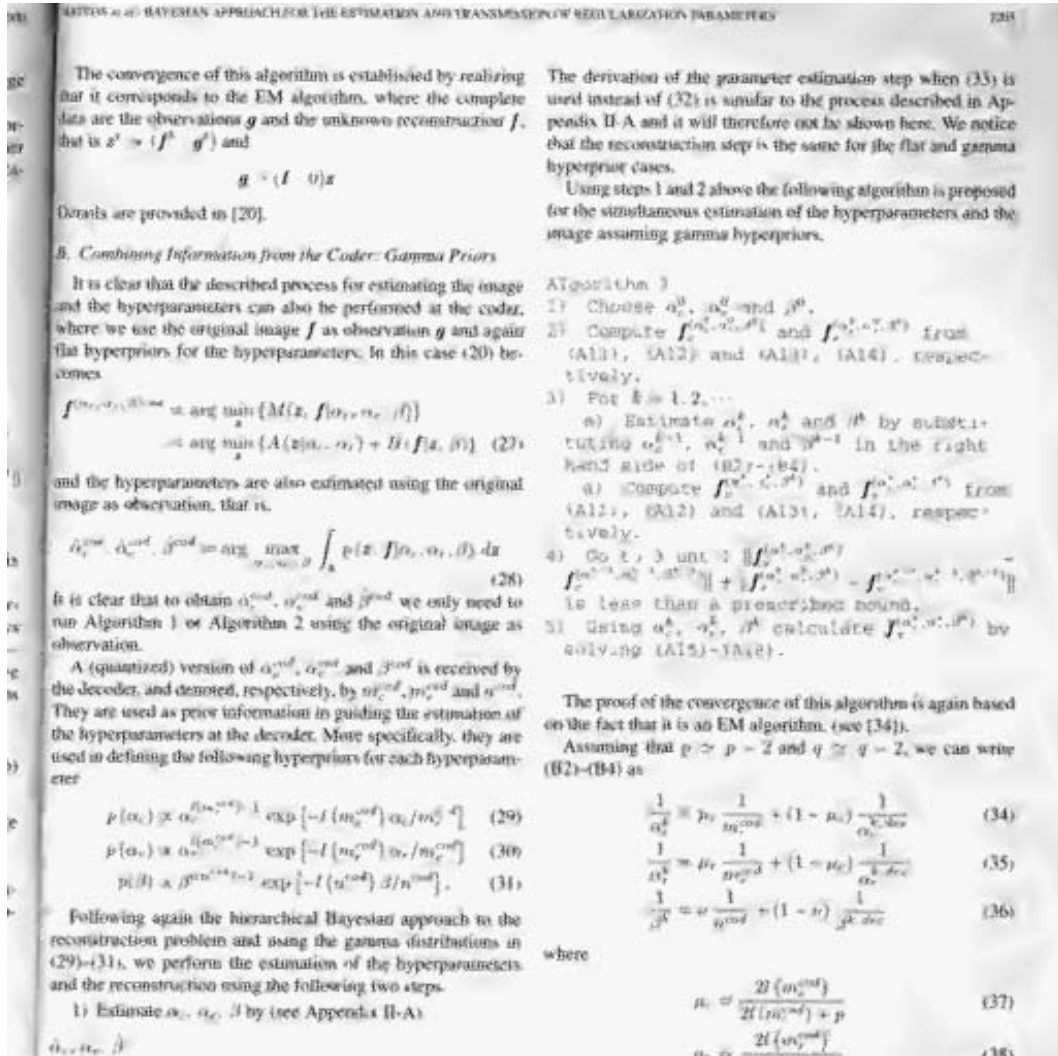


Figura 5.16: Imagem PP1205 512×512 codificada com taxa-distorção, $\lambda = 50$, que determinou 4 passos de quantização dos nós, triângulos menores de 2×1 pixels, taxa de 0.760 bpp e PSNR de 24.75 dB.

TABLE II
PSNR OBTAINED BY ESTIMATING THE PARAMETERS AT THE CODER

Image	bpp	Alg. 1 at the coder	Alg. 2 at the coder
airplane	0.32	30.92	30.94
airplane	0.55	34.35	34.26
Lena	0.29	31.37	31.38
Lena	0.54	34.76	34.77
peppers	0.32	30.60	30.63
peppers	0.53	32.48	32.51

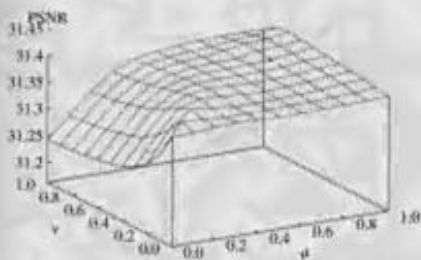


Fig. 6. PSNR for different values of μ and ν as the *Lena* image compressed at 0.29 bpp.

original image as observation, and then using these parameters in (20) to obtain the reconstruction. The results are shown in Table II. It can be seen that the PSNR improves slightly in this case.

The parameters obtained at the coder and the decoder were then combined. The same normalized confidence parameters μ_c and μ_d , defined in (37) and (38), were used for α_c and α_d . The values used in the experiments were $\mu_c = \mu_d = \mu \in \{0.0, 0.1, \dots, 1\}$. The normalized confidence parameter ν , defined in (39), belongs to the same range. The 3-D plot in Fig. 6 shows the PSNR as a function of μ and ν for the *Lena* highly compressed image. The center part of the compressed image and the best reconstruction, corresponding to the parameter values $m_c^{cod} = \alpha_c^{cod} = 30.82^{-1}$, $m_d^{cod} = \alpha_d^{cod} = 5.36^{-1}$ and $\nu^{cod} = \beta^{cod} = 36.36^{-1}$ with $\mu = 0.9$ and $\nu = 0.0$ is displayed in Fig. 7(b). The corresponding PSNR is 31.40 dB. Similar results are obtained using other high compressed images showing that best reconstructions in terms of PSNR are obtained using μ



(a)



(b)

Figura 5.17: Imagem PP1209 512×512 codificada com taxa-distorção, $\lambda = 50$, que determinou 4 passos de quantização dos nós, triângulos menores de 2×1 pixels, taxa de 0.811 bpp e PSNR de 27.21 dB.

Capítulo 6

Conclusões

Nesta tese, apresentamos um algoritmo para obter representações de imagens naturais por meio de malhas triangulares hierárquicas, e esquemas eficientes para quantização e codificação dessa informação, atendendo a critérios de taxa de compressão versus distorção das imagens.

A imagem é aproximada, inicialmente, por uma malha bem fina, que vai sendo simplificada, região por região, através do agrupamento de triângulos que forma triângulos maiores, onde eles conseguem representar bem aquela região da imagem original. Assim vai sendo criada uma malha hierárquica, onde cada nível possui triângulos de área 4 vezes maior do que os triângulos do nível imediatamente abaixo na hierarquia. Os arcos são também permutados de forma a otimizar esta aproximação.

Numa primeira etapa do trabalho, escolhemos limiares de distorção para as tomadas de decisão quanto ao tamanho dos triângulos da malha mais fina, o número de passos de quantização das amplitudes dos nós, as permutas ou não cada arco e o agrupamento ou não de triângulos. Ao final, introduzimos um critério taxa-distorção para a otimização dessas escolhas, com a utilização de apenas um parâmetro, λ , no qual passam a se basear todas essas decisões.

Os resultados da codificação em taxas muito baixas de bits são muito bons, inclusive quando comparados a codificadores com transformada wavelet. Considerando que a codificação se processa diretamente sobre a imagem, sem nenhuma transformação preliminar, esse codificador pode ter vantagens, por exemplo, na codificação de quadros-diferença em seqüências de vídeo com estimação e compensação

de movimento. Os codificadores com transformadas não têm um bom desempenho neste tipo de imagens.

Em taxas de bits mais altas, no entanto, a eficiência do codificador vai baixando, principalmente quando a representação utiliza a malha mais fina do algoritmo. A malha mais fina não consegue também aproximar bem regiões de alta frequência da imagem, quando se deseja distorções muito baixas. Nesse caso, o maior número necessário de passos de quantização das amplitudes dos nós também piora o resultado em termos de taxa-distorção.

Para reduzir este problema, poderá ser tentado, nos níveis mais baixos da malha hierárquica, o uso da divisão por biseção [12], onde os triângulos são divididos em 2 e não em 4 menores de um nível para o seguinte na malha. No entanto, essa mudança implica no aumento do código da árvore hierárquica por aumentar o seu número de níveis, mas dessa forma poderíamos chegar ao nível dos pixels da imagem.

Uma quantização diferencial dos nós também pode ser testada para aumentar a eficiência do quantizador utilizado, como comentado na seção 5.4. Esquemas mais complexos de predição, como os usados no codificador CALIC ou técnicas de dicionários [24], por exemplo, podem permitir aumento na compactação dos dados tanto das amplitudes dos nós como dos estados dos arcos, em relação aos utilizados neste trabalho.

Outra possibilidade prevista como extensão ao codificador proposto é a de se representar toda a imagem independentemente por malhas de cada um dos níveis possíveis. A partir dessas N representações otimizadas da imagem para cada malha, haveria um processo de eleição por critério taxa-distorção do nível da malha para cada região da imagem e de obtenção das amplitudes ótimas dos nós.

A substituição da interpolação linear de pixels dentro dos triângulos por outros tipos de interpolação pode suavizar as bordas dos triângulos e, em alguns casos, melhorar a a qualidade visual e a relação sinal/ruído das imagem restauradas. Por exemplo, uma interpolação que utiliza os vetores normais aos vértices ou nós dos triângulos que compõem a malha [25].

Bibliografia

- [1] LECHAT, P., SANSON, H., LABELLE, L., “Image Approximation by Minimization of a Geometric Distance Applied to a 3D Finite Elements Based Model”. In: *IEEE International Conference on Image Processing*, Washington, DC, Sept. 1997.
- [2] LECHAT, P., LAURENT, N., SANSON, H., “Scalable Image Coding with Fine Granularity Based on Hierarchical Mesh”. In: *Proceedings of Visual Communications and Image Processing*, San Jose, CA, Jan. 1999.
- [3] LAURENT, N., ROBERT, G., “Mesh Based Video Coding at Very Low Bistrate”. In: *IEEE International Conference on Image Processing*, Vancouver, Canada, Oct. 2000.
- [4] DEMARET, L., ROBERT, G., “Scalable Image Coder Mixing DCT and Triangular Meshes”. In: *IEEE International Conference on Image Processing*, Vancouver, Canada, Oct. 2000.
- [5] LAURENT, N., LECHAT, P., SANSON, H., “Limitation of Triangles Overlapping in Mesh-Based Motion Estimation Using Augmented Lagrangian”. In: *IEEE International Conference on Image Processing*, Chicago, Illinois, Sept. 1998.
- [6] JANSEN, M., CHOI, H., LAVU, S., *et al.*, “Multiscale Image Processing using Normal Triangulated Meshes”. In: *IEEE International Conference on Image Processing*, Thessaloniki, Greece, Sept. 2001.
- [7] JANSEN, M., BARANIUK, R., LAVU, S., “Multiscale Approximation of Piecewise Smooth Two-Dimensional Functions using Normal Triangulated Meshes”, *submitted*, , Feb. 2003.

- [8] LU, T., LE, Z., YUN, D. Y. Y., “Piecewise Linear Image Coding using Surface Triangulation and Geometric Compression”. In: *IEEE Data Compression Conference*, Snowbird, Utah, March 2000.
- [9] HOPPE, H., DEROSE, T., DUCHAMP, T., *et al.*, “Mesh Optimization”. In: *Computer Graphics (SIGGRAPH Proceedings)*, pp. 19–26, Anaheim, CA, Aug. 1993.
- [10] HOPPE, H., “Progressive Meshes”. In: *Computer Graphics (SIGGRAPH Proceedings)*, pp. 99–108, New Orleans, Louisiana, Aug. 1996.
- [11] ECK, M., DEROSE, T., DUCHAMP, T., *et al.*, “Multiresolution Analysis of Arbitrary Meshes”. In: *Computer Graphics (SIGGRAPH Proceedings)*, pp. 173–182, Los Angeles, CA, Aug. 1995.
- [12] VELHO, L., ZORIN, D., “4-8 Subdivision”, *Computer Aided Geometric Design*, v. 18, n. 5, pp. 397–427, June 2001.
- [13] SAID, A., PEARLMAN, W. A., “A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees”, *IEEE Transactions on Circuits and Systems for Video Technology*, v. 6, pp. 243–250, June 1996.
- [14] VIEIRA, A. W., VELHO, L., LOPES, H., *et al.*, “Fast Stellar Mesh Simplification”. In: *Proceedings of the XVI Brazilian Symposium on Computer Graphics and Image Processing, IEEE Press*, pp. 27–34, 2003.
- [15] VIEIRA, A. W., LOPES, H., “Mesh Simplification and Compression Made Simple Using the Corner-Table”. In: *Proceedings of the second edition of the Brazilian Workshop on Games and Digital Entertainment*, 2003.
- [16] VIEIRA, A. W., *A topological framework for mesh simplification*. Ph.D. dissertation, Pontifícia Universidade Católica do Rio de Janeiro, 2003.
- [17] BARRETO, C. S., MENDONÇA, G. V., da SILVA, E. A. B., “Image Coding with Coarse Mesh Approximation”. In: *IEEE South American Workshop on Circuits and Systems*, Rio de Janeiro, Brazil, Nov. 2001.

- [18] BARRETO, C. S., MENDONÇA, G. V., da SILVA, E. A. B., “Image Coding at Very Low Bitrates with Regular Triangular Mesh”. In: *IEEE International Telecommunication Symposium*, Natal, Brazil, Sept. 2002.
- [19] SHAPIRO, J. M., “Embedded Image Coding Using Zerotrees of Wavelets Coefficients”, *IEEE Transactions on Signal Processing*, v. 41, n. 12, pp. 3445–3462, Dec. 1993.
- [20] de CARVALHO, M. B., *Compressão de Sinais Multi-Dimensionais usando Padrões de Recorrência de Escalas*. Ph.D. dissertation, Universidade Federal do Rio de Janeiro, March 2001.
- [21] ORTEGA, A., RAMCHANDRAN, K., “Rate-Distortion Methods for Image and Video Compression”, *IEEE Signal Processing Magazine*, pp. 23–50, Nov. 1998.
- [22] SHANNON, C. E., “A Mathematical Theory of Communication”, *The Bell System Technical Journal*, v. 27, n. 3, 4, pp. 379–423, 623–656, July, Oct. 1948.
- [23] SHANNON, C. E., “Coding Theorems for a Discrete Source with a Fidelity Criterion”. In: *IRE National Convention Record, Part. 4*, pp. 142–163, 1959.
- [24] SAYOOD, K., *Introduction to Data Compression*. 2 ed. Morgan Kaufmann Pub., 2000.
- [25] WATT, A., WATT, M., *Advanced Animation and Rendering Techniques*. Addison-Wesley Pub., 1992.