

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA

COMPRESSÃO DE DOCUMENTOS COMPOSTOS
UTILIZANDO O H.264/AVC-INTRA

Alexandre Zaghetto

Brasília, Maio de 2009

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

COMPRESSÃO DE DOCUMENTOS COMPOSTOS
UTILIZANDO O H.264/AVC-INTRA

Alexandre Zaghetto

ORIENTADOR: Ricardo Lopes de Queiroz

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA

Publicação: PPGENE.TD 039/09

Brasília/DF: Maio - 2009

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA
COMPRESSÃO DE DOCUMENTOS COMPOSTOS
UTILIZANDO O H.264/AVC-INTRA

Alexandre Zaghetto

*Tese de doutorado submetida ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Doutor em Engenharia Elétrica*

Banca Examinadora

Ricardo Lopes de Queiroz, PhD.

UnB/ ENE
Orientador

Francisco Assis de O. Nascimento, Dr.

UnB/ ENE
Examinador interno

Juliana Fernandes Camapum, Ph.D.

UnB/ ENE
Examinadora interna

Pedro de Azevedo Berger, Dr.

UnB/ CIC
Examinador externo

Eduardo Antônio Barros da Silva, Ph.D.

COPPE/UFRJ
Examinador externo

FICHA CATALOGRÁFICA

ZAGHETTO, A.

Compressão de Documentos Compostos
utilizando o H.264/AVC-INTRA. [Distrito Federal] 2009.

xxviii, 177p., 297 mm (ENE/FT/UnB, Doutor, Telecomunicações

Processamento de Sinais, 2009). Tese de Doutorado.

Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Elétrica.

1. Processamento de imagens

2. Documentos compostos

3. Compressão de documentos

4. *Mixed Raster Content*

5. H.264/AVC

6. Compressão de livros

I. ENE/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

ZAGHETTO, A. (2009). Compressão de documentos compostos utilizando o H.264/AVC-INTRA. Tese de Doutorado em Engenharia Elétrica com ênfase em Telecomunicações, Publicação PP-GENE.TD 039/09, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 177p.

CESSÃO DE DIREITOS

NOME DO AUTOR: Alexandre Zaghetto.

TÍTULO DA TESE DE DOUTORADO: Compressão de documentos compostos utilizando o H.264/AVC-INTRA.

GRAU / ANO: Doutor / 2009

É concedida à Universidade de Brasília permissão para reproduzir cópias desta tese de doutorado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta tese de doutorado pode ser reproduzida sem a autorização por escrito do autor.

Alexandre Zaghetto

SQN 408 Bloco N Apto. 103

70856-140 Asa Norte, Brasília - DF - Brasil.

Dedicatória

*À minha esposa, Sônia,
e aos filhos, Tarcisio, Leomar, Vinicius e Nadja.*

Agradecimentos

Aos meus avós, **Maria Aparacida Fabre Beltramini**, **Mario Beltramini**, **José Rebelo Furtado**, **Alda Constantina Zagheto** e **Maciste Zagheto**, pelos momentos de carinho a mim dedicados. Em especial, a meu avô **Maciste Zagheto**, que contribuiu para desenvolver em mim o gosto pela investigação, ao permitir que eu pudesse passar minhas férias escolares ajudando-o em sua oficina.

Ao meu pai, **Sidney Zagheto**, e a minha mãe, **Elvira Aparecida Beltramini Zagheto**, por terem cumprido tão bem a função de primeiros mestres. Foram responsáveis por cultivar em mim valores imponderáveis, que transcendem à simples técnica de lidar mecanicamente com os objetos do mundo manifesto.

Ao meu irmão **Cauê Zagheto**, que habita em meu coração como Rama no coração de Lakshman. E à minha cunhada **Daniele Arrais**, por sua “leveza”.

À minha irmã **Letícia Zagheto**, por materializar um ideal de superação, demonstrando como é possível pegar “suas asas quebradas e aprender a voar”. À minha pequena sobrinha **Isadora Satya**, a mais nova integrante de nossa família, pelas boas risadas que é capaz de provocar com seu comportamento espontâneo.

À minha esposa **Sônia Regina Ferreira Zagheto**, pelo amor, pela tolerância e pelo suporte sem o qual eu não poderia ter seguido o caminho que escolhi para minha vida.

Aos filhos, **Tarcisio**, **Leomar**, **Vinícius** e **Nadja**, por enriquecerem minha experiência como ser humano.

Aos amigos **Pedro da Fonseca Vieira**, **Eustaquio Grilo**, **André Henrique de Siqueira**, **Karla Patrícia da Silva de Siqueira**, **Jaques Furlanetto de Camargo** e **Agda Macedo de Camargo**, pelos momentos de reflexão construtiva. Aos parceiros **Gabriel Max de Siqueira**, **Hannah Gabrielle de Siqueira** e **Hadrian Lucas de Siqueira**, que não hesitam em pronunciar nosso grito de guerra “MAZEL TOV!”, trazendo alegria constante à nossa convivência. Aos amigos **Enio de Souza Couto** e **Daniele Ramos de Oliveira**, pelo carinho e por mostrarem como se vive de forma séria, porém desconstruída.

Um agradecimento especial ao meu Professor Orientador, **Ricardo Lopes de Queiroz**, que, ao me aceitar como aluno, abriu-me um universo de novas possibilidades.

À Professora **Juliana Fernandes Camapum** e ao Professor **Lúcio Martins da Silva**, por terem participado do meu trabalho na condição de examinadores. Aos professores **Francisco Assis de Oliveira Nascimento**, **Pedro de Azevedo Berger** e **Eduardo Antônio Barros da Silva**, por, além de atuarem como examinadores, terem sido a palavra que, na hora exata, reorientou-me a jornada.

Ao Professor **Adson Ferreira da Rocha**, pela confiança e apoio constantes.

Ao Professor **Luiz Pereira Calôba**, pelo impulso inicial, cujos efeitos se manifestam ainda nos dias de hoje.

A todos os professores do **Departamento de Engenharia Elétrica** da UnB que, de uma maneira ou de outra, acabaram por contribuir com minha formação acadêmica.

Aos colegas de trabalho mais próximos, aqui apresentados em ordem lexicográfica, **Alberto López Delis**, **Bruno Luigi Macchiavello Espinoza**, **Camilo Chang Dorea**, **Diogo Caetano Garcia**, **Edson Mintsu Hung**, **Fabiano Araujo Soares**, **Jorge Andrés Cormane Angarita**, **Karen França de Oliveira**, **Marcelo Villegas**, **Marcus Vinícius Chaffim Costa**, **Maria do Carmo dos Reis**, pelo apoio e pela tolerância com relação às minhas “alucinações”. Em especial, ao colega **Tiago Alves da Fonseca**, por ter realizado a revisão do texto da tese.

Ao Professor **Hilton Martins Mendes**, pelos momentos de filosofia e prática no Yoga.

À **Faculdade Jesus Maria José (FAJESU)** em seu conjunto, por ter sido o ambiente que possibilitou meu amadurecimento profissional. Em especial, ao coordenador do curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Professor **João Batista Simão**, que durante cinco anos cuidou de mim como um pai, e ao coordenador do curso de Tecnologia em Redes de Computadores, Professor **Ronald Emerson Scherolt da Costa**, pelo exemplo de atuação incansável.

Aos “caroneiros” **Jhone Caldeira Silva**, **Olimpio Ribeiro Gomes**, **Jorge Luis de Araújo Coelho** e **Sandra Baccarin**, por terem transformado nossos carros em salas móveis de debates e depoimentos diários e, por isso, terem tornado nossas “viagens” mais agradáveis.

Aos brasileiros, que fazem ser possível a realidade da Universidade Pública e, conseqüentemente, minha realidade profissional.

Ao sempre presente **Francesco**, il “Poverello” d’Assisi, por ter demonstrado ser possível uma vida de humildade. Que suas lições possam, algum dia, nortear minhas ações ainda tão impregnadas de egoísmo.

Aos gigantes cujos ombros me serviram de apoio para que, apesar de minha cegueira, eu pudesse enxergar um pouco mais.

E, finalmente, Àquele a quem devo minha existência.

RESUMO

Dadas a perspectiva de se armazenar a produção intelectual humana em meio digital e a necessidade de se realizar tal tarefa de forma econômica, a presente tese reúne o que há de mais avançado em codificação de imagens, com o objetivo de propor novos métodos que viabilizam a compressão eficiente de documentos digitais. Verificou-se que o H.264/AVC, o mais novo padrão para compressão de vídeo, aparece como uma opção no que se refere à compressão de imagens estáticas, superando, em muitos casos, o JPEG2000, um padrão considerado o estado-da-arte em compressão de imagens em tons contínuos. Para documentos compostos, o desempenho do H.264/AVC é ainda mais expressivo. Sendo assim, as contribuições da presente tese distribuem-se em quatro métodos que utilizam o referido padrão como elemento funcional básico, a saber: método 1, *Advanced Video Coding - Composto*, que agrega ao H.264/AVC a possibilidade de se codificar adaptativamente regiões de texto e imagem a partir da análise do conteúdo de macroblocos; método 2, *Compressão MRC baseada em H.264/AVC-I e JBIG2 para Documentos Gerados Eletronicamente*, que integra o H.264/AVC e o JBIG2 ao modelo de imageamento MRC (*Mixed Raster Content*) e propõe uma técnica de *data-filling* baseada nos modos de predição intra do H.264/AVC; método 3, *Compressão MRC baseada em H.264/AVC-I e JBIG2 para Documentos Digitalizados*, que oferece técnicas de pré/pós-processamento como extensões do modelo MRC; e método 4, *Compressão de Livros Digitalizados utilizando o H.264/AVC*, que procura explorar a recorrência de padrões na compressão das páginas de livros digitalizados. Vários experimentos foram realizados com o objetivo de se verificar a eficiência dos métodos propostos e os resultados mostraram um ganho de desempenho objetivo e/ou subjetivo, com relação a abordagens já conhecidas.

ABSTRACT

Given the perspective of storing human intellectual production using digital media and the need for executing this task in an economic way, the present thesis assembles the most advanced techniques on image coding, in order to propose new methods that enable efficient digital document compression. It has been verified that H.264/AVC, the newest video compression standard, can also be used to encode still images. In many cases, it outperforms state-of-art coders such as JPEG2000. For compound documents, the gains over JPEG2000 are even more expressive. In this scenario, the contributions of the present thesis are distributed over four methods that use the H.264/AVC as a basic functional element, namely: method 1, *Advanced Video Coding - Composto*, which, based on a macroblock content analysis, adaptively encodes text and image regions; method 2, *MRC Compression of Electronically Generated Documents using H.264/AVC-I e JBIG2*, which combines MRC (Mixed Raster Content) with H.264/AVC and JBIG2, and proposes a new data-filling technique based on the H.264/AVC intra prediction; method 3, *MRC Compression of Scanned Documents using H.264/AVC-I e JBIG2*, which offers pre/post-processing techniques as extensions of the MRC imaging model; and method 4, *Compression of Scanned Books using H.264/AVC*, which explores pattern recurrence to encode pages of scanned books. Many experiments were carried out in order to verify the efficiency of the proposed methods. Results showed objective and/or subjective gains over known approaches.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CONTEXTO	1
1.2	APRESENTAÇÃO DO PROBLEMA E JUSTIFICATIVA	1
1.3	DEFINIÇÕES INICIAIS	2
1.4	COMPRESSÃO DE IMAGENS	2
1.5	COMPRESSÃO DE DOCUMENTOS DIGITAIS	3
1.6	MÉTODOS PROPOSTOS	3
1.7	ORGANIZAÇÃO DA TESE	4
2	COMPRESSÃO DE IMAGENS	7
2.1	PROCESSAMENTO DIGITAL DE IMAGENS	7
2.1.1	AQUISIÇÃO DE IMAGENS	7
2.1.2	REPRESENTAÇÃO DIGITAL DE IMAGENS	11
2.2	PADRÕES DE COMPRESSÃO DE IMAGENS	14
2.2.1	PADRÕES DE COMPRESSÃO DE IMAGENS BINÁRIAS	14
2.2.2	PADRÕES DE COMPRESSÃO DE IMAGENS EM TONS CONTÍNUOS	22
3	O MODO INTRA DO PADRÃO H.264/AVC	29
3.1	COMPRESSÃO DE VÍDEO	29
3.2	O PADRÃO H.264/AVC	32
3.3	A PREDIÇÃO INTRA DO H.264/AVC	34
3.3.1	MODOS DE PREDIÇÃO <i>I4</i>	35
3.3.2	MODOS DE PREDIÇÃO <i>I8</i>	40
3.3.3	MODOS DE PREDIÇÃO <i>I16</i>	46
3.4	OTIMIZAÇÃO TAXA-DISTORÇÃO LAGRANGEANA	51
3.5	CODIFICAÇÃO ARITMÉTICA BINÁRIA ADAPTATIVA BASEADA EM CONTEXTO	52
3.6	O H.264/AVC UTILIZADO COMO UM CODIFICADOR DE IMAGENS	53
4	COMPRESSÃO DE DOCUMENTOS DIGITAIS	59
4.1	CONSIDERAÇÕES INICIAIS	59
4.2	COMPRESSÃO DE DOCUMENTOS BASEADA EM CODIFICADOR ÚNICO	60
4.3	COMPRESSÃO DE DOCUMENTOS BASEADA EM MÚLTIPLOS CODIFICADORES: O MODELO DE IMAGEAMENTO <i>Mixed Raster Content</i>	61
4.3.1	SEGMENTAÇÃO EM CAMADAS	65

4.3.2	PREENCHIMENTO DE REGIÕES IRRELEVANTES (<i>Data-filling</i>).....	68
5	MÉTODO 1: ADVANCED VIDEO CODING - COMPOSTO.....	77
5.1	CONSIDERAÇÕES INICIAIS	77
5.2	MOTIVAÇÃO E CONTEXTO	77
5.3	ALOCAÇÃO DE TAXA BASEADA EM SEGMENTAÇÃO.....	78
5.4	ALGORITMO DE SELEÇÃO DE QP E QP_{Texto}	83
5.5	RESULTADOS	85
5.6	CODIFICAÇÃO DE REGIÃO DE INTERESSE	101
6	MÉTODO 2: COMPRESSÃO MRC BASEADA EM H.264/AVC-I E JBIG2 PARA DOCUMENTOS GERADOS ELETRONICAMENTE.....	105
6.1	CONSIDERAÇÕES INICIAIS	105
6.2	<i>Data-filling</i> BASEADO NAS PREDIÇÕES I16 DO AVC-I	105
6.3	CODIFICAÇÃO DOS PLANOS	106
6.4	RESULTADOS	107
7	MÉTODO 3: COMPRESSÃO MRC BASEADA EM H.264/AVC-I E JBIG2 PARA DOCUMENTOS DIGITALIZADOS	123
7.1	CONSIDERAÇÕES INICIAIS	123
7.2	O PROBLEMA COM TRANSIÇÕES SUAVES.....	124
7.3	AGUÇAMENTO E RESSUAVIZAÇÃO DE TRANSIÇÕES	126
7.4	PARAMETRIZAÇÃO DA SUAVIDADE DE TRANSIÇÕES.....	131
7.5	RESULTADOS	133
8	MÉTODO 4: COMPRESSÃO DE LIVROS DIGITALIZADOS UTILIZANDO O H.264/AVC.....	153
8.1	CONSIDERAÇÕES INICIAIS	153
8.2	PRÉ-PROCESSAMENTO PARA CODIFICAÇÃO DE LIVROS DIGITALIZADOS UTILIZANDO O H.264/AVC	154
8.3	RESULTADOS	155
9	CONCLUSÃO	165
9.1	CONSIDERAÇÕES INICIAIS	165
9.2	MÉTODO 1: ADVANCED VIDEO CODING - COMPOSTO.....	165
9.3	MÉTODO 2: COMPRESSÃO MRC BASEADA EM H.264/AVC-I E JBIG2 PARA DOCUMENTOS GERADOS ELETRONICAMENTE	166
9.4	MÉTODO 3: COMPRESSÃO MRC BASEADA EM H.264/AVC-I E JBIG2 PARA DOCUMENTOS DIGITALIZADOS.....	166
9.5	MÉTODO 4: COMPRESSÃO DE LIVROS DIGITALIZADOS UTILIZANDO O H.264/AVC.....	167
9.6	CONSIDERAÇÕES FINAIS	167
	REFERÊNCIAS BIBLIOGRÁFICAS	169

LISTA DE FIGURAS

1.1	Ilustração da distinção feita entre documento digitalizado e documento gerado eletronicamente, ambos considerados documentos digitais.....	2
2.1	Sistema de imageamento: (a) sensor único; (b) sensores dispostos em vetor; e (c) sensores dispostos em matriz.	8
2.2	Convencionam-se situar a origem dos eixos de uma imagem no canto superior esquerdo.	9
2.3	Ilustração de um sistema de imageamento.	9
2.4	Imagem “goldhill”: exemplo de imagem em escala de cinza.	10
2.5	Imagem “Lena”: exemplo de imagem colorida, acompanhada de suas três componentes <i>RGB</i>	10
2.6	Espaços de cores: (a) <i>RGB</i> ; e (b) <i>YC_bC_r</i>	11
2.7	Imagem “Lena” representada no espaço de cores <i>YC_bC_r</i> : (a) luminância <i>Y</i> ; (b) componente de crominância <i>C_b</i> ; e (c) componente de crominância <i>C_r</i>	11
2.8	Imagem “acanoavirou”: exemplo de imagem binária ou “preto e branco”.	12
2.9	Exemplo de imagem com $M = 8$ linhas, $N = 8$ colunas e $K = 256$ níveis de cinza. Cada quadrado representa um <i>pixel</i> . O número no interior dos <i>pixels</i> indica o nível de cinza a que pertencem.	13
2.10	Imagem binária representada como um fluxo de dígitos binários unidimensional, organizado a partir da varredura <i>raster</i>	15
2.11	Ilustração da codificação aritmética.	17
2.12	Ilustração da decodificação aritmética.	18
2.13	As frequências de ocorrência de 0’s e 1’s estão condicionadas a dois contextos, (a) C_1 e (b) C_2 . Nas figuras, X representa o <i>pixel</i> a ser codificado.	20
2.14	Exemplos de tipos de regiões: (a) região de símbolos; (b) região de <i>halftone</i> ; e (c) região genérica.	22
2.15	Diagrama de blocos do JPEG: codificação e decodificação.	23
2.16	Varredura <i>zig-zag</i> : padrão para reordenação dos coeficientes da DCT.	24
2.17	Imagem “baboon” codificada com o JPEG: (a) imagem original (8 <i>bits/pixel</i>); (b) imagem reconstruída com PSNR = 28,2310 dB a 1,3946 <i>bits/pixel</i> ; e (c) imagem reconstruída com PSNR = 23,4256 dB a 0,4563 <i>bits/pixel</i> . As imagens (d), (e) e (f) são ampliações das imagens (a), (b) e (c), respectivamente.	25
2.18	Diagrama de blocos do JPEG2000: codificação e decodificação.	26
2.19	Sub-bandas da DWT.	26
2.20	Decomposição da imagem “Lena” em (a) um e (b) dois níveis, utilizando a CDF 9/7.	27

2.21	Imagem “cameraman” codificada com o JPEG2000: (a) imagem original (8 <i>bits/pixel</i>); (b) imagem reconstruída com PSNR = 36,1248 dB a 0,9857 <i>bits/pixel</i> ; e (c) imagem reconstruída com PSNR = 30,7664 dB a 0,4929 <i>bits/pixel</i> . As imagens (d), (e) e (f) são ampliações das imagens (a), (b) e (c), respectivamente.....	27
2.22	Comparação dos codificadores JPEG e JPEG2000: (a) imagem original (8 <i>bits/pixel</i>); (b) imagem reconstruída com JPEG (PSNR = 28,3434 dB a 0,5076 <i>bits/pixel</i>); e (c) imagem reconstruída com JPEG2000 (PSNR = 32,7589 dB a 0,4997 <i>bits/pixel</i>). As imagens (d), (e) e (f) são ampliações das imagens (a), (b) e (c), respectivamente.	28
2.23	Comparação entre o JPEG e o JPEG2000: (a) imagem “peppers”; e (b) curvas PSNR para a imagem “peppers”.....	28
3.1	Vídeo digital representado como uma matriz tridimensional $V(m, n, \ell)$, composta por M linhas, N colunas e L quadros.	29
3.2	Diagrama de blocos para um CODEC de vídeo.....	30
3.3	Modelo temporal: (a) quadro predito (anterior); (b) quadro atual; e (c) quadro residual.	30
3.4	Estimação de movimento.	31
3.5	Estimação e compensação de movimento: (a) vetores de movimento; (b) quadro predito (por compensação de movimento); e (c) quadro residual. Foram utilizados blocos de 16×16 e janela de busca de 32×32 <i>pixels</i>	31
3.6	Partição com tamanho de bloco variável. Os números ao redor dos blocos representam suas dimensões em <i>pixels</i>	33
3.7	Codificação H.264/AVC.	33
3.8	Decodificação H.264/AVC.	34
3.9	Predição intra para blocos de luminância com tamanho 4×4 <i>pixels</i> . Oito “direções de predição” e modo DC.	36
3.10	Exemplo de predições <i>I4</i> : (a) bloco original a ser predito; (b) a (j) predições geradas a partir dos modos de predição 0 a 8, respectivamente.	36
3.11	Predição intra para blocos de luminância com tamanho 8×8 <i>pixels</i> . Oito “direções de predição” e modo DC.	40
3.12	Exemplo de predições <i>I8</i> : (a) bloco original a ser predito; (b) a (j) predições geradas a partir dos modos de predição 0 a 8, respectivamente.	41
3.13	Predição intra para blocos de luminância com tamanho 16×16 <i>pixels</i>	46
3.14	Exemplo de predições <i>I16</i> : (a) bloco original a ser predito; (b) a (e) predições geradas a partir dos modos de predição 0 a 3, respectivamente.	47
3.15	Exemplos de quadros preditos (a, c, e) e residuais (b, d, f) obtidos a partir das predições: <i>I16</i> (a, b); <i>I8</i> (c, d); e <i>I4</i> (e, f), para a imagem “Lena”.....	50
3.16	Diagrama de blocos do AVC-I: codificação e decodificação.	54
3.17	Curvas PSNR comparando o AVC-I e o JPEG2000 na compressão das imagens: (a, b) “Lena”; (c, d) “goldhill”; e (e, f) “cameraman”.....	55
3.18	Curvas PSNR comparando o AVC-I e o JPEG2000 na compressão das imagens: (a, b) “barbara”; (c, d) “peppers”; e (e, f) “baboon”.....	56

3.19	Curvas PSNR comparando o AVC-I e o JPEG2000 na compressão de documentos compostos: (a, b) “compound1”; (c, d) “compound2”; e (e, f) “scannedcp”.	57
3.20	Curvas PSNR comparando o AVC-I e o JPEG2000 na compressão de documentos compostos: (a, b) “mixed1”; (c, d) “mixed2”; e (e, f) “mixed3”.	58
4.1	Parte ampliada do documento “compound1”. Observe os efeitos de borda ao redor dos caracteres.	60
4.2	Ilustração do modelo de imageamento MRC. O documento original é representado utilizando-se três camadas: <i>Foreground</i> (FG), <i>Background</i> (BG) e máscara (<i>Mask</i>)...	62
4.3	(a) Documento original x ; (b) máscara binária M ; e (c) $M(i, j) \cdot BG(i, j)$; e (d) $(1 - M(i, j)) \cdot FG(i, j)$. Na máscara binária a cor branca é representada pelo valor numérico 1, enquanto o preto é representado por 0. Isso garante que $x(i, j) = M(i, j) \cdot BG(i, j) + (1 - M(i, j)) \cdot FG(i, j)$.	63
4.4	Diagrama de blocos ilustrando os procedimentos de decomposição em camadas, pré-processamento, codificação, decodificação, pós-processamento e fusão das camadas. ...	64
4.5	Cada plano do modelo MRC pode ser processado e comprimido por meio de algoritmos específicos, resultando em um comportamento global de taxa e distorção mais eficiente do que a abordagem baseada em um codificador único.	64
4.6	(a) Macrobloco original; e (b) possíveis limiares: $l = 75$.	66
4.7	(a) Macrobloco original; (b) $\bar{m}_n(i, j)$: máscara invertida relativa ao limiar $t_n = 36$. (c) $\bar{m}_{nd}(i, j)$ máscara deslocada de uma coluna à direita; e (d) valor absoluto da diferença entre $\bar{m}_n(i, j)$ e $\bar{m}_{nd}(i, j)$, ou seja, $\bar{m}_{ndif}(i, j) = \bar{m}_n(i, j) - \bar{m}_{nd}(i, j) $.	67
4.8	Exemplo de segmentação: (a) macrobloco original; (b) custo observado para cada um dos possíveis limiares t_n ; (c), (d) e (e) representam o FG, a máscara e o BG, respectivamente, para $t_n = 36$.	68
4.9	Decomposição em camadas do documento “compound1” antes do procedimento de <i>data-filling</i> : (a) BG; (b) máscara; e (c) FG.	69
4.10	Diagrama que ilustra a decomposição MRC e o procedimento de <i>data-filling</i> .	70
4.11	Exemplo de preenchimento por valor constante igual à média: (a) e (c) BG e FG; (b) e (d) BG_{df} e FG_{df} , versões processadas de BG e FG, respectivamente.	71
4.12	Exemplo de preenchimento iterativo no domínio espacial: em cada etapa, as cores preta e branca indicam os <i>pixels</i> irrelevantes que estão sendo processados na etapa atual e os que serão processados e etapas futuras, respectivamente.	72
4.13	Exemplo de preenchimento iterativo utilizando a DCT: (a) plano inicial I_0 ; (b) iteração I_1 ; (c) iteração I_2 ; e (d) iteração I_3 .	73
4.14	Exemplo de preenchimento por filtragem: (a) $p(i, j)$ (região X preenchida pela média da região U); (b) $p_f(i, j)$ (plano após filtragem por janela retangular 3×3); (c) plano $p_f(i, j)$ após restauração da região U ; (d) linha 9 da imagem (a); e (e) linha 9 da imagem (c). Observe como o processo de filtragem suavizou a transição entre as regiões X e U .	74

4.15	Exemplo de preenchimento utilizando filtragem segmentada. A amostra $X = p_f(6, 7)$ é determinada a partir da filtragem segmentada que utiliza o filtro variante ao deslocamento $h_{6,7}(u, v)$. No exemplo, $p_f(6, 7) = (8 \cdot (67 + 71 + 45 + 44) + 5 \cdot 80)/37 = 60$. Para o melhor entendimento do exemplo, sugere-se a análise atenta das Equações 4.22 e 4.23.	75
4.16	Exemplo de preenchimento por filtragem segmentada: (a) $p(i, j)$ (região X preenchida pela média da região U); (b) plano após filtragem segmentada por janela triangular 3×3 e restauração da região U ; (c) linha 9 da imagem (a); e (d) linha 9 da imagem (b). Observe como o processo de filtragem suavizou a transição entre as regiões X e U	76
5.1	Comportamento gráfico da Equação 5.1. Menores valores de QP implicam menores valores de λ , e vice-versa. Para $0 \leq QP \leq 12$, $\lambda < 1$. Para $13 \leq QP \leq 51$, $\lambda > 1$. Em (a) e (b) λ é dado em escala linear e logarítmica, respectivamente.	78
5.2	Relação entre a taxa R e a distorção D . Considerando a função de custo J , quando $\lambda \rightarrow \infty$, a distorção torna-se irrelevante e R é o único fator que determina a maneira como o macrobloco será codificado. Por outro lado, quando $\lambda \rightarrow 0$, a operação do codificador é determinada exclusivamente a partir do valor da distorção D , sendo a taxa irrelevante.	79
5.3	Segmentação: (a) exemplo de documento composto; (b) resultado da segmentação.	80
5.4	Exemplos das classes de macroblocos: (a) classe 0 (regiões pictóricas), (b) classe 1 (regiões de interior de texto) e (c) classe 2 (regiões de borda de texto).	81
5.5	Máscara de classificação associada à imagem da Figura 5.3 (b). As classe 0 (pictóricas), 1 (interior de texto) e 2 (borda de texto) foram representadas pelos níveis 0 (preto), 128 (cinza) e 255 (branco), respectivamente.	82
5.6	<i>H.264/AVC-INTRA - Composto</i> ou AVC-C: esquema do codificador proposto.	82
5.7	Teste preliminar realizado com o AVC-C: (a) $QP = 0$ (melhor qualidade possível para as classes 0 e 1) e $QP_{\text{Texto}} = 51$ (pior qualidade possível para a classe 2); e (b) $QP = 51$ (pior qualidade possível para as classes 0 e 1) e $QP_{\text{Texto}} = 0$ (melhor qualidade possível para a classe 2).	84
5.8	Documento composto “DSPG”.	85
5.9	Comparação objetiva entre os codificadores AVC-C, AVC-I e JPEG2000 para o documento “DSPG”: (a) PSNR global; (b) PSNR de regiões pictóricas; e (c) PSNR de regiões de texto. Note que a qualidade de regiões de texto pode ser consideravelmente aumentada enquanto a qualidade global não é demasiadamente afetada.	86
5.10	Comparação subjetiva entre codificadores: (a) exemplo de região pictórica do documento original “DSPG”; e (b) exemplo de região textual do mesmo documento.	87
5.11	Comparação subjetiva entre codificadores. Documento “DSPG” codificado e reconstruído com o AVC-C a aproximadamente 0,16 bpp: (a) região pictórica; e (b) região textual.	88
5.12	Comparação subjetiva entre codificadores. Documento “DSPG” codificado e reconstruído com o AVC-I a aproximadamente 0,16 bpp: (a) região pictórica; e (b) região textual.	89

5.13	Comparação subjetiva entre codificadores. Documento “DSPG” codificado e reconstruído com o JPEG2000 a aproximadamente 0,16 bpp: (a) região pictórica; e (b) região textual.	90
5.14	Documento composto: “compound2”.....	91
5.15	Comparação objetiva entre os codificadores AVC-C, AVC-I e JPEG2000 para o documento “compound2”: (a) PSNR global; (b) PSNR de regiões pictóricas; e (c) PSNR de regiões de texto.	92
5.16	Documento composto: “pietà”.	93
5.17	Comparação objetiva entre os codificadores AVC-C, AVC-I e JPEG2000 para o documento “pietà”: (a) PSNR global; (b) PSNR de regiões pictóricas; e (c) PSNR de regiões de texto.....	94
5.18	Documento composto: “compound1”.....	95
5.19	Comparação objetiva entre os codificadores AVC-C, AVC-I e JPEG2000 para o documento “pietà”: (a) PSNR global; (b) PSNR de regiões pictóricas; e (c) PSNR de regiões de texto.....	96
5.20	Documento composto: “lennatexto”.	97
5.21	Comparação objetiva entre os codificadores AVC-C, AVC-I e JPEG2000 para o documento “lennatexto”: (a) PSNR global; (b) PSNR de regiões pictóricas; e (c) PSNR de regiões de texto.	98
5.22	Documento composto: “shiva”.	99
5.23	Comparação objetiva entre os codificadores AVC-C, AVC-I e JPEG2000 para o documento “shiva”: (a) PSNR global; (b) PSNR de regiões pictóricas; e (c) PSNR de regiões de texto.....	100
5.24	Codificação de região de interesse: (a) imagem original, com região de interesse em destaque; (b) máscara da região de interesse; e (c) image codificada com $QP_{ROI} = 20$ e $QP = 51$ ($0,37 \text{ bits/pixel}$). Note como a qualidade da região de interesse foi preservada, apesar da degradação geral da imagem.....	102
6.1	<i>Data-filling</i> baseado nas previsões $I16$ do AVC-I: (a) máscara binária; (b) BG_0 ; (c) BG'_0 ($QP = 30$); (d) BG_{pred} ; e (e) BG_{df}	106
6.2	Codificador MRC proposto: propõe-se um codificador que utiliza o H.264/AVC na compressão dos planos FG e BG, e o JBIG2 na compressão da máscara binária.	107
6.3	(a) Documento original: “compound1”; (b) máscara binária; (c) curvas PSNR para o documento “compound1”, comparando os codificadores AVC-I, JPEG2000, MRC: FG e BG codificados com AVC-I e máscara codificada com JBIG2, e MRC: FG e BG codificados com JPEG2000 e máscara codificada com JBIG2. Fatores de ponderação considerados no algoritmo de segmentação em camadas: $\alpha_1 = 100$, $\alpha_2 = 1$, $\alpha_3 = 40$. .	108
6.4	(a) Documento original: “pietà”; (b) máscara binária; (c) curvas PSNR para o documento: “pietà”. Fatores de ponderação considerados no algoritmo de segmentação em camadas: $\alpha_1 = 500$, $\alpha_2 = 1$, $\alpha_3 = 40$	109
6.5	(a) Documento original: “lennatexto”; (b) máscara binária; (c) curvas PSNR para o documento: “lennatexto”. Fatores de ponderação considerados no algoritmo de segmentação em camadas: $\alpha_1 = 300$, $\alpha_2 = 1$, $\alpha_3 = 40$	110

6.6	(a) Documento original: “mixed1”; (b) máscara binária; (c) curvas PSNR para o documento: “mixed1”. Fatores de ponderação considerados no algoritmo de segmentação em camadas: $\alpha_1 = 500$, $\alpha_2 = 1$, $\alpha_3 = 100$.	111
6.7	(a) Documento original: “mixed2”; (b) máscara binária; (c) curvas PSNR para o documento: “mixed2”. Fatores de ponderação considerados no algoritmo de segmentação em camadas: $\alpha_1 = 500$, $\alpha_2 = 1$, $\alpha_3 = 100$.	112
6.8	(a) Documento original: “mixed3”; (b) máscara binária; (c) curvas PSNR para o documento: “mixed3”. Fatores de ponderação considerados no algoritmo de segmentação em camadas: $\alpha_1 = 500$, $\alpha_2 = 1$, $\alpha_3 = 40$.	113
6.9	(a) Documento original: “mixed4”; (b) máscara binária; (c) curvas PSNR para o documento: “mixed4”. Fatores de ponderação considerados no algoritmo de segmentação em camadas: $\alpha_1 = 500$, $\alpha_2 = 1$, $\alpha_3 = 40$.	114
6.10	Comparação subjetiva entre codificadores. Parte ampliada dos documentos originais (a) “mixed2” e (b) “mixed4”.	116
6.11	Comparação subjetiva entre codificadores. Parte ampliada dos documentos codificados e reconstruídos com o AVC-I: (a) “mixed2” e (b) “mixed4”.	117
6.12	Comparação subjetiva entre codificadores. Parte ampliada dos documentos codificados com o JPEG2000: (a) “mixed2” e (b) “mixed4”.	118
6.13	Comparação subjetiva entre codificadores. Parte ampliada dos documentos codificados com o MRC (AVC-I/JBIG2), que é o modelo proposto: (a) “mixed2” e (b) “mixed4”.	119
6.14	Comparação subjetiva entre codificadores. Parte ampliada dos documentos codificados com o MRC (JPEG2000/JBIG2): (a) “mixed2” e (b) “mixed4”.	120
7.1	Ilustração das etapas de pré-processamento, codificação, decodificação e pós-processamento.	124
7.2	Exemplos em uma dimensão do efeito da segmentação em transições suaves. Em (a) e (c) duas transições e suas respectivas máscaras, indicadas em linhas pontilhadas, podem ser observadas. No planos (b) FG e (d) BG, as respectivas regiões irrelevantes são substituídas por valores constantes, enquanto parte das transições não, causando os picos indesejáveis.	125
7.3	Picos indesejados: note o aparecimento de um contorno ao redor das bordas do texto (halo), tanto em (a) FG como em (b) BG.	125
7.4	Processo de transferência intercamada. A transferência tem por objetivo evitar que <i>pixels</i> sejam desnecessariamente processados.	127
7.5	(a) Material digitalizado original; (b) máscara M ; (c) vizinhança candidata E ; (d) <i>pixels</i> a serem processados C .	128
7.6	Imagem após pré-processamento. Note que as transições, antes suaves, agora apresentam-se aguçadas. O restante do FG e do BG permanece intacto.	128
7.7	(a) FG e (c) BG originais; (b) FG e (d) BG processados. Note o contorno ao redor das bordas dos planos originais. O pré-processamento eleva a qualidade do FG e do BG, devido à remoção dos picos indesejados.	129
7.8	Pré/pós-processamento: (a) imagem original; (b) imagem aguçada; e (c) imagem ressuavizada.	130

7.9	Comparação entre os codificadores AVC-I e MRC (com e sem pré/pós-processamento). Observe como o modelo MRC com o pré/pós-processamento proposto melhora a PSNR em mais de 1,0 dB em relação ao AVC-I e ao MRC sem pré/pós-processamento, a 0,1 <i>bit/pixel</i>	134
7.10	(a) Efeito da transformação de escala dos planos FG e BG. Os fatores de transformação de escala utilizados foram $S = 1, 1/2$ e $1/4$; e (b) desempenho final do codificador proposto.....	135
7.11	Desempenho do codificador MRC proposto, em comparação com o AVC-I: documento “horario”.....	136
7.12	Desempenho do codificador MRC proposto, em comparação com o AVC-I: documento “vetruvian”.....	137
7.13	Desempenho do codificador MRC proposto em comparação com o AVC-I: documento “thinker”.....	138
7.14	Desempenho do codificador proposto, em comparação com o AVC-I: documento “volta”.....	139
7.15	Comparação subjetiva entre codificadores. Parte ampliada do documento “volta” (Figura 7.14): (a) original; (b) codificado e reconstruído com o AVC-I (PSNR = 27,9314 dB); (c) codificado e reconstruído com o JPEG2000 (PSNR = 24,6271 dB); (d) codificado e reconstruído com o codificador MRC proposto (PSNR = 32,7260 dB). A taxa de <i>bits</i> observada foi aproximadamente 0,1 <i>bit/pixel</i>	140
7.16	Desempenho do codificador MRC proposto, em comparação com os codificadores AVC-I, JPEG2000 e MRC baseado em JPEG2000: documento “super1”.....	141
7.17	Desempenho do codificador MRC proposto, em comparação com os codificadores AVC-I, JPEG2000 e MRC baseado em JPEG2000: documento “IEEE1”.....	142
7.18	Desempenho do codificador MRC proposto, em comparação com os codificadores AVC-I, JPEG2000 e MRC baseado em JPEG2000: documento “IEEE2”.....	143
7.19	Desempenho do codificador MRC proposto, em comparação com os codificadores AVC-I, JPEG2000 e MRC baseado em JPEG2000: documento “IEEE3”.....	144
7.20	Comparação entre codificadores. Parte ampliada 1 do documento “IEEE3”: (a) original; (b) codificado e reconstruído com o AVC-I (PSNR = 26,5116 dB); (c) codificado e reconstruído com o JPEG2000 (PSNR = 23,9971 dB); (d) codificado e reconstruído com o codificador MRC proposto (PSNR = 24,2050 dB). A taxa de <i>bits</i> observada foi aproximadamente 0,19 <i>bit/pixel</i>	145
7.21	Comparação entre codificadores. Parte ampliada 2 do documento “IEEE3”: (a) original; (b) codificado e reconstruído com o AVC-I (PSNR = 26,5116 dB); (c) codificado e reconstruído com o JPEG2000 (PSNR = 23,9971 dB); (d) codificado e reconstruído com o codificador MRC proposto (PSNR = 24,2050 dB). A taxa de <i>bits</i> observada foi aproximadamente 0,19 <i>bit/pixel</i>	146
7.22	Comparação entre codificadores. Parte ampliada 3 do documento “IEEE3”: (a) original; (b) codificado e reconstruído com o AVC-I (PSNR = 26,5116 dB); (c) codificado e reconstruído com o JPEG2000 (PSNR = 23,9971 dB); (d) codificado e reconstruído com o codificador MRC proposto (PSNR = 24,2050 dB). A taxa de <i>bits</i> observada foi aproximadamente 0,19 <i>bit/pixel</i>	147

7.23	Comparação entre codificadores. Parte ampliada 4 do documento “IEEE3”: (a) original; (b) codificado e reconstruído com o AVC-I (PSNR = 26,5116 dB); (c) codificado e reconstruído com o JPEG2000 (PSNR = 23,9971 dB); (d) codificado e reconstruído com o codificador MRC proposto (PSNR = 24,2050 dB). A taxa de <i>bits</i> observada foi aproximadamente 0,19 <i>bit/pixel</i>	148
7.24	Efeito do pre/pós-processamento: (a) parte ampliada do documento “IEEE3” original; (b) parte ampliada do documento “IEEE3” ressuavizado (pré/pós-processado); e (c) parte ampliada do documento “IEEE3” não ressuavizado (pré-processado apenas).	149
7.25	Comparação entre codificadores: (a) documento “Sócrates” original; (b) parte do documento “Sócrates” original; (c) parte do documento “Sócrates”codificado com o AVC-I (PSNR = 25,3025 e $QP = 45$); (d) parte do documento “Sócrates”codificado com o JPEG2000 (PSNR = 23,7246); e (e) parte do documento “Sócrates”codificado com o MRC proposto (PSNR = 28,3906, $QP_{BG} = 41$, $QP_{FG} = 51$, $S_{BG} = 1$ e $S_{FG} = 4$). A taxa de <i>bits</i> observada foi de aproximadamente 0,2 <i>bit/pixel</i>	151
8.1	Esquema de pré-processamento das páginas do livro digitalizado.....	154
8.2	Exemplo de três páginas de um livro digitalizado: (a) página 1; (b) página 2; e (c) página 3.	155
8.3	Sequência de vídeo “samkhya” criada a partir das três páginas mostradas na Figura 8.2: os quadros de 1 a 4, 5 a 8, e 9 a 12 são criados a partir das páginas 1, 2 e 3, respectivamente.	155
8.4	Ilustração do processo de predição do macrobloco atual a partir da busca dentro de uma janela em quadros de referência anteriormente codificados.....	156
8.5	Sequência “guita”: 2 páginas, 8 quadros: (a) página inicial; (b) comparação entre o método proposto (pré-processamento das páginas e codificação utilizando o H.264/AVC), com $QR = 5$ quadros e $JB = 08, 16$ e 32 <i>pixels</i> , e a abordagem clássica, que comprime cada página independentemente (codificação utilizando o AVC-I ou o JPEG2000); e (c) comparação entre o método proposto, com $JB = 32$ <i>pixels</i> e $QR = 1, 3$ e 5 quadros, e os codificadores AVC-I e JPEG2000. Para $JB = 32$ e $QR = 5$, por exemplo, o método proposto supera o AVC-I em mais de 2 dB e o JPEG2000 em mais de 5 dB, a uma taxa de 0,5 <i>bit/pixel</i>	157
8.6	Sequência “krishnamurti”: 2 páginas, 8 quadros: (a) página inicial; e (b) comparação entre o método proposto (pré-processamento das páginas e codificação utilizando o H.264/AVC) e a abordagem clássica, que comprime cada página independentemente (codificação utilizando o AVC-I ou o JPEG2000).	158
8.7	Sequência “samkhya”: 3 páginas, 12 quadros: (a) página inicial; e (b) comparação entre o método proposto (pré-processamento das páginas e codificação utilizando o H.264/AVC) e a abordagem clássica, que comprime cada página independentemente (codificação utilizando o AVC-I ou o JPEG2000).	159
8.8	Sequência “sanskrit”: 4 páginas, 16 quadros: (a) página inicial; e (b) comparação entre o método proposto (pré-processamento das páginas e codificação utilizando o H.264/AVC) e a abordagem clássica, que comprime cada página independentemente (codificação utilizando o AVC-I ou o JPEG2000).	160

8.9 Sequência “fédon”: 5 páginas, 20 quadros: (a) página inicial; e (b) comparação entre o método proposto (pré-processamento das páginas e codificação utilizando o H.264/AVC) e a abordagem clássica, que comprime cada página independentemente (codificação utilizando o AVC-I ou o JPEG2000).	161
8.10 Sequência “principia”: 8 páginas, 32 quadros: (a) página inicial; e (b) comparação entre o método proposto (pré-processamento das páginas e codificação utilizando o H.264/AVC) e a abordagem clássica, que comprime cada página independentemente (codificação utilizando o AVC-I ou o JPEG2000).	162
8.11 Avaliação subjetiva: (a) trecho original da sétima página da sequência “principia”, bem como suas versões codificadas e reconstruídas por meio dos codificadores (b) AVC-I, (c) JPEG2000 e (d) método proposto (pré-processamento das páginas e codificação utilizando o H.264/AVC) a aproximadamente 0,25 <i>bits</i> /pixels. Observe como o método proposto fornece uma qualidade subjetiva superior.	163

LISTA DE TABELAS

5.1	Codificação de “DSPG”.....	85
5.2	Codificação de “compound2”.....	91
5.3	Codificação de “pietà”.....	93
5.4	Codificação de “compound1”.....	95
5.5	Codificação de “lennatexto”.....	97
5.6	Codificação de “shiva”.....	99

LISTA DE SÍMBOLOS

Siglas

ANSI	American National Standards Institute
AVC	Advanced Video Coding
AVC-C	Advanced Video Coding - Compound
AVC-I	Advanced Video Coding - INTRA
BG	Background
CABAC	Context-based Binary Arithmetic Coding
CCITT	Comité Consultatif International de Télégraphique et Téléphonique
CDF	Cohen-Daubechies-Feauveau
CODEC	Codificador-Decodificador
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
FG	Foreground
FRExt	Fidelity Range Extensions
HVS	Human Visual System
I4	Intra 4×4
I8	Intra 8×8
I16	Intra 16×16
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
ITU	International Telecommunication Union
JBIG	Joint Bi-level Image Processing Group
JPEG	Joint Photograph Experts Group
JVT	Joint Video Team
MMP	Multidimensional Multiscale Parser
MPEG	Motion Picture Experts Group
MRC	Mixed Raster Content
PDF	Portable Document File
PS	PostScript
PSNR	Peak Signal-to-noise Ratio
QP	Quantization Parameter
VCEG	Video Coding Experts Group

Capítulo 1

Introdução

1.1 Contexto

Documentos digitais estão se tornando cada vez mais comuns. Recentemente, por exemplo, o jornal inglês *The Times*, publicado diariamente desde 1785, disponibilizou todo seu acervo na *Internet*, gratuitamente, até o ano de 1985 [1]. Um outro exemplo é o portal *Domínio Público* [2], mantido pelo Governo Federal, que propõe o compartilhamento de conhecimento, também gratuito, por meio de um arquivo digital. A idéia de uma Biblioteca Mundial que possa concentrar toda a produção intelectual da humanidade passa a ser uma realidade possível. Uma iniciativa como essa tornaria acidentes como o ocorrido com a antiga Biblioteca de Alexandria menos danosos, uma vez que em meio digital os documentos podem ser mais facilmente replicados. Além disso, sabe-se que a decomposição é um processo natural que não pode ser interrompido, apenas desacelerado. Alguns livros, por serem muito antigos, mal podem ser manuseados. Transferí-los para o meio digital seria uma maneira de torná-los potencialmente perpétuos e acessíveis a todos. Por fim, a representação digital do conhecimento humano abriria novas fronteiras à pesquisa. Basta imaginar pesquisadores podendo ter acesso quase que imediato a manuscritos originais, dentro de suas respectivas áreas de atuação.

1.2 Apresentação do Problema e Justificativa

A geração de um acervo de documento digitais implica a necessidade de se ter meios de armazenamento. E isso se traduz em espaço físico e consumo de energia. Quanto a acessar o conteúdo digital, assume-se que aquele que o requisita deseja recebê-lo no menor tempo possível. Sendo assim, a forma de se representar um documento torna-se uma preocupação, o que estimula o desenvolvimento de técnicas que ofereçam uma representação binária mínima. Dadas a perspectiva de se armazenar a produção intelectual humana em meio digital e a necessidade de se realizar tal propósito de forma econômica, a presente tese justifica-se por reunir o que há de mais avançado em codificação de imagens com o objetivo de propor novos métodos que viabilizam a compressão eficiente de documentos digitais.

1.3 Definições Iniciais

Para que o restante do texto seja melhor compreendido, será feita uma distinção conceitual entre *documento digital*, *documento digitalizado* e *documento gerado eletronicamente*. Documento digital é aquele cujo suporte são os meios de armazenamento digital como o CD, o DVD, o *Hard Disk*, entre outros. Tais documentos podem ser manipulados com o auxílio de um computador digital. Um documento digitalizado é um documento digital cujo suporte original não é o meio digital, mas sim o papel. O documento gerado eletronicamente é um documento digital, cuja elaboração e armazenamento ocorreram diretamente no meio digital e, por isso, o papel não aparece como suporte intermediário. Assim, os documentos digitalizados e os documentos gerados eletronicamente são subcategorias dos documentos digitais, conforme ilustrado na Figura 1.1.

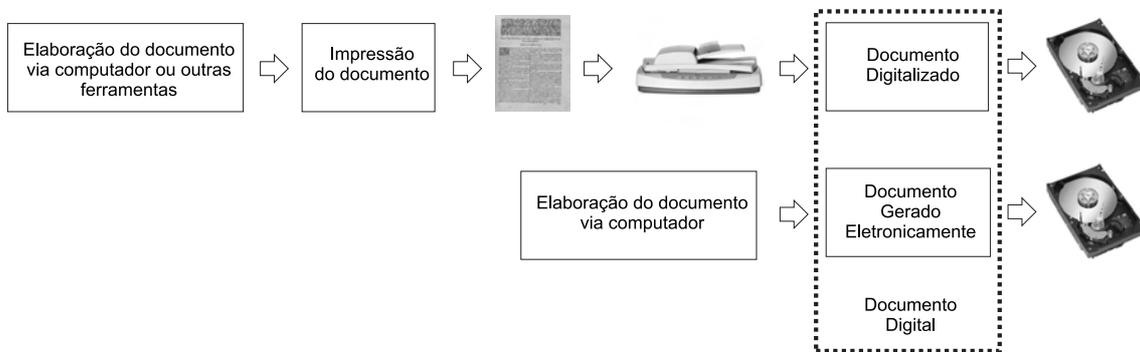


Figura 1.1: Ilustração da distinção feita entre documento digitalizado e documento gerado eletronicamente, ambos considerados documentos digitais.

1.4 Compressão de Imagens

A *compressão de imagens* é um subtópico do *Processamento Digital de Imagens* [3] e estuda técnicas por meio das quais se consegue reduzir a quantidade de dados necessária para se representar uma imagem. Vários algoritmos de compressão foram propostos ao longo da história do Processamento Digital de Imagens: alguns deles voltados para a compressão de *imagens binárias*, como o JBIG [4] e o JBIG2 [5], outros destinados à compressão de *imagens em tons contínuos*, como o JPEG [6] e o JPEG2000 [7]. Mais recentemente, ainda, o *Multidimensional Multiscale Parser* (ou MMP) [8] foi proposto como um codificador universal de imagens.

O H.264/AVC [9], o mais novo padrão para compressão de vídeo, aparece também como uma opção no que se refere à compressão de *imagens estáticas*. Originalmente não foi concebido para tal finalidade, mas, apesar disso, os avanços incorporados ao padrão não apenas estabeleceram um novo patamar em compressão de vídeo, mas também fizeram dele um excelente codificador de imagens em tons contínuos.

1.5 Compressão de Documentos Digitais

Algoritmos de compressão são geralmente desenvolvidos tendo-se em mente um tipo particular de imagem e aplicação. Quando se comprime texto, por exemplo, é importante preservar as bordas e o formato dos caracteres para que a leitura seja facilitada. O sistema visual humano, por outro lado, trabalha diferentemente para imagens em tons contínuos, melhor mascarando erros de alta frequência. Dessa maneira, algoritmos destinados à compressão de documentos digitais, para serem eficientes, precisam levar em conta a diversidade de conteúdo. Uma página de revista pode apresentar imagens naturais, textos e gráficos, por exemplo, e o algoritmo de compressão deve ser capaz de codificar essa página atendendo às exigências específicas impostas por esses elementos.

Geralmente, a compressão de documentos digitais é realizada a partir de dois paradigmas diferentes. O primeiro deles, baseado em um único codificador, procura ajustar seus parâmetros de configuração à medida em que o documento vai sendo codificado, na tentativa de comprimir cada região da melhor forma possível. O outro, baseado em múltiplos codificadores, procura decompor o documento em várias camadas, aplicando algoritmos de compressão específicos a cada uma delas. Tal é o *modus operandi* do modelo de imageamento conhecido como *Mixed Raster Content* (ou MRC) [10].

1.6 Métodos Propostos

Como dito anteriormente, o H.264/AVC passou a ser visto como uma opção no que se refere à compressão de imagens em tons contínuos. Sua eficiência em taxa e distorção até mesmo supera, em muitos casos, o JPEG2000. Em se tratando de documentos digitais caracterizados pela presença simultânea de imagens naturais, textos e gráficos, a diferença de desempenho entre o H.264/AVC e o JPEG2000 chega a ser ainda mais pronunciada. A observação de tal fenômeno fez com que as contribuições da presente tese tivessem esse padrão como elemento funcional básico. Vislumbra-se um futuro em que os avanços representados pelo H.264/AVC irão servir de base para a definição de codificadores de imagens mais eficientes. A seguir, os quatro métodos propostos para compressão de documentos digitais serão descritos, bem como as contribuições associadas a cada um deles:

1. **Método 1** - *Advanced Video Coding - Composto*: a principal contribuição desse método foi agregar ao H.264/AVC a possibilidade de se codificar adaptativamente regiões de texto e imagem a partir da análise do conteúdo de macroblocos. Além disso, oferece um algoritmo para seleção dos parâmetros de quantização que atendem a critérios de taxa e distorção impostos pelo usuário. Uma outra contribuição foi tornar o H.264/AVC apto a codificar regiões de interesse.
2. **Método 2** - *Compressão MRC baseada em H.264/AVC-I e JBIG2 para Documentos Gerados Eletronicamente*: na compressão MRC, espera-se um melhor desempenho em taxa e distorção quando cada camada é individualmente codificada por meio de algoritmos mais eficientes. O método propõe um modelo MRC baseado no H.264/AVC e no JBIG2, além de um algoritmo para preenchimento de regiões “irrelevantes” baseado nas previsões intra do H.264/AVC.

3. **Método 3** - *Compressão MRC baseada em H.264/AVC-I e JBIG2 para Documentos Digitalizados*: o modelo MRC é bastante adequado à compressão de documentos gerados eletronicamente. Em documentos dessa natureza, as transições de caracteres são suficientemente abruptas e podem ser perfeitamente modeladas a partir de um máscara binária. Em documentos digitalizados, por outro lado, as bordas de texto são caracterizadas por transições suaves, fazendo com que a segmentação em planos seja dificultada. A contribuição do presente método está em um algoritmo que, no codificador, parametriza a suavidade das transições dos caracteres, torna suas bordas abruptas, realiza a compressão MRC baseada no H.264/AVC e no JBIG2, e reestabelece, no decodificador, a suavidade original do documento.
4. **Método 4** - *Compressão de Livros Digitalizados utilizando o H.264/AVC*: a contribuição desse método está na utilização dos modos de predição inter do H.264/AVC como uma forma de se realizar a compressão de livros a partir do casamento aproximado de padrões. Tal proposta é implementada por meio do pré-processamento das páginas a serem codificadas, de forma a transformá-las em uma sequência de vídeo posteriormente comprimida pelo H.264/AVC.

Vários experimentos foram realizados e os resultados mostram que os métodos propostos apresentam um ganho de desempenho objetivo e/ou subjetivo com relação a abordagens já conhecidas.

1.7 Organização da Tese

A tese está dividida em nove capítulos. O primeiro é constituído pela presente introdução. Os Capítulos 2, 3 e 4 destinam-se à revisão da literatura técnica, necessária ao pleno entendimento dos métodos propostos nos Capítulos 5, 6, 7 e 8.

No Capítulo 2 serão tratados conceitos básicos em processamento digital de imagens, tais como aquisição e representação digital de imagens, padrões de compressão de imagens binárias (JBIG e JBIG2) e de imagens em tons contínuos (JPEG e JPEG2000).

O Capítulo 3 traz conceitos sobre compressão de vídeo e apresenta o padrão H.264/AVC. Dentro desse contexto, fala sobre a predição intra, a otimização taxa-distorção lagrangeana, a codificação aritmética binária adaptativa baseada em contexto, e por fim chama a atenção para o fato de que o H.264/AVC operando no modo intra torna-se um excelente compressor de imagens em tons contínuos, superando, inclusive, padrões considerados o estado-da-arte em compressão de imagens estáticas.

O Capítulo 4 explora alguns conceitos relativos à compressão de documentos digitais. Apresenta a compressão de documentos baseada em um codificador único, a compressão de documentos baseada em múltiplos codificadores, a segmentação em camadas e o preenchimento de regiões “irrelevantes” (ou *Data-filling*).

Os Capítulos 5, 6, 7 e 8 apresentam as contribuições da presente tese, distribuídas por quatro métodos destinados à compressão de documentos.

Por fim, o Capítulo 9 é destinado à apresentação das conclusões, bem como das propostas para

trabalhos futuros. As publicações que resultaram da pesquisa desenvolvida também são listadas nesse capítulo.

Capítulo 2

Compressão de Imagens

2.1 Processamento Digital de Imagens

O *Processamento Digital de Imagens* [3, 11, 12, 13, 14, 15, 16, 17] é uma sub-disciplina associada ao *Processamento Digital de Sinais* [18, 19, 20, 21, 22]. Assim, é a *Engenharia Elétrica* [23] a responsável por ter desenvolvido uma tradição nessa área.

A origem do Processamento Digital de Imagens está intimamente relacionada ao aparecimento de computadores digitais capazes de realizar tarefas de processamento realmente significativas. Dessa maneira, pode-se dizer que tal disciplina teve sua origem no início dos anos 60 [3].

A presente seção apresenta alguns conceitos básicos relacionados ao Processamento Digital de Imagens.

2.1.1 Aquisição de Imagens

Uma *cena* pode ser definida como uma entidade tridimensional formada a partir da reflexão de energia radiante. Um fluxo de radiação de uma fonte, ao se propagar pelo espaço, pode interagir com a superfície dos objetos de uma cena. A energia absorvida pelos objetos geralmente provoca aquecimento. Já a energia refletida carrega informações a respeito desses objetos. As propriedades observadas no fluxo de energia refletido dependem diretamente das propriedades físico-químicas dos elementos com os quais o fluxo de energia emitido pela fonte interagiu. Sensores que sejam sensíveis à faixa de radiação refletida podem registrar tais propriedades.

Sistemas construídos com a finalidade de se capturar uma cena real e registrá-la em uma *imagem* são chamados *sistemas de imageamento*. Um dos exemplos mais conhecidos é o formado por máquina e filme fotográficos, que registram as propriedades de ondas eletromagnéticas capazes de sensibilizar o sistema visual humano (HVS ou, do inglês, *Human Visual System*) [14]. Um sistema de imageamento pode ter apenas um sensor ou vários sensores dispostos em tira ou em matriz, conforme ilustrado na Figura 2.1.

Assim, torna-se evidente que, apesar da cena ser uma entidade tridimensional, uma imagem pode ser representada por uma função bidimensional, $f(x, y)$, onde x e y são coordenadas espaciais

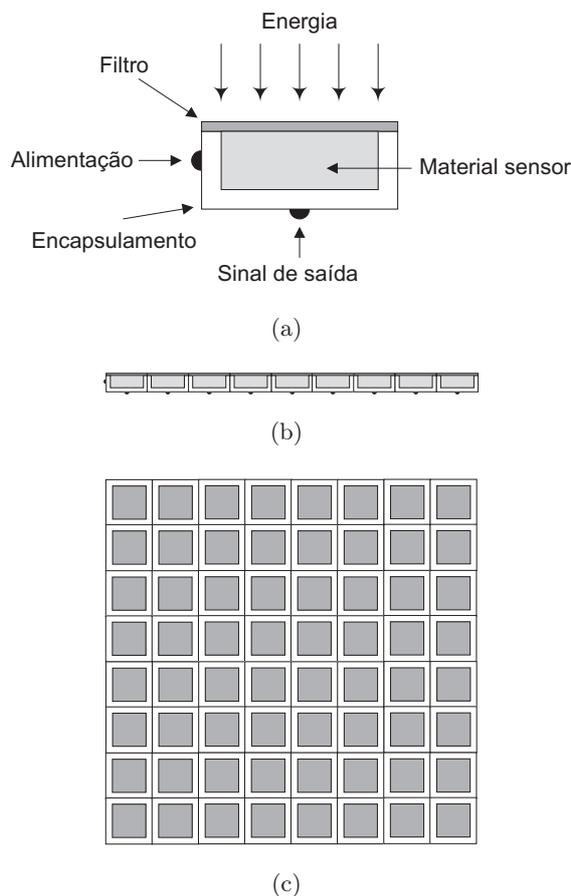


Figura 2.1: Sistema de imageamento: (a) sensor único; (b) sensores dispostos em vetor; e (c) sensores dispostos em matriz.

e a amplitude de f no ponto (x, y) é a intensidade da imagem no referido ponto. Convencionase situar a origem dos eixos de uma imagem no canto superior esquerdo, conforme ilustrado na Figura 2.2.

Quando gerada por meio de um processo físico, a função $f(x, y)$ pode ser caracterizada a partir da quantidade de luz incidente em uma cena, $i(x, y)$, e da proporção de luz refletida pelos objetos que a compõem, $r(x, y)$, conforme a Equação 2.1 [3]. Tais componentes são denominados *iluminamento* [24] e *reflectância*, respectivamente.

$$f(x, y) = i(x, y)r(x, y) \quad (2.1)$$

Teoricamente os valores de $i(x, y)$ podem variar de 0 a $\infty \text{ lm/m}^2$ (*lúmens por metro quadrado* ou *lux*), enquanto os valores de $r(x, y)$ podem variar de 0 a 1. Assim, é fácil deduzir que $f(x, y)$ pode apresentar valores entre 0 e ∞ . A Figura 2.3 ilustra o processo de aquisição de uma imagem digital.

Não constitui objetivo do presente trabalho abordar conceitos referentes à *Ciência das Cores* [25, 26]. No entanto, algumas definições fazem-se necessárias.

Imagens em escala de cinza, também chamadas de *monocromáticas*, são definidas como sendo

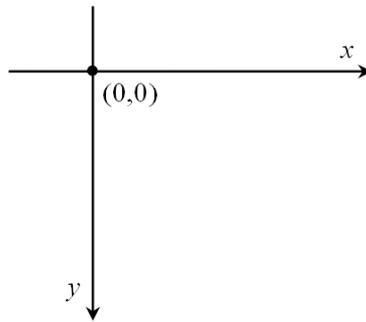


Figura 2.2: Convenciona-se situar a origem dos eixos de uma imagem no canto superior esquerdo.

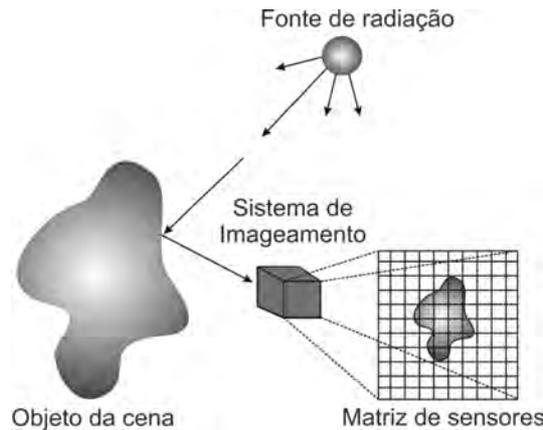


Figura 2.3: Ilustração de um sistema de imageamento.

aquelas que não carregam informação de cor. Para tais imagens, $f(x, y)$ assume apenas valores de intensidade que são percebidos como variando do branco ao preto, passando por níveis de cinza. O intervalo que contém os possíveis níveis de cinza para $f(x, y)$ é chamado de *escala de cinza*. A Figura 2.4 mostra um exemplo de imagem em escala de cinza.

Imagens coloridas são aquelas que carregam informação de cor. No espaço de cores *RGB* (*Red*, *Green*, *Blue*), por exemplo, a imagem é definida a partir de três imagens componentes, cada qual responsável por registrar as intensidades de vermelho, verde e azul refletidas por uma cena. A Figura 2.5 mostra um exemplo de imagem colorida, acompanhada de suas três componentes *RGB*.

O *RGB*, porém, não é necessariamente o espaço mais eficiente para se representar cores. Sabe-se que o sistema visual humano é mais sensível à *luminância* (brilho) do que à cor. No entanto, o *RGB* não tira proveito desse fato. Nesse espaço de cores, a luminância está espalhada pelas componentes *R*, *G* e *B*, tornando-as igualmente relevantes. Uma outra representação muito popular, denominada *YC_bC_r* [27], separa a componente de luminância *Y* das componentes de cor, *C_b* e *C_r*, também chamadas de *crominância*. A componente *Y* pode ser vista como uma versão monocromática da imagem colorida. A conversão de uma imagem *RGB* em *YC_bC_r*, e vice-versa, pode ser realizada por meio das Equações 2.2 e 2.3, respectivamente. A principal vantagem da representação *YC_bC_r* é que as componentes *C_b* e *C_r* podem ser amostradas a uma resolução espacial (ver Seção 2.1.2) mais baixa que a componente *Y*, sem que haja uma queda significativa na qualidade visual da imagem. A Figura 2.6 ilustra os espaços de cores *RGB* e *YC_bC_r*. Já a Figura 2.7 mostra as componentes



Figura 2.4: Imagem “goldhill”: exemplo de imagem em escala de cinza.



Figura 2.5: Imagem “Lena”: exemplo de imagem colorida, acompanhada de suas três componentes RGB .

YC_bC_r para a imagem “Lena”.

$$\begin{aligned}
 Y &= 0,299R + 0,587G + 0,114B \\
 C_b &= 0,564(B - Y) \\
 C_r &= 0,713(R - Y)
 \end{aligned}
 \tag{2.2}$$

$$\begin{aligned}
 R &= Y + 1,402C_r \\
 G &= Y - 0,344C_b - 0,714C_r \\
 B &= Y + 1,772C_b
 \end{aligned}
 \tag{2.3}$$

Há uma outra classe de imagens que engloba representações com apenas dois níveis de cinza, o preto e o branco, conforme ilustrado na Figura 2.8¹, e, por isso, são chamadas de *binárias*. Uma imagem binária Y_b pode ser obtida a partir de uma imagem em escala de cinza Y_c , por meio da Equação 2.4. Tal equação determina que *pixels* em Y_c cujos valores são menores ou iguais a um limiar τ recebem o valor 0 em sua versão binária Y_b . Os demais *pixels* de Y_b recebem o valor 1.

¹Música do folclore brasileiro. Arranjo do violonista Eustaquio Grilo, professor do Departamento de Música da Universidade de Brasília.

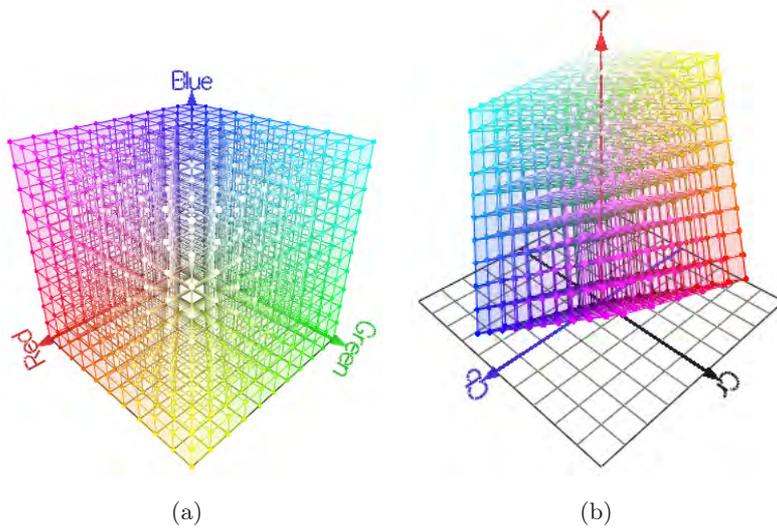


Figura 2.6: Espaços de cores: (a) RGB ; e (b) YC_bC_r .



Figura 2.7: Imagem “Lena” representada no espaço de cores YC_bC_r : (a) luminância Y ; (b) componente de crominância C_b ; e (c) componente de crominância C_r .

$$Y_b(i, j) = \begin{cases} 0 & : Y_c(i, j) \leq \tau \\ 1 & : Y_c(i, j) > \tau \end{cases} \quad (2.4)$$

Imagens em escala de cinza e coloridas também são denominadas *imagens em tons contínuos*, em oposição às binárias [3]. No presente trabalho serão tratadas apenas imagens adquiridas em escala de cinza. Assim, de agora em diante, o termo “imagem” irá se referir exclusivamente a essa classe de imagens.

2.1.2 Representação Digital de Imagens

Uma imagem pode ser contínua com respeito às coordenadas espaciais x e y , e com respeito à intensidade $f(x, y)$. O processo de digitalização é resultado da discretização tanto das coordenadas espaciais como da intensidade.

A *amostragem* consiste em discretizar o domínio de definição da função $f(x, y)$, transformando-



Figura 2.8: Imagem “acanoavirou”: exemplo de imagem binária ou “preto e branco”.

o em uma grade de pontos regularmente espaçados entre si. Se δx e δy são intervalos nas direções x e y , respectivamente, então, após a amostragem, $f(x, y)$ passa a ter valores definidos apenas para os pontos $(n\delta x, m\delta y)$, onde n e $m \in \mathbb{N}$. Tal procedimento define o menor detalhe discernível em uma cena, o que caracteriza a *resolução espacial* da imagem.

A *quantização*, por sua vez, define um conjunto finito de níveis de cinza com os quais uma imagem pode ser representada. Assim, para um intervalo de nível de cinza dado por δz , $f(n\delta x, m\delta y)$ só apresentará valores definidos por $k\delta z$, onde $k \in \mathbb{N}$. É importante lembrar que os pontos $(n\delta x, m\delta y)$ devem estar contidos no domínio $[0, r] \times [0, s]$, ou seja, $0 \leq n\delta x \leq r$ e $0 \leq m\delta y \leq s$, onde r e s representam os limites espaciais da imagem. Da mesma maneira, $k\delta z$ deve ser menor ou igual ao maior nível de cinza da imagem. A quantização define, dessa forma, o que é chamado de *resolução em nível de cinza*.

Como x , y e $f(x, y)$ assumem um número finito de valores, deduz-se que uma imagem pode ser representada por um conjunto finito de elementos básicos aos quais dá-se o nome de *elementos de imagem* ou, simplesmente, *pixels* (*picture elements*).

É muito comum, quando se trabalha com imagens digitais, ignorar os intervalos de discretização δx , δy e δz . Dessa maneira, uma imagem pode ser vista como uma matriz de *pixels*, $F(m, n)$, composta por M linhas e N colunas (dita M por N , ou $M \times N$), cada *pixel* podendo assumir um dos K possíveis níveis da escala de cinza [14]. Na representação matricial, os índices m e n indicam apenas o número da linha e da coluna de F , respectivamente, e não uma coordenada física real. É importante observar que os valores de $m\delta y$ e $n\delta x$, anteriormente situados sobre o eixo vertical y e sobre o eixo horizontal x do plano cartesiano, são mapeados para o índice de linha m e o índice de coluna n da matriz F , respectivamente. É comum utilizar $M \times N$ como sendo a resolução espacial e K como a resolução em nível de cinza da imagem.

A discussão anterior sugere a seguinte representação matemática para uma imagem:

$$F(m, n) = \begin{bmatrix} F(0, 1) & F(0, 2) & \dots & F(0, N - 1) \\ F(1, 1) & F(1, 2) & \dots & F(1, N - 1) \\ F(2, 1) & F(2, 2) & \dots & F(2, N - 1) \\ \vdots & \vdots & \ddots & \vdots \\ F(M - 1, 1) & F(M - 1, 2) & \dots & F(M - 1, N - 1) \end{bmatrix}. \quad (2.5)$$

Tipicamente, a quantidade de níveis de cinza de uma imagem é dada por uma potência de 2, conforme mostra a Equação 2.6,

$$K = 2^b. \quad (2.6)$$

Assim, se cada *pixel* da imagem é representado por b *bits* (*binary digits*), a escala de cinza dessa imagem possui K níveis igualmente espaçados dentro do intervalo $[0, K - 1]$, onde 0 e $K - 1$ estão para o preto e o branco, respectivamente [3]. Se $b = 8$ *bits*, por exemplo, os valores de $F(m, n)$ estão entre 0 e 255. Fica claro, então, que $F(m, n)$ não armazena valores físicos reais de intensidade luminosa, mas sim índices numéricos inteiros que representam tais valores.

A Figura 2.9 ilustra uma imagem composta por $M = 8$ linhas e $N = 8$ colunas e $K = 256$ níveis de cinza. Cada quadrado menor representa um *pixel*. Os números no interior dos *pixels* indicam os níveis de cinza a que pertencem.

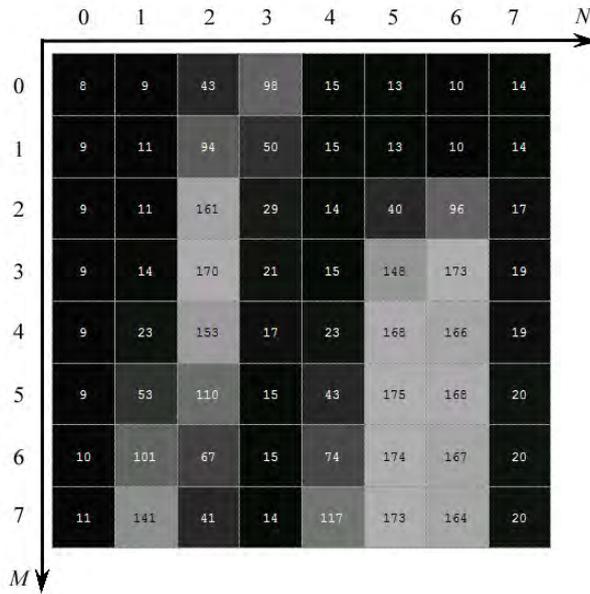


Figura 2.9: Exemplo de imagem com $M = 8$ linhas, $N = 8$ colunas e $K = 256$ níveis de cinza. Cada quadrado representa um *pixel*. O número no interior dos *pixels* indica o nível de cinza a que pertencem.

Por fim, o número de *bits* necessários para se armazenar uma imagem $M \times N$ com K níveis de cinza é dado pela Equação 2.7,

$$N_{bits} = M \times N \times b. \quad (2.7)$$

Para o exemplo da Figura 2.9, $N_{bits} = 8 \times 8 \times 8 = 512$ *bits*.

Partindo-se da idéia de que o armazenamento e a transmissão de imagens constituem uma necessidade, a maioria dos formatos de arquivo de imagem incorpora técnicas de *compressão*. De uma maneira bastante simplificada, a compressão consiste em se gerar uma representação que ocupe uma menor quantidade de *bits* que a representação original. Alguns conceitos relacionados à compressão de imagens, bem como os padrões de compressão que foram utilizados no presente trabalho, serão brevemente descritos a seguir.

2.2 Padrões de Compressão de Imagens

A *compressão de imagens* é um subtópico do Processamento Digital de Imagens e estuda técnicas por meio das quais se consegue reduzir a quantidade de dados necessária para se representar uma imagem. Tal redução é alcançada pela eliminação de informações *visualmente irrelevantes* ou *estatisticamente redundantes* de acordo com processos que podem ou não resultar em perdas. A compressão dita *sem perdas* (*lossless*), também chamada de *codificação de entropia* [28, 29], consiste na remoção de informações estatisticamente redundantes e é capaz de gerar uma versão comprimida da imagem a partir da qual é possível obter uma versão reconstruída F_r idêntica à original F_o . Já a compressão dita *com perdas* (*lossy*) consiste na eliminação de informações visualmente irrelevantes e gera uma versão comprimida a partir da qual a reconstrução perfeita não é possível [30, 12, 31]. Um programa ou um dispositivo desenvolvido com o objetivo de se comprimir uma imagem é chamado de *codificador*. Já a decodificação é realizada por um *decodificador*. Assim, o par CODificador/DECodificador é chamado de *CODEC* [27].

A seguir serão brevemente descritos os padrões JBIG e JBIG2, utilizados na compressão de imagens binárias (preto e branco), e os padrões JPEG e JPEG2000, utilizados na compressão de imagens em tons contínuos (coloridas e em escala de cinza).

2.2.1 Padrões de Compressão de Imagens Binárias

Em várias aplicações de compressão, a imagem binária é representada por um fluxo de dígitos binários unidimensional, organizado a partir de uma varredura da imagem que vai da esquerda para a direita e de cima para baixo (também chamada de varredura *raster*), conforme ilustrado na Figura 2.10. Até o início da década de 1980, algoritmos desenvolvidos com a finalidade de se comprimir tais sequências de *bits* eram essencialmente baseados no conceito de *alfabeto estendido* [31], onde não apenas dígitos isolados mas também grupos de *bits* passam a ser eventos estatísticos a serem codificados. Sabe-se, porém, que tais algoritmos podem tornar-se computacionalmente impraticáveis quando a busca pela eficiência em compressão acaba por gerar dicionários muito extensos. Além disso, a reorganização da imagem em um estrutura unidimensional não permite explorar estatisticamente a ocorrência de padrões bidimensionais naturais, como, por exemplo, os caracteres de um texto. Em 1981, Langdon e Rissanen [32] propuseram uma solução para esse problema, quando descreveram um procedimento no qual a *codificação aritmética* [33, 34, 35, 31] adaptativa a um contexto² bidimensional poderia ser utilizada na compressão de imagens em preto

²Conjunto de *pixels* situados na vizinhança imediata do *pixel* a ser codificado.

e branco. A codificação aritmética adaptativa³, aliada à utilização de contextos bidimensionais, possibilitou um compromisso mais eficiente entre a taxa de compressão e a complexidade algorítmica [36]. Os citados pesquisadores estabeleceram, dessa forma, um ponto de referência no que diz respeito à compressão de imagens dessa categoria, influenciando os padrões que serão brevemente descritos nas seções 2.2.1.3 e 2.2.1.4.

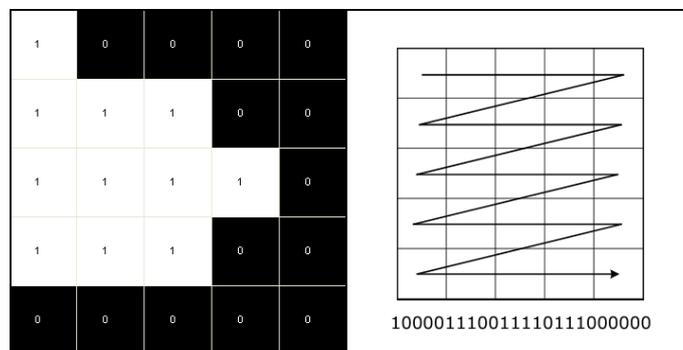


Figura 2.10: Imagem binária representada como um fluxo de dígitos binários unidimensional, organizado a partir da varredura *raster*.

É importante destacar que no âmbito do presente trabalho a compressão de imagens binárias será sempre sem perdas. Assim, antes de prosseguir, faz-se necessário compreender o conceito de *entropia*, conforme definido pela *Teoria da Informação*.

2.2.1.1 O Conceito de Entropia

Durante algum tempo tentou-se definir uma medida quantitativa para a informação. Foi Claude Elwood Shannon, porém, que em 1948 lançou as bases para o que ficou conhecido como Teoria da Informação [28, 29].

Inicialmente, Shannon definiu uma quantidade denominada *autoinformação* (*self-information*), conforme exposto a seguir. Seja um evento A , o qual é um conjunto de saídas de algum experimento aleatório. Se $P(A)$ é a probabilidade de ocorrência desse evento, então a autoinformação associada a A é dada pela Equação 2.8,

$$i(A) = \log_b \frac{1}{P(A)} = -\log_b P(A). \quad (2.8)$$

A unidade de informação depende da base do \log . Se a base b for igual a 2, a unidade é o *bit*. Por exemplo, se *Cara* e *Coroa* são as possíveis saídas para o lançamento de uma moeda viciada, e

$$P(\text{Cara}) = 1/3 \quad e \quad P(\text{Coroa}) = 2/3,$$

então,

$$i(\text{Cara}) = 1,5850 \text{ bits} \quad e \quad i(\text{Coroa}) = 0,5850 \text{ bit}.$$

Ou seja, a ocorrência de *Cara* carrega mais informação que a ocorrência de *Coroa*.

³O uso do termo “adaptativa” refere-se ao fato das probabilidades de ocorrência de eventos estatísticos serem atualizadas à medida em que a codificação acontece.

Considere agora um conjunto de eventos independentes A_i , os quais são conjuntos de saídas de um determinado experimento. A autoinformação média associada ao experimento é definida pela Equação 2.9,

$$H = \sum P(A_i) \cdot i(A_i) = - \sum P(A_i) \cdot \log_b P(A_i). \quad (2.9)$$

A quantidade H também é chamada de *entropia*. Se A_i são símbolos gerados por uma fonte, então a entropia pode ser entendida como a quantidade média de *bits* necessária para se codificar a saída da fonte. A compressão sem perdas, por mais eficiente que seja, não pode codificar a saída de uma fonte a uma quantidade média de *bits* inferior a H . Normalmente, não é possível conhecer *a priori* a entropia de uma fonte física e, por isso, deve-se estimá-la. Considerando, por exemplo, a sequência,

1 1 2 2 2 3 3 2 2 1 1 1 2 3 3 2 2 1 1 2,

e assumindo que as frequências de ocorrência dos números 1, 2 e 3 estão bem caracterizadas pela quantidade de vezes que tais números ocorrem, então, a entropia da fonte pode ser estimada por,

$$H = -P(1) \cdot \log_2 P(1) - P(2) \cdot \log_2 P(2) - P(3) \cdot \log_2 P(3) = 1,5129 \text{ bits}.$$

Isso significa que o esquema de compressão sem perdas mais eficiente que se pode construir para tal sequência é capaz de codificá-la a uma taxa mínima de 1,5129 *bits/amostra*.

A seguir serão tratados os aspectos gerais da codificação aritmética.

2.2.1.2 Codificação Aritmética

Conforme discutido anteriormente, a codificação aritmética [33, 34, 35, 31] é apresentada como uma alternativa prática à codificação baseada em alfabeto estendido como, por exemplo, a *codificação de Huffman* [37, 38]. A presente seção tem por finalidade apresentar os princípios gerais da codificação aritmética, o que será realizado por meio de um exemplo ilustrativo.

Seja Λ um alfabeto composto por três letras,

$$\Lambda = \{a_1, a_2, a_3\}, \quad (2.10)$$

com $P(a_1) = 0,8$, $P(a_2) = 0,15$ e $P(a_3) = 0,05$. Considere, também, a *função de densidade acumulada*, definida por,

$$F_x(i) = \sum_{k=1}^i P(a_k), \quad (2.11)$$

a partir da qual é possível calcular os valores $F_x(1) = 0,8$, $F_x(2) = 0,95$ e $F_x(3) = 1,0$, utilizados na divisão do intervalo unitário, $[0, 1, 0]$, em três subintervalos,

$$[F_x(i-1), F_x(i)], \text{ para } i = 1, 2 \text{ e } 3, \quad (2.12)$$

conforme ilustrado na Figura 2.11, Etapa 1.

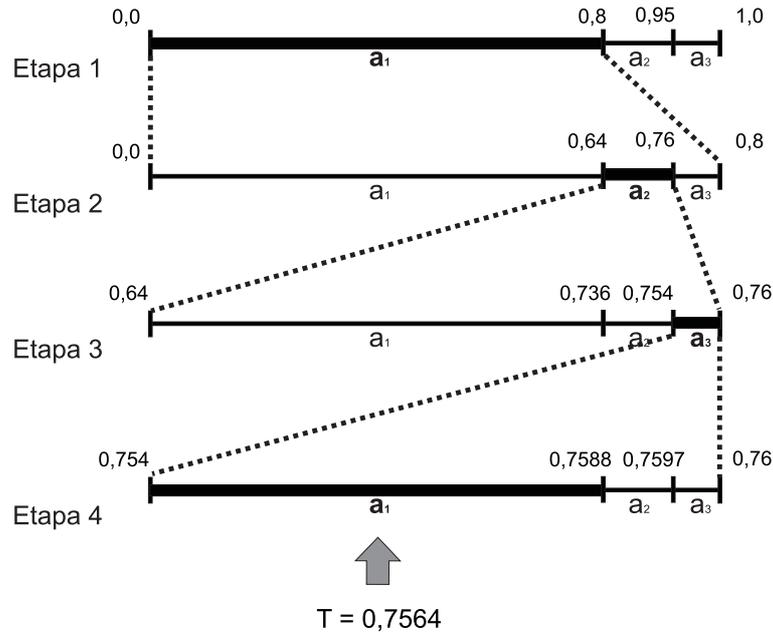


Figura 2.11: Ilustração da codificação aritmética.

Suponha, agora, que a sequência de símbolos $\mathbf{x} = a_1 a_2 a_3 a_1$ deva ser codificada. Como o primeiro símbolo é o a_1 , divide-se o subintervalo referente a esse símbolo, $[0, 0, 8]$, em três novos subintervalos, mantendo-se a mesma proporção utilizada na divisão do intervalo unitário, $[0, 0, 1, 0]$, conforme ilustrado na Etapa 2 da Figura 2.11. O segundo símbolo da sequência é o a_2 . Isso faz com que, na Etapa 3, o subintervalo referente a esse símbolo, $[0, 64, 0, 76]$, seja subdividido em três novos subintervalos, empregando-se a mesma proporção anteriormente utilizada. Em seguida, o símbolo a_3 é recebido pelo codificador. O subintervalo referente a esse símbolo, $[0, 754, 0, 760]$, é dividido em três outros, conforme ilustrado na Etapa 4. Por fim, o codificador recebe o último símbolo, a_1 . Calcula-se o ponto médio do intervalo que o contém, $[0, 754, 0, 7588]$, e utiliza-se esse valor como um rótulo para toda a sequência, ou seja,

$$T = \frac{0,754 + 0,7588}{2} = 0,7564. \quad (2.13)$$

Tal rótulo passa a ser, então, uma representação única para a sequência \mathbf{x} . Deseja-se, porém, gerar um código binário, também único, para T . E isso é possível apenas se sua representação no sistema de numeração binário for truncada⁴ no mínimo em,

$$1 + \left\lceil \log_2 \frac{1}{P(\mathbf{x})} \right\rceil \text{ bits}, \quad (2.14)$$

onde $P(\mathbf{x})$ é a probabilidade de ocorrência da sequência \mathbf{x} e $\lceil \cdot \rceil$ é um operador que retorna o menor inteiro maior que o argumento. No exemplo,

$$P(\mathbf{x}) = P(a_1) \cdot P(a_2) \cdot P(a_3) \cdot P(a_1). \quad (2.15)$$

⁴Truncar, do latim *truncare*, verbo transitivo: significa separar do tronco. No sentido figurado, o adjetivo *truncado* é utilizado aqui para passar a idéia de uma estrutura abruptamente terminada.

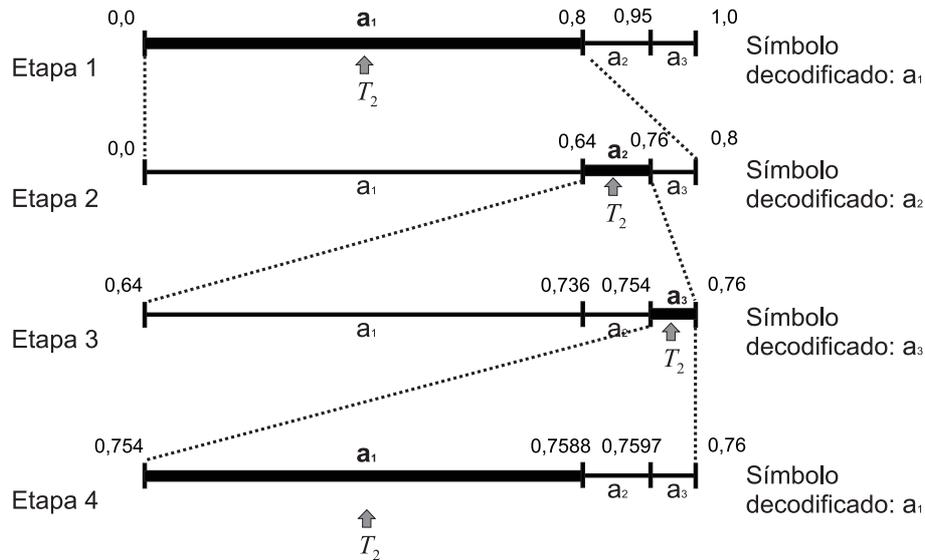


Figura 2.12: Ilustração da decodificação aritmética.

Logo, segundo a Equação 2.14,

$$1 + \left\lceil \log_2 \frac{1}{P(a_1) \cdot P(a_2) \cdot P(a_3) \cdot P(a_1)} \right\rceil =$$

$$1 + \left\lceil \log_2 \frac{1}{P(a_1)} + \log_2 \frac{1}{P(a_2)} + \log_2 \frac{1}{P(a_3)} + \log_2 \frac{1}{P(a_1)} \right\rceil = 9 \text{ bits},$$

e a aproximação binária para $T = 0,7564$ é dada por $T_2 = 0,110000011_2$. Para efeitos práticos, toma-se apenas a parte fracionária de T_2 , e a sequência \mathbf{x} passa a ser representada pelo código 110000011_2 .

No processo de decodificação, ilustrado pela Figura 2.12, inicialmente divide-se o intervalo unitário em três subintervalos, definidos segundo a Equação 2.12. Verifica-se, então, a que subintervalo o número representado pelo código binário pertence. No exemplo, $T_2 \in [0,00,8]$, indicando que o primeiro símbolo a ser decodificado é o a_1 . Na etapa seguinte, o intervalo $[0,00,8]$ é dividido em três outros subintervalos. Verifica-se agora que $T_2 \in [0,64,0,76]$. Desse modo, o próximo símbolo a ser decodificado é o a_2 . Prosseguindo com a decodificação, o intervalo $[0,64,0,76]$ também é subdividido em três outros. Nesse novo estado, $T_2 \in [0,754,0,76]$, ou seja, o símbolo decodificado é o a_3 . Finalmente, subdivide o intervalo $[0,754,0,76]$ em três novos subintervalos e observa-se que $T_2 \in [0,754,0,7588]$. Ou seja, a_1 é decodificado, gerando-se, no decodificador, a sequência original $\mathbf{x} = a_1 a_2 a_3 a_1$.

Com a codificação aritmética, o comprimento médio por símbolo, l_{ac} , é dado por,

$$H \leq l_{ac} \leq H + \frac{2}{m}, \quad (2.16)$$

onde m é o número de símbolos a serem representados por um único rótulo. Já na codificação de Huffman, que é um exemplo de técnica baseada no conceito de alfabeto estendido, o comprimento médio por símbolo, l_{hf} , é dado por,

$$H \leq l_{hf} \leq H + \frac{1}{m}. \quad (2.17)$$

Uma primeira análise pode indicar a superioridade da codificação de Huffman sobre a codificação aritmética, pois, para um dado valor de m , o limite superior de l_{hf} é sempre menor que o limite superior de l_{ac} . Porém, na codificação de Huffman é necessário gerar um dicionário com os códigos para todas as possíveis sequências de comprimento m . Assim, se o alfabeto fosse composto por k símbolos, então o dicionário seria composto por k^m códigos. Ou seja, para $k = 8$ e $m = 32$, por exemplo, o dicionário teria 8^{32} códigos. Isso torna essa abordagem impraticável do ponto de vista da complexidade computacional. Na codificação aritmética, por outro lado, gera-se apenas o código binário referente ao rótulo da sequência a ser codificada. Dessa maneira, salvo em casos particulares, a codificação aritmética passa a ser uma forma de se obter taxas de compressão mais próximas da entropia do que as obtidas por meio de algoritmos baseados em alfabeto estendido.

Mais detalhes sobre a codificação aritmética não serão tratados aqui. Sobre esse assunto a literatura é vasta. Inclusive, do ponto de vista da implementação, aquele que se interessar pode consultar o trabalho realizado por Sayood [31].

2.2.1.3 O Padrão JBIG

O *Joint Bi-level Image Processing Group* (JBIG) [39] é um grupo formado por especialistas do *International Organization for Standardization* (ISO) [40], do *International Electrotechnical Commission* (IEC) [41] e do *Consultative Committee for International Telephone and Telegraph* (CCITT). Os dois primeiros são grupos industriais, enquanto o CCITT é o antigo nome do atual *International Telecommunication Union - Telecommunication Standardization Sector* (ITU-T), uma organização intergovernamental responsável por coordenar padronizações relacionadas a telecomunicações [42]. Formado em 1988, o grupo teve por objetivo especificar um padrão para compressão de imagens binárias.

Publicado em 1993 [4], as principais vantagens do padrão JBIG sobre outros padrões de compressão de imagens binárias, como o *FAX Group 3* [43] e o *FAX Group 4* [44], são a codificação para *transmissão progressiva* e a superioridade em compressão [45, 46, 47, 31].

Na transmissão progressiva, uma representação da imagem em baixa resolução é enviada inicialmente e um refinamento posterior ocorre por meio da transmissão de informação adicional. A abordagem desse aspecto do JBIG não é relevante ao entendimento do presente trabalho. No entanto, o leitor que desejar aprofundar seu conhecimento no assunto pode encontrar mais informações em uma discussão realizada por Witten, Moffat e Bell [48].

A eficiência na compressão, por sua vez, deve-se principalmente à codificação aritmética adaptativa ao contexto, sugerida pelo padrão. Para entender como tal codificador é capaz de realizar a compressão de forma eficiente, considere o seguinte exemplo. Sejam $P(0)$ e $P(1)$ as probabilidades de ocorrência de *pixels* pretos (0) e brancos (1) em uma imagem binária, respectivamente. A entropia para essa fonte é dada por,

$$H = -P(0) \cdot \log_2 P(0) - P(1) \cdot \log_2 P(1).$$

Assim, se,

$$P(0) = 0,5 \text{ e } P(1) = 0,5,$$

então,

$$H = 1 \text{ bit}.$$

Ou seja, utilizando-se um único codificador aritmético, a taxa de *bits* média para se codificar a imagem em questão é de 1 *bit/pixel*. Nessa abordagem, a imagem está sendo entendida simplesmente como um fluxo de dígitos binários unidimensional.

Considere, agora, que as frequências de ocorrência de 0's e 1's estejam condicionadas a dois contextos, C_1 e C_2 , conforme ilustrado na Figura 2.13. Se, por exemplo, as probabilidades de

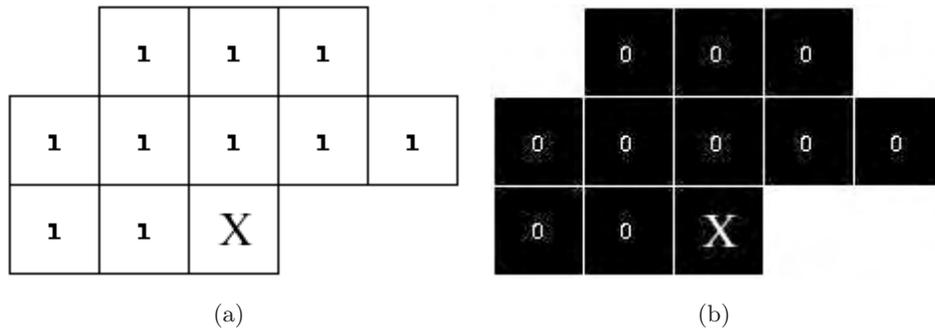


Figura 2.13: As frequências de ocorrência de 0's e 1's estão condicionadas a dois contextos, (a) C_1 e (b) C_2 . Nas figuras, X representa o *pixel* a ser codificado.

ocorrência de 0's e 1's no contexto C_1 forem,

$$P_1(0) = 0,2 \text{ e } P_1(1) = 0,8,$$

respectivamente, então a entropia é dada por,

$$H_1 = -P_1(0) \cdot \log_2 P_1(0) - P_1(1) \cdot \log_2 P_1(1) = 0,7219 \text{ bit}.$$

Por outro lado, se para o contexto C_2 as probabilidades de ocorrência de 0's e 1's forem,

$$P_2(0) = 0,1 \text{ e } P_2(1) = 0,9,$$

respectivamente, então a entropia é dada por,

$$H_2 = -P_2(0) \cdot \log_2 P_2(0) - P_2(1) \cdot \log_2 P_2(1) = 0,4690 \text{ bit}.$$

Agora, sejam as probabilidade de ocorrência dos contextos C_1 e C_2 ,

$$P(C_1) = 0,2 \text{ e } P(C_2) = 0,8,$$

respectivamente. Se forem utilizados dois codificadores aritméticos, um para cada contexto, então a taxa de *bits* média para se codificar a imagem seria,

$$H = P(C_1) \cdot H_1 + P(C_2) \cdot H_2 = 0,5196 \text{ bit},$$

ou seja, quase metade da taxa média resultante ao se utilizar um único codificador aritmético.

O JBIG utiliza de 1024 a 4096 codificadores aritméticos que se adaptam à imagem codificada, o que o torna extremamente eficiente. Se for considerada uma vizinhança composta por 10 *bits*, como a mostrada na Figura 2.13, torna-se possível indicar até 1024 codificadores aritméticos diferentes (de 000000000_2 a 111111111_2), cada qual tendo suas tabelas de probabilidades de ocorrência de 0's e 1's vinculadas a um contexto específico. Assim, à medida que a codificação ocorre, o contexto do *pixel* atual é analisado e o codificador vinculado a esse contexto é utilizado. Os probabilidades de ocorrência de 0's e 1's do codificador escolhido são, então, atualizadas e a codificação prossegue até que todos os *pixels* da imagem tenham sido codificados. A vantagem de se utilizar codificadores aritméticos em vez de codificadores baseados no conceito de alfabeto estendido está no fato de que com a codificação aritmética é possível gerar códigos para um grupo de m *bits*, por exemplo, sem que seja necessário gerar todos os possíveis códigos de comprimento m .

O codificador adotado pelo padrão é o IBM QM-coder, que é uma modificação do Q-coder [49]. O JBIG é coberto por 24 patentes, a maioria delas relacionadas ao QM-coder [46], porém uma implementação em ANSI C, para fins acadêmicos, foi desenvolvida por Kuhn [50] e pode ser obtida gratuitamente. Tal implementação foi utilizada nos testes aqui realizados.

2.2.1.4 O Padrão JBIG2

O JBIG foi o primeiro padrão publicado pelo *Joint Bi-level Image Processing Group*, em 1993. Como dito na subseção anterior, tal padrão pode ser entendido como a combinação de dois algoritmos: um para compressão de imagens binárias e outro para transmissão progressiva. Apesar de também poder realizar a compressão com perdas, a qualidade da imagem reconstruída é consideravelmente baixa. Assim, sua principal aplicação é em compressão de imagens binárias sem perdas.

Em 2000 o comitê JBIG publicou um novo padrão para compressão de imagens binárias que ficou conhecido como JBIG2 [5]. Tal padrão foi intencionalmente desenvolvido para realizar a compressão com perdas, sem perdas e com perdas para sem perdas (*lossy-to-lossless*) de imagens em preto e branco. O JBIG2 apresenta definições apenas para o decodificador, o que flexibiliza o projeto do codificador e possibilita constantes melhorias. Por esse motivo, os procedimentos de compressão são sempre observados a partir do ponto de vista da decodificação.

A abordagem proposta pelo JBIG2 consiste em segmentar a imagem a ser comprimida em três tipos de regiões - *regiões de símbolos*, *regiões de halftone* e *regiões genéricas* - e aplicar uma técnica de compressão adequada a cada uma delas [51, 52, 31]. Regiões de símbolos são aquelas que contêm elementos de texto. A decodificação dessas regiões é realizada a partir de técnicas de "casamento e substituição" [53, 54] que utilizam um dicionário de símbolos. Regiões de *halftone*, por sua vez, são aquelas compostas por imagens contendo padrões de *halftone*. Essas regiões podem ser decodificadas de duas maneiras: a partir de um dicionário de padrões de *halftone*, ou a partir do mesmo procedimento de decodificação aplicado às regiões genéricas. Regiões genéricas são classificadas como tal por não se enquadrarem em nenhum dos dois tipos de regiões anteriormente citados (de símbolos e de *halftone*). Tipicamente, são formadas por elementos gráficos (*line-art*). O procedimento para decodificação de regiões genéricas utiliza a codificação de Huffman, conforme especificada no padrão ITU-T T.6 (FAX Group 4), ou a codificação aritmética baseada no MQ-

coder [55], um codificador similar ao proposto pelo JBIG (QM-coder) [52]. O procedimento de decodificação de regiões genéricas também é utilizado na decodificação dos dicionários de símbolos e de padrões de *halftone*. A Figura 2.14 mostra trechos de três imagens binárias. Cada trecho ilustra um dos tipos de regiões anteriormente descritos.

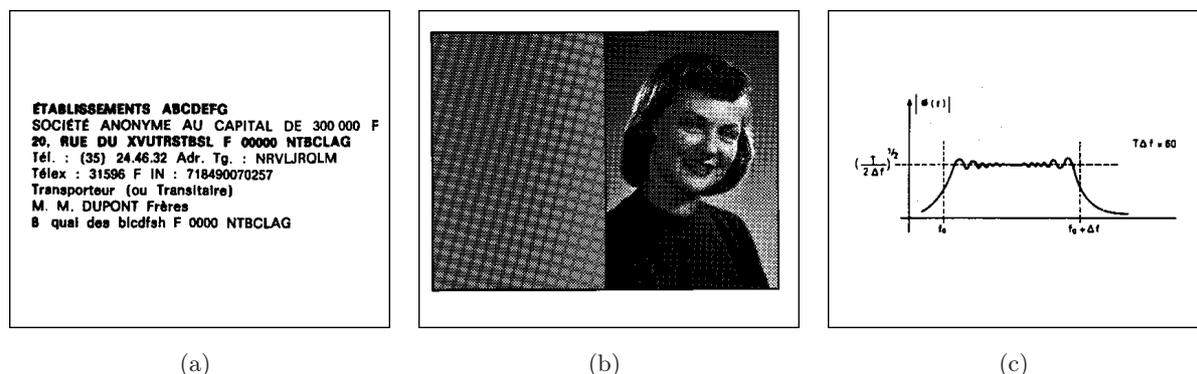


Figura 2.14: Exemplos de tipos de regiões: (a) região de símbolos; (b) região de *halftone*; e (c) região genérica.

É importante ressaltar, ainda, que a segmentação em regiões [56] não é descrita na parte normativa do JBIG2, pois é entendida como uma etapa de pré-processamento que não afeta a compatibilidade com o decodificador.

Como a compressão com perdas é permitida, os símbolos e os padrões de *halftone* contidos nos dicionários não precisam corresponder exatamente aos encontrados na imagem original. Um procedimento de decodificação para refinamento de regiões, porém, pode ser utilizado, transformando uma representação com perdas em uma representação sem perdas (*lossy-to-lossless*). Os dados adicionais gerados pelo processo de refinamento também são codificados com o MQ-coder.

As inovações apresentadas por esse novo padrão foram capazes de fazê-lo superar seu antecessor JBIG por um fator de 2 a 4 em compressão [57], estabelecendo o estado-da-arte em compressão de imagens binárias.

2.2.2 Padrões de Compressão de Imagens em Tons Contínuos

Em contraste com os padrões de compressão de imagens binárias, descritos na Seção 2.2.1, os padrões de compressão de imagens em tons contínuos aqui abordados baseiam-se em técnicas de codificação por transformada [31, 58, 3], com perdas. De uma forma bastante simplificada, a transformada leva os *pixels* da imagem, originalmente no domínio espacial, para um outro domínio no qual a maior parte da informação visual é concentrada em poucos elementos, também chamados de *coeficientes*. Os coeficientes de menor relevância são quantizados para zero, o que resulta em compressão. Tal compressão é dita com perdas, pois os elementos zerados não podem ser recuperados no decodificador, havendo perda de informação.

Devido às perdas resultantes do processo de quantização dos coeficientes da transformada, faz-se necessária a escolha de uma medida de distorção. Tais medidas podem ser divididas em duas categorias: *subjetiva* e *objetiva* [14]. A medida é dita subjetiva quando a qualidade da imagem

reconstruída é avaliada por um ser humano. Já na medida objetiva calcula-se a diferença entre a imagem original F_o e a imagem reconstruída F_r . Medidas largamente utilizadas na literatura são o erro médio quadrático (MSE ou, do inglês, *Mean Squared Error*), a soma das diferenças absolutas (SAD ou *Sum of Absolute Differences*), o erro médio absoluto (MAE ou *Mean Absolute Error*) e a razão sinal ruído de pico (PSNR ou *Peak Signal-to-Noise Ratio*),

$$MSE = \frac{1}{M \times N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (F_o(m, n) - F_r(m, n))^2 \quad (2.18)$$

$$SAD = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |F_o(m, n) - F_r(m, n)| \quad (2.19)$$

$$MAE = \frac{1}{M \times N} \times SAD \quad (2.20)$$

$$PSNR = 10 \log_{10} \frac{MAX^2}{MSE} \text{ dB.} \quad (2.21)$$

Aqui, MAX representa o máximo valor que um *pixel* pode assumir.

Infelizmente, essas medidas nem sempre coincidem com a avaliação subjetiva. No presente trabalho optou-se por utilizar a PSNR como medida de distorção por ser uma das mais utilizadas na literatura.

A seguir serão apresentados dois padrões de compressão de imagens em tons contínuos propostos pelo *Joint Photograph Experts Group* [39].

2.2.2.1 O Padrão JPEG

Criado em 1986, o *Joint Photograph Experts Group* (JPEG) é, da mesma forma que o JBIG, um grupo formado pelos órgãos ISO, IEC e ITU. Em 1992, o grupo publicou o primeiro padrão internacional para compressão de imagens em tons contínuos, popularmente conhecido como JPEG [6, 59], ilustrado na Figura 2.15.

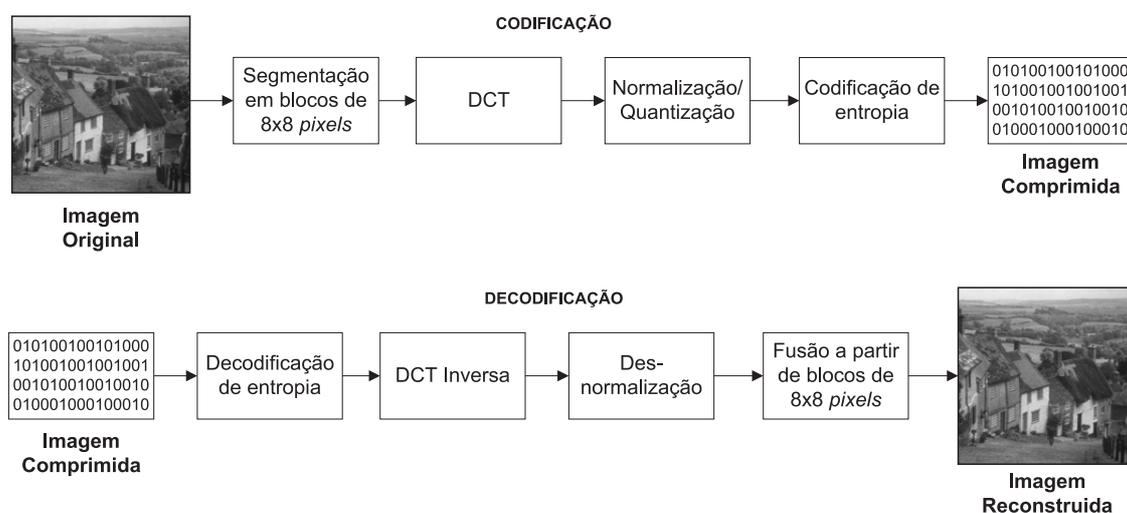


Figura 2.15: Diagrama de blocos do JPEG: codificação e decodificação.

Na etapa inicial da compressão JPEG, a imagem é dividida em blocos de 8×8 *pixels*, processados da esquerda para a direita, de cima para baixo. Do valor de intensidade de cada *pixel* são subtraídos 128 níveis de cinza e a *Transformada de Cosseno Discreta* (DCT ou *Discrete Cosine Transform*) [60] bidimensional é computada. Em seguida, são realizadas simultaneamente a normalização e a quantização escalar uniforme [61] dos coeficientes da transformada. É nessa etapa que, dependendo da quantização/normalização aplicada, uma quantidade maior ou menor de coeficientes é zerada, resultando em uma maior ou menor taxa de compressão. É importante ressaltar, porém, que quanto maior é a taxa de compressão, pior é a qualidade da imagem comprimida. Após a quantização/normalização, a matriz 8×8 de coeficientes é reordenada a partir da varredura *zig-zag* [59], conforme ilustrado na Figura 2.16, resultando em uma sequência unidimensional de 64 coeficientes. A etapa final da compressão consiste na codificação de entropia dos coeficientes, desenvolvida principalmente com o objetivo de se tirar vantagem das sequências de zeros que aparecem quando os coeficientes são reordenados segundo a varredura proposta. Na decodificação, o processo inverso é realizado.

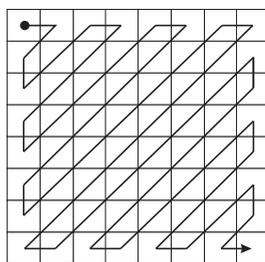


Figura 2.16: Varredura *zig-zag*: padrão para reordenação dos coeficientes da DCT.

Há uma implementação em *software* do JPEG, realizada pelo *Independent JPEG Group* (IJG) [62], disponível para ser utilizada gratuitamente. A Figura 2.17 mostra a imagem “baboon” original e a comprimida a diferentes taxas de *bits*.

Na verdade, o padrão JPEG especifica uma família de técnicas de compressão de imagens das quais podem-se escolher aquelas que atendam às exigências de uma determinada aplicação. Consiste em quatro modos de operação, cada um oferecendo múltiplas opções de configuração. Aqui foram tratados apenas seus aspectos gerais.

2.2.2.2 O Padrão JPEG2000

Uma grande mudança de paradigma em compressão de imagens ocorreu a partir do início da década de 1990, após a popularização da *Transformada Wavelet Discreta* (DWT ou *Discrete Wavelet Transform*) [63, 64, 65]. As propriedades dessa transformada incentivaram o estudo de uma nova classe de algoritmos de compressão de imagens [66, 67, 68] que resultou na especificação do padrão JPEG2000.

Desenvolvido a partir dos princípios da DWT, o JPEG2000 [7, 69] é o mais novo padrão de compressão proposto pelo *Joint Photograph Experts Group*. Suporta compressão com e sem perdas e atualmente representa o estado-da-arte em compressão de imagens. A Figura 2.18 ilustra o CODEC JPEG2000.

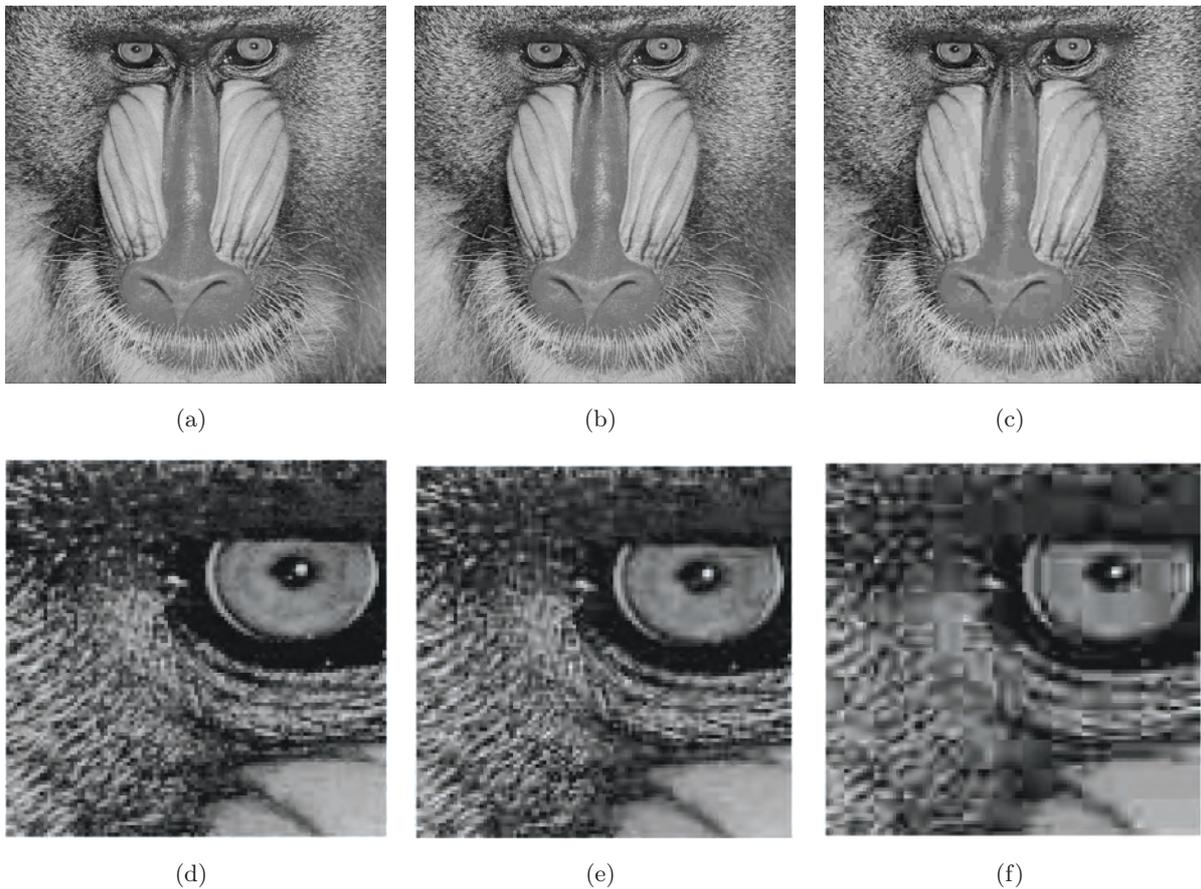


Figura 2.17: Imagem “baboon” codificada com o JPEG: (a) imagem original (8 *bits/pixel*); (b) imagem reconstruída com PSNR = 28,2310 dB a 1,3946 *bits/pixel*; e (c) imagem reconstruída com PSNR = 23,4256 dB a 0,4563 *bits/pixel*. As imagens (d), (e) e (f) são ampliações das imagens (a), (b) e (c), respectivamente.

O primeiro passo da compressão é a subtração de 128 níveis de cinza da intensidade de cada *pixel* da image original. Em seguida a DWT é computada. Para uma compressão sem perdas a transformada biortogonal 5/3 é utilizada. Em aplicações com perdas utiliza-se a *Cohen-Daubechies-Feauveau* (CDF) 9/7. Em ambos os casos, a transformação, também chamada de decomposição no caso da DWT, resulta em quatro sub-bandas: uma aproximação de baixa frequência e as características de frequência horizontal, vertical e diagonal da imagem, conforme ilustrado na Figura 2.19.

A decomposição pode prosseguir por meio de iterações subsequentes aplicadas aos coeficientes da aproximação de baixa-frequência. A Figura 2.20 ilustra a decomposição da imagem “Lena” em um e dois níveis. Para mais detalhes sobre *Wavelets*, o leitor pode consultar os trabalhos de Strang e Nguyen [65], e Daubechies [70].

Após a etapa de transformação, a maior parte da informação visual passa a concentrar-se em uma quantidade de coeficientes que é inferior à quantidade de *pixels* da imagem original. Os coeficientes são, então, quantizados por meio da *quantização escalar de faixa morta* (do inglês, *Dead Zone Quantization*) [61] e codificados entropicamente com o MQ-coder.

Na parte 5 do JPEG2000 dois *softwares* de referência [71] são disponibilizados (um em JAVA e

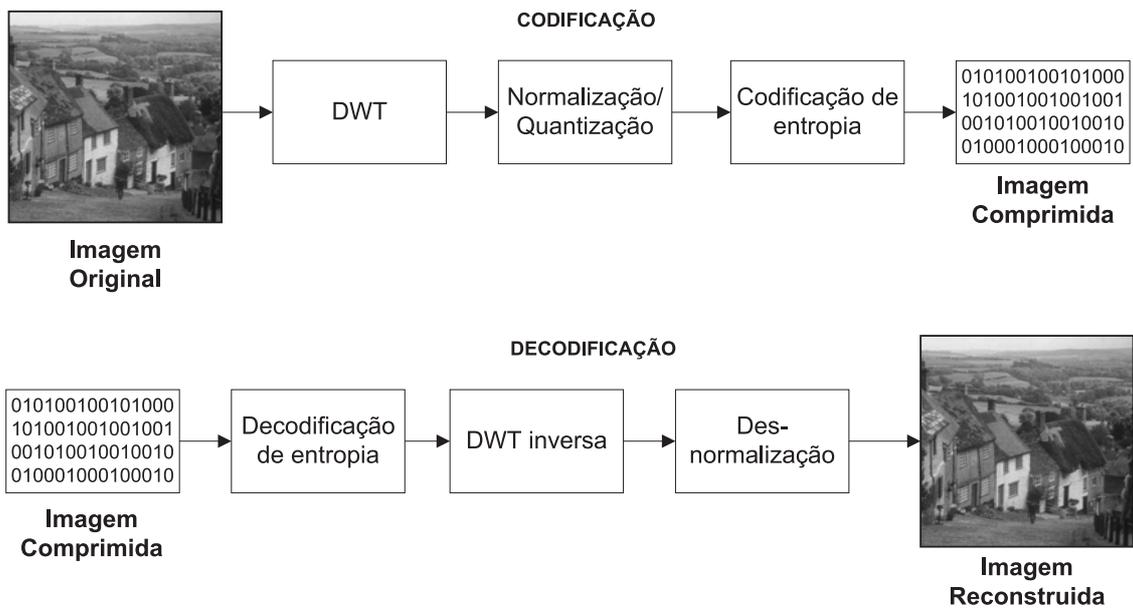


Figura 2.18: Diagrama de blocos do JPEG2000: codificação e decodificação.

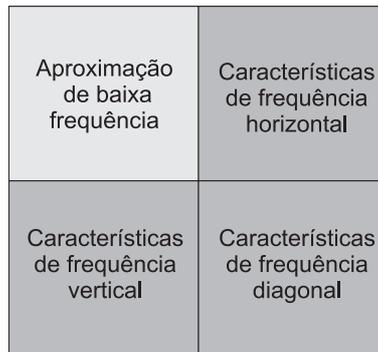


Figura 2.19: Sub-bandas da DWT.

outro em ANSI C) para que desenvolvedores possam realizar testes de validação de seus sistemas de codificação. A Figura 2.21 mostra a imagem “cameraman” comprimida com o *software* de referência a diferentes taxas.

A Figura 2.22 mostra, para efeito de comparação, a imagem “Barbara” comprimida com o JPEG e com JPEG2000 a aproximadamente $0,5 \text{ bit/pixel}$. Note a superioridade em termos objetivos (PSNR) e subjetivos (o *efeito de bloco*⁵ produzido pelo JPEG não é observado no JPEG2000). A Figura 2.23 (b) mostra curvas PSNR comparando objetivamente a codificação JPEG e JPEG2000 na compressão da imagem “peppers”, mostrada na Figura 2.23 (a).

Assim como no caso do JPEG, foram abordados apenas os aspectos gerais do JPEG2000. Para mais detalhes a respeito do padrão, o próprio *Joint Photograph Experts Group* sugere consultar o trabalho realizado por Taubman e Marcellin [69].

⁵Artefatos resultantes de métodos de codificação baseados na segmentação da imagem em blocos.

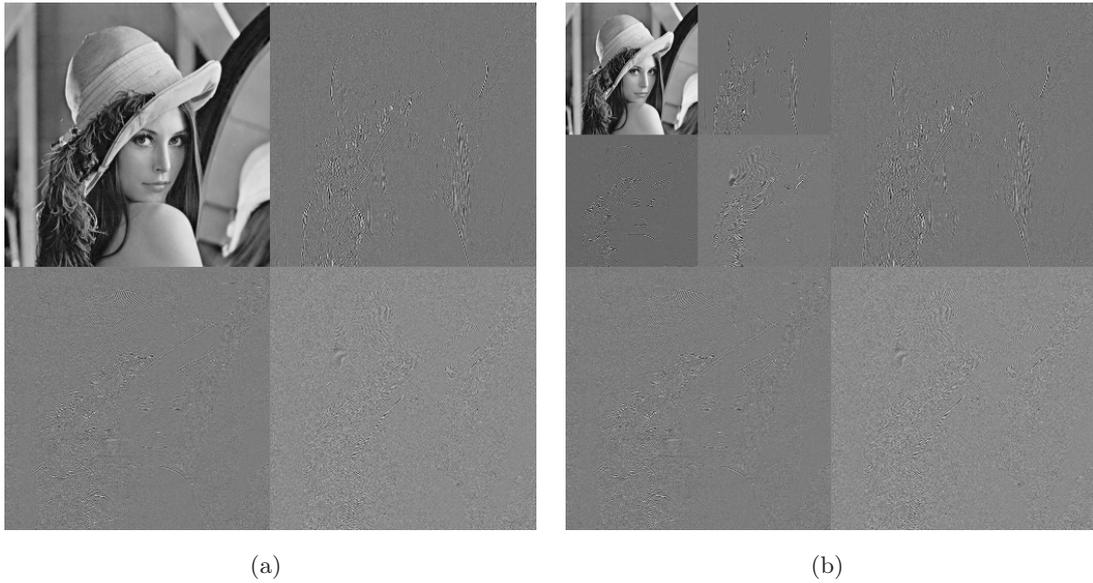


Figura 2.20: Decomposição da imagem “Lena” em (a) um e (b) dois níveis, utilizando a CDF 9/7.

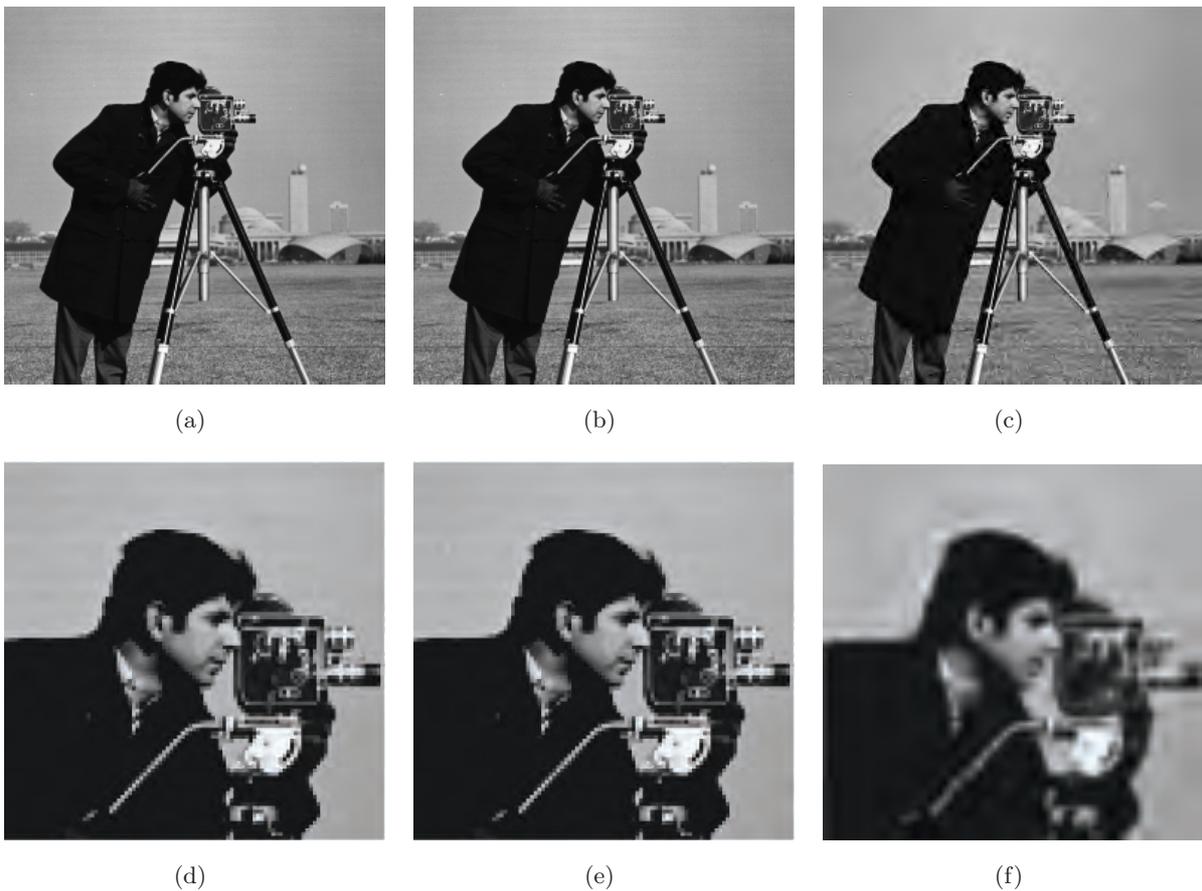


Figura 2.21: Imagem “cameraman” codificada com o JPEG2000: (a) imagem original (8 bits/pixel); (b) imagem reconstruída com $\text{PSNR} = 36,1248 \text{ dB}$ a $0,9857 \text{ bits/pixel}$; e (c) imagem reconstruída com $\text{PSNR} = 30,7664 \text{ dB}$ a $0,4929 \text{ bits/pixel}$. As imagens (d), (e) e (f) são ampliações das imagens (a), (b) e (c), respectivamente.



Figura 2.22: Comparação dos codificadores JPEG e JPEG2000: (a) imagem original (8 *bits/pixel*); (b) imagem reconstruída com JPEG (PSNR = 28,3434 dB a 0,5076 *bits/pixel*); e (c) imagem reconstruída com JPEG2000 (PSNR = 32,7589 dB a 0,4997 *bits/pixel*). As imagens (d), (e) e (f) são ampliações das imagens (a), (b) e (c), respectivamente.

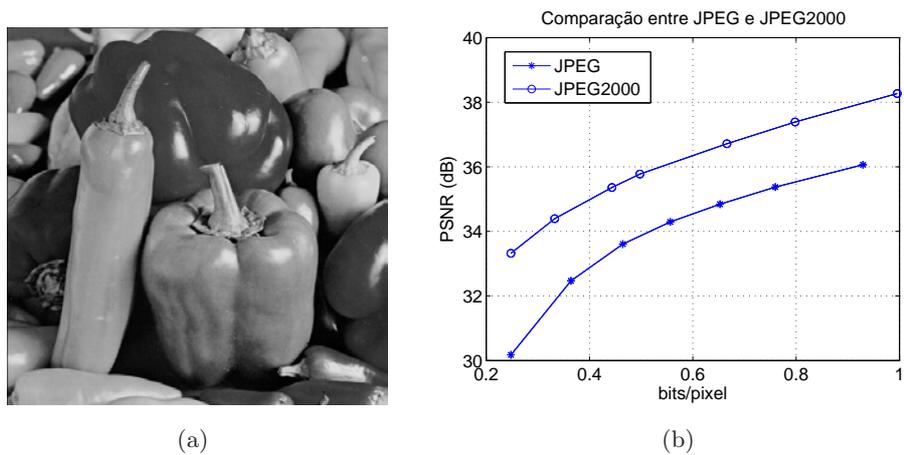


Figura 2.23: Comparação entre o JPEG e o JPEG2000: (a) imagem “peppers”; e (b) curvas PSNR para a imagem “peppers”.

Capítulo 3

O Modo INTRA do padrão H.264/AVC

3.1 Compressão de Vídeo

Um *vídeo natural* pode ser definido como uma sucessão temporal contínua de cenas [72]. O processo de digitalização de um vídeo natural consiste na captura de imagens digitais, conforme descrito no Capítulo 2, a intervalos de tempo regulares δt . Nesse contexto, tais imagens também são chamadas de *quadros (frames)* [27]. Após a amostragem temporal, o vídeo passa a ter valores definidos apenas para os instantes de tempo $\ell\delta t$, onde $\ell \in \mathbb{N}$. É muito comum ignorar o intervalo de captura δt e representar o vídeo digital como uma matriz tridimensional de *pixels*, $V(m, n, \ell)$, composta por M linhas, N colunas e L quadros, cada *pixel* podendo assumir um dos K possíveis níveis da escala de cinza, conforme ilustrado na Figura 3.1.

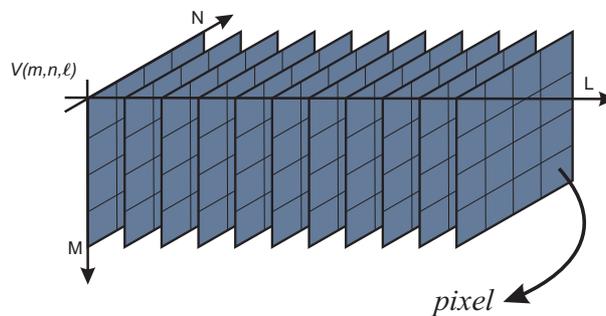


Figura 3.1: Vídeo digital representado como uma matriz tridimensional $V(m, n, \ell)$, composta por M linhas, N colunas e L quadros.

Suponha, agora, que durante duas horas um vídeo é capturado a uma taxa de 30 quadros por segundo e que cada quadro apresente $M = 288$ linhas e $N = 352$ colunas. Essa sequência seria composta por $L = 2 \times 3600 \times 30 = 216000$ quadros, cada quadro apresentando $288 \times 352 = 101376$ *pixels*. Se cada *pixel* fosse codificado com 8 *bits* (1 *byte*), o vídeo ocuparia aproximadamente 22 GB (1 GB = 10^9 *bytes*, no Sistema Internacional). Assim, torna-se evidente que a compressão é uma necessidade, tanto para efeito de armazenamento, quanto para efeito de transmissão.

Uma forma simplificada de se realizar a compressão de um vídeo digital é tratar o quadro como uma imagem independente e aplicar um codificador de imagens, como o JPEG2000, a cada quadro

da sequência. Essa forma de compressão também é chamada na literatura de compressão *intra-quadro* ou, simplesmente, compressão *intra*. Um exemplo de CODEC que utiliza tal abordagem é o *Motion JPEG2000* [73], que atualmente é o padrão de compressão adotado em cinema digital [74, 75, 76]. Um codificador mais elaborado, porém, pode, além de realizar a compressão *intra*, explorar as similaridades entre quadros vizinhos com o objetivo de reduzir a redundância temporal. Esse modo de compressão é denominado compressão *inter-quadros* ou, simplesmente, compressão *inter*. A Figura 3.2 ilustra um CODEC de vídeo composto por três unidades funcionais básicas: um *modelo temporal*, um *modelo espacial* e um *codificador de entropia*.

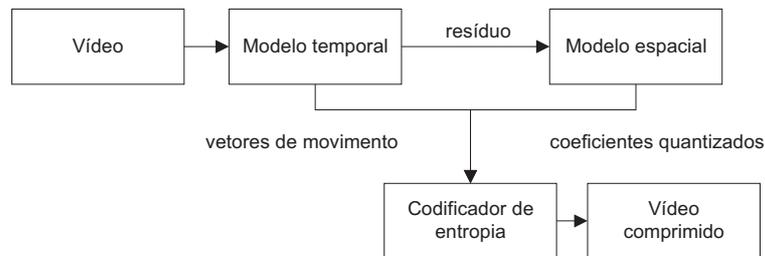


Figura 3.2: Diagrama de blocos para um CODEC de vídeo.

O modelo temporal explora o fato de quadros vizinhos apresentarem, com muita frequência, redundância entre si. Tipicamente, gera-se uma previsão do quadro atual a partir dos quadros vizinhos (*quadros de referência*). Tal previsão é denominada de *quadro predito*. Computa-se, então, a diferença entre o quadro predito e o *quadro atual* obtendo-se um *quadro residual*. Uma forma simplificada de se gerar um quadro residual é utilizar o quadro anterior como quadro predito. A Figura 3.3 mostra um exemplo de quadro residual obtido a partir de dois quadros sucessivos da sequência de vídeo “foreman”.

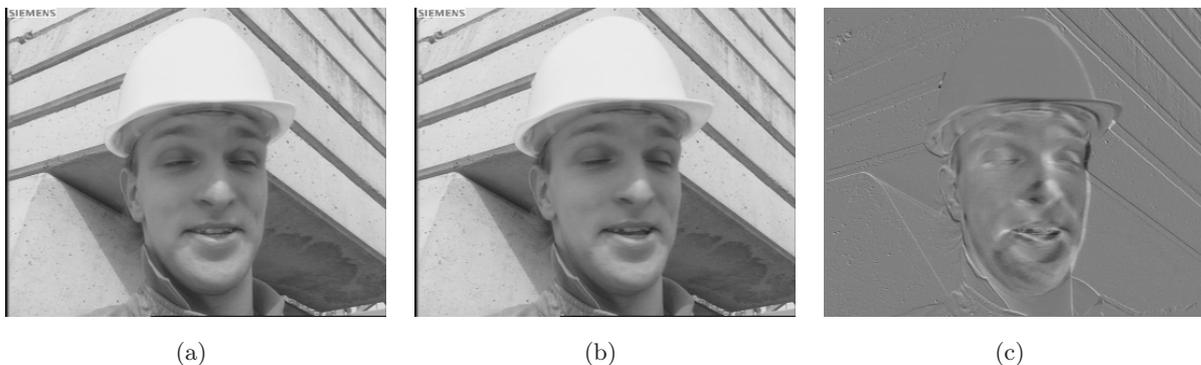


Figura 3.3: Modelo temporal: (a) quadro predito (anterior); (b) quadro atual; e (c) quadro residual.

Em modelos temporais mais complexos, o quadro atual é inicialmente dividido em blocos menores (de 16×16 *pixels*, por exemplo). Em seguida, buscam-se em quadros de referência, dentro de uma janela, blocos que melhor predizem os blocos do quadro atual, conforme ilustrado na Figura 3.4. Esse procedimento é chamado de *estimação de movimento* e tem como resultado *vetores de movimento*, que indicam os deslocamentos entre os blocos atuais e seus respectivos preditores. A partir dos quadros de referência e dos vetores de movimento, gera-se um quadro predito que é subtraído do quadro atual, realizando-se a *compensação de movimento* [72].

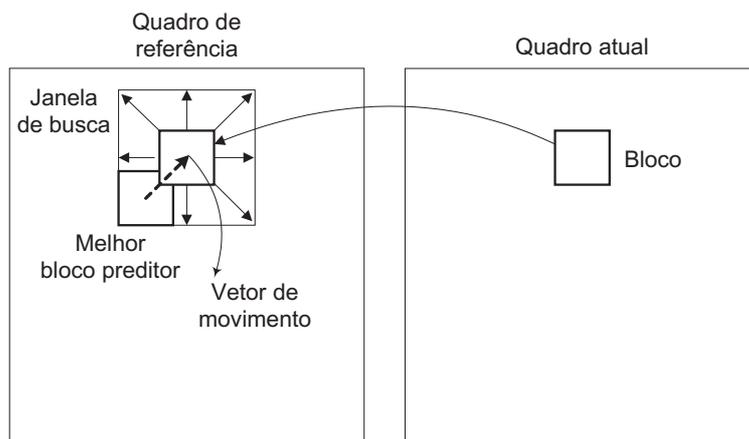


Figura 3.4: Estimação de movimento.

A Figura 3.5 mostra os vetores de movimento, o quadro predito e o quadro residual obtidos após a estimativa e a compensação de movimento realizadas a partir dos quadros das Figuras 3.3 (a) e (b). Na estimativa de movimento foram utilizados blocos de 16×16 *pixels* e uma janela de busca de 32×32 *pixels*. É interessante observar que o resíduo nessas condições é bem menos pronunciado que o obtido no caso ilustrado pela Figura 3.3 (c).

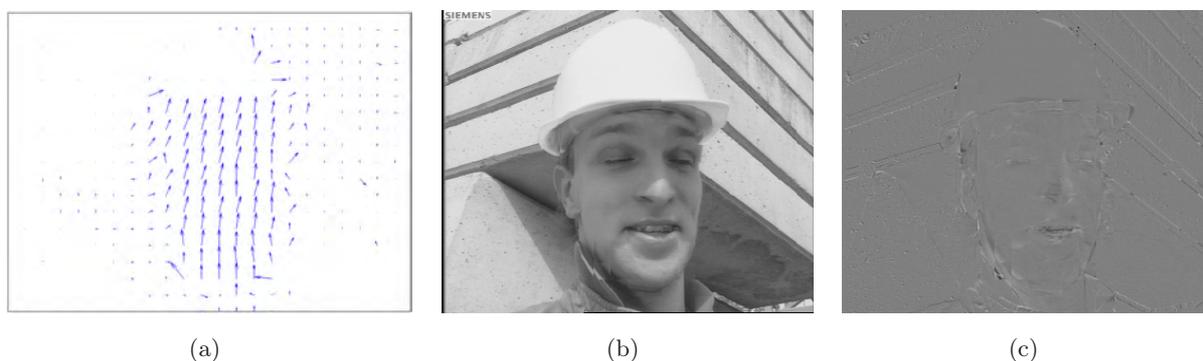


Figura 3.5: Estimativa e compensação de movimento: (a) vetores de movimento; (b) quadro predito (por compensação de movimento); e (c) quadro residual. Foram utilizados blocos de 16×16 e janela de busca de 32×32 *pixels*.

Por fim, o quadro residual é apresentado ao modelo espacial, responsável por reduzir as redundâncias entre amostras residuais vizinhas. Uma das maneiras de se alcançar tal objetivo é realizar uma etapa de transformação seguida de quantização. Os coeficientes quantizados são, então, codificados entropicamente, junto com os vetores de movimento.

Há mais detalhes que envolvem o projeto de um CODEC de vídeo, principalmente no que diz respeito à construção do modelo temporal. Aqui a discussão será limitada ao exposto até o momento, pois tais noções já são suficientes para o entendimento do que se segue.

3.2 O Padrão H.264/AVC

O H.264/AVC é o mais novo padrão para compressão de vídeo. Foi publicado em março de 2003 pelo *Joint Video Team* (JVT) [77], um grupo formado pelo ISO/IEC *Joint Technical Committee 1, Subcommittee 29, Working Group 11 (JTC1/SC29/WG11)* ou *Motion Picture Experts Group* (MPEG) [78] e pelo ITU *Study Group 16, Question 6 (SG16 Q.6), Working Party 3* ou *Video Coding Experts Group* (VCEG) [79]. O padrão foi publicado sob a denominação *H.264* [9] pela ITU-T e *MPEG-4 Part 10* [80] pela ISO/IEC. Também é chamado de *Advanced Video Coding* (AVC).

O padrão já foi amplamente discutido na literatura [72, 81, 82, 83, 84, 85]. Muitas publicações ilustram o desempenho do H.264/AVC, comparando-o a outros codificadores de vídeo como o *MPEG-2* [86] e o *H.263* [87]. Todos os resultados apontam para uma redução média da taxa de *bits* pela metade, dado um mesmo valor de distorção. O que o torna tão eficiente são pequenas melhorias que individualmente contribuem um pouco para o aumento substancial do ganho geral. Dentre as novidades apresentadas pelo H.264/AVC, pode-se citar [88]:

- Estimativa de movimento em *quarto-de-pixel* (*quarter-pel*): os vetores de movimento podem ter 4×4 -vezes mais precisão.
- Quadros de referência arbitrários: teoricamente, qualquer quadro da sequência pode ser utilizado como referência na predição inter.
- Filtragem para redução de efeito de bloco (*deblocking*): o efeito de bloco presente nos quadros de referência pode ser reduzido antes desses quadros serem utilizados na predição do quadro atual. Isso evita que artefatos de bloco passados degradem a predição do quadro atual.
- Codificação aritmética binária adaptativa baseada em contexto: possibilita uma melhor codificação de entropia.
- Predição intra-quadro: modos de predição intra podem ser selecionados a partir de uma análise de taxa e distorção, tanto na codificação intra, como para competir com a codificação inter.
- Partição com tamanho de bloco variável: a estimativa de movimento pode ser executada com tamanhos de bloco diferentes. O codificador pode quebrar um *macrobloco* (blocos de 16×16 *pixels*) em 2 ou 4 submacroblocos, os quais podem ser quebrados em 2 ou 4 sub-blocos cada, conforme ilustrado pela Figura 3.6. Dessa forma, os submacroblocos e sub-blocos podem melhor casar com as fronteiras de objetos presentes no quadro.
- Transformada com tamanho de bloco variável: os macroblocos podem ser transformados por meio de uma combinação de transformadas 8×8 e 4×4 . Emprega-se a combinação de melhor desempenho taxa-distorção. A transformada utilizada no H.264/AVC é uma aproximação de baixa complexidade da DCT [72].

As Figuras 3.7 e 3.8 ilustram o codificador e o decodificador H.264.

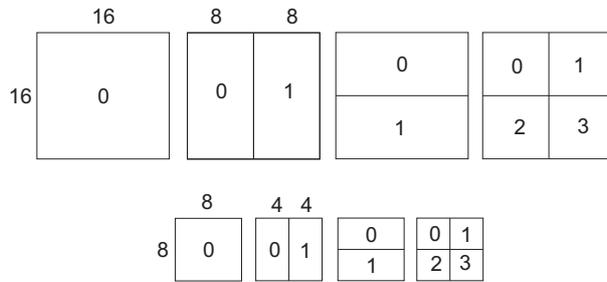


Figura 3.6: Partição com tamanho de bloco variável. Os números ao redor dos blocos representam suas dimensões em *pixels*.

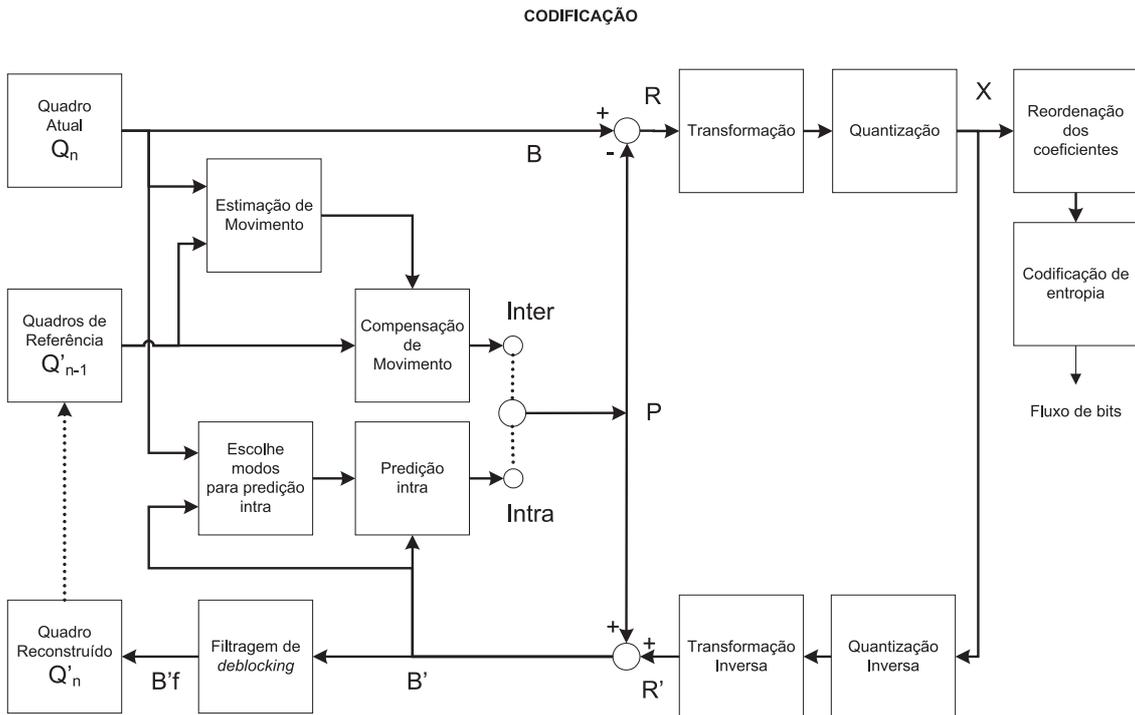


Figura 3.7: Codificação H.264/AVC.

Na codificação, o quadro atual Q_n é inicialmente dividido em macroblocos. Cada macrobloco é codificado nos modos intra ou inter, e para cada sub-bloco B dentro de um macrobloco gera-se um preditor P com base nas amostras previamente reconstruídas. No modo intra, P é formado a partir de amostras reconstruídas, mas não filtradas B' . No modo inter, P é formado por estimação e compensação de movimento a partir de quadros de referência anteriormente reconstruídos e filtrados Q'_{n-1} . Após a etapa de predição, P é subtraído do sub-bloco atual B , gerando um sub-bloco residual R , que, então, é submetido a uma etapa de transformação. Os coeficientes da transformada são quantizados, reordenados e codificados entropicamente junto com a informação lateral (vetores de movimentos, modos de predição etc.), resultando no fluxo de *bits*. Para cada sub-bloco B o codificador gera localmente uma versão reconstruída B' , que fica disponível como referência para predições futuras. A reconstrução tem início com a quantização inversa de X e a transformação inversa dos coeficientes da transformada, resultando no sub-bloco residual reconstruído R' . Esse sub-bloco é somado a P , dando origem a B' , que é a versão reconstruída do sub-bloco atual. Essa

versão reconstruída pode ser utilizada como referência na predição intra dos macroblocos que fazem parte do mesmo *slice* [72] no qual B' está inserido. Um filtro de *deblocking* é aplicado a B' e um novo quadro de referência Q'_n é construído a partir de uma série de sub-blocos filtrados B'_f . Tal quadro fica disponível para ser utilizado como referência na predição inter.

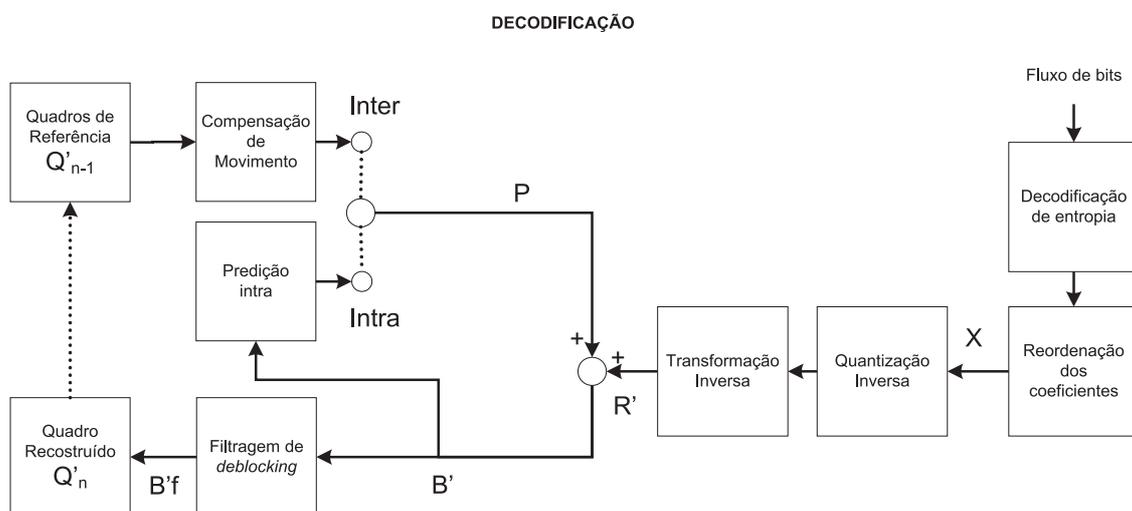


Figura 3.8: Decodificação H.264/AVC.

Na decodificação, o fluxo de *bits* é inicialmente decodificado entropicamente. Após a reordenação, os coeficientes X passam por um processo de quantização inversa, seguida de transformação inversa, dando origem a R' . A partir da informação lateral também decodificada a partir do fluxo de *bits*, P é recuperado. Somando-se P com R' obtém-se B' , que filtrado gera o sub-bloco reconstruído B'_f .

O código de referência do H.264/AVC, denominado JM [89], é disponibilizado pelo *Heinrich-Hertz-Institut* [90]. Todos os testes aqui realizados foram baseados na versão 10.2 do JM.

Na próxima seção, o modo de predição intra do H.264/AVC será estudado em detalhes.

3.3 A Predição Intra do H.264/AVC

Os elementos comuns à maioria dos padrões de codificação de vídeo estão presentes no H.264/AVC:

- Os quadros são processados em blocos;
- Cada bloco é composto por sinais de luminância e croma;
- Estimativa e compensação de movimento;
- Transformada em bloco;
- Quantização escalar; e
- Codificação de entropia.

Uma das novidades que contribuem para o alto desempenho do H.264/AVC é a *predição intra*, que oferece nove modos para blocos de luminância com tamanho 4×4 *pixels* (aqui denotados por *I4*), nove modos para blocos de luminância com tamanho 8×8 *pixels* (aqui denotados por *I8*), quatro modos para blocos de luminância com tamanho 16×16 *pixels* (aqui denotados por *I16*) e quatro modos para blocos de crominância com tamanho 8×8 *pixels*. Vale lembrar que no presente trabalho estão sendo consideradas apenas imagens em escala de cinza. Por esse motivo, o processamento da componente de crominância não será abordado. De maneira geral, a predição intra reduz a redundância espacial aproveitando-se da correlação espacial existente entre macroblocos adjacentes.

3.3.1 Modos de Predição *I4*

Os nove modos de predição para blocos de luminância 4×4 *pixels* são mostrados na Figura 3.9. A predição $pred4x4_L(m, n)$ das amostras a até p do bloco atual é realizada a partir das amostras $p(m, n)$ (A até M) dos blocos vizinhos. As amostras a e M correspondem à $pred4x4_L(0, 0)$ e $p(-1, -1)$, respectivamente. Existem no total oito “direções de predição” e um modo DC, conforme detalhado mais adiante [9]. Nas equações, o operador $round(\cdot)$ retorna o número inteiro mais próximo do argumento e o operador $\lfloor \cdot \rfloor$ retorna o maior inteiro menor que o argumento. Como exemplo, a Figura 3.10 mostra um bloco 4×4 a ser predito e suas respectivas predições *I4*.

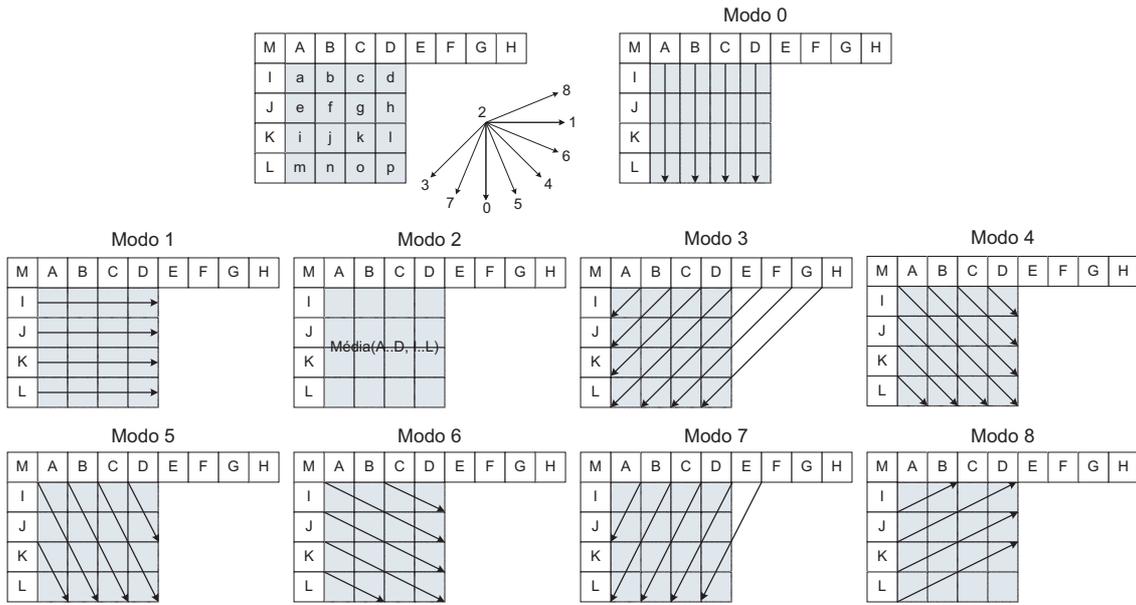


Figura 3.9: Predição intra para blocos de luminância com tamanho 4×4 pixels. Oito “direções de predição” e modo DC.

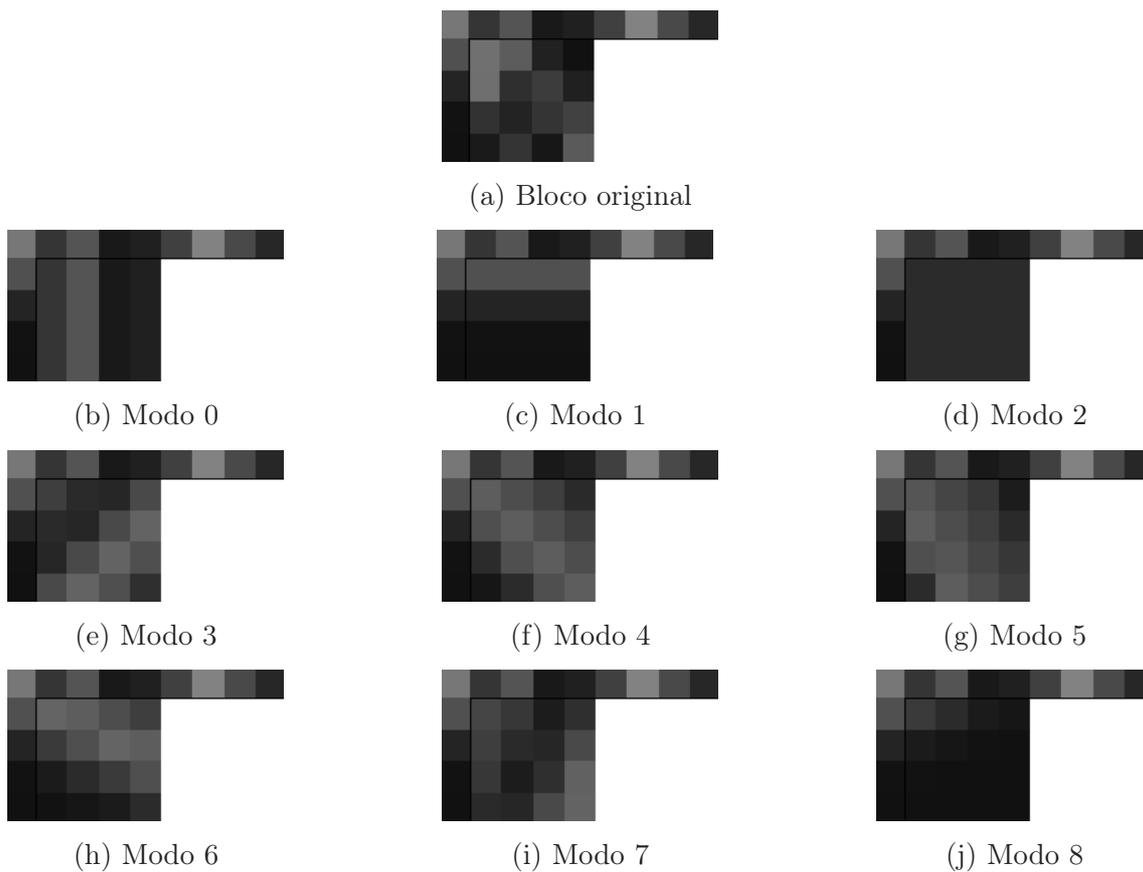


Figura 3.10: Exemplo de predições $I4$: (a) bloco original a ser predito; (b) a (j) predições geradas a partir dos modos de predição 0 a 8, respectivamente.

- *Modo 0* (Vertical): esse modo deve ser considerado apenas se as amostras $p(-1, n)$, com $n = 0..3$ (A até D), estão disponíveis.

Para $m, n = 0..3$,

$$pred4x4_L(m, n) = p(-1, n).$$

- *Modo 1* (Horizontal): esse modo deve ser considerado apenas se as amostras $p(m, -1)$, com $m = 0..3$ (I até L), estão disponíveis.

Para $m, n = 0..3$,

$$pred4x4_L(m, n) = p(m, -1).$$

- *Modo 2* (DC): esse modo é sempre considerado.

Para $m, n = 0..3$,

Se as amostras, $p(-1, n)$, com $n = 0..3$ (A até D), e $p(m, -1)$, com $m = 0..3$ (I até L), estão disponíveis, então,

$$pred4x4_L(m, n) = round \left(\frac{\sum_{j=0}^3 p(-1, j) + \sum_{i=0}^3 p(i, -1)}{8} \right),$$

senão, se alguma amostra $p(-1, n)$, com $n = 0..3$ (A até D), não está disponível e as amostras $p(m, -1)$, com $m = 0..3$ (I até L), estão disponíveis, então,

$$pred4x4_L(m, n) = round \left(\frac{\sum_{i=0}^3 p(i, -1)}{4} \right),$$

senão, se alguma amostra $p(m, -1)$, com $m = 0..3$ (I até L), não está disponível e as amostras $p(-1, n)$, com $n = 0..3$ (A até D), estão disponíveis, então,

$$pred4x4_L(m, n) = round \left(\frac{\sum_{j=0}^3 p(-1, j)}{4} \right),$$

senão, se alguma amostra $p(-1, n)$, com $n = 0..3$ (A até D), não está disponível e alguma amostra $p(m, -1)$, com $m = 0..3$ (I até L), também não está disponível, então,

$$pred4x4_L(m, n) = 128.$$

- *Modo 3* (Diagonal esquerda abaixo): esse modo deve ser considerado apenas se as amostras $p(-1, n)$, com $n = 0..7$ (A até H), estão disponíveis.

Para $m, n = 0..3$,

se $m = 3$ e $n = 3$ então,

$$pred4x4_L(m, n) = round\left(\frac{p(-1, 6) + 3p(-1, 7)}{4}\right),$$

senão,

$$pred4x4_L(m, n) = round\left(\frac{p(-1, m+n) + 2p(-1, m+n+1) + p(-1, m+n+2)}{4}\right).$$

- *Modo 4* (Diagonal direita abaixo): esse modo deve ser considerado apenas se as amostras $p(-1, n)$, com $n = 0..3$ (A até D), e $p(m, -1)$, com $m = -1..3$ (M e I até L), estão disponíveis.

Para $m, n = 0..3$,

se $n > m$ então,

$$pred4x4_L(m, n) = round\left(\frac{p(-1, n-m-2) + 2p(-1, n-m-1) + p(-1, n-m)}{4}\right),$$

senão, se $n < m$ então,

$$pred4x4_L(m, n) = round\left(\frac{p(m-n-2, -1) + 2p(m-n-1, -1) + p(m-n, -1)}{4}\right),$$

senão, se $n = m$ então,

$$pred4x4_L(m, n) = round\left(\frac{p(-1, 0) + 2p(-1, -1) + p(0, -1)}{4}\right).$$

- *Modo 5* (Vertical à direita): esse modo deve ser considerado apenas se as amostras $p(-1, n)$, com $n = 0..3$ (A até D), e $p(m, -1)$, com $m = -1..3$ (M e I até L), estão disponíveis.

Para $m, n = 0..3$,

seja $zVR = 2n - m$,

se $zVR = 0, 2, 4$ ou 6 então,

$$pred4x4_L(m, n) = round\left(\frac{p(-1, n - \lfloor m/2 \rfloor - 1) + p(-1, n - \lfloor m/2 \rfloor)}{2}\right),$$

senão, se $zVR = 1, 3$, ou 5 então,

$$pred4x4_L(m, n) = round\left(\frac{p(-1, n - \lfloor m/2 \rfloor - 2) + 2p(-1, n - \lfloor m/2 \rfloor - 1) + p(-1, n - \lfloor m/2 \rfloor)}{4}\right),$$

senão, se $zVR = -1$ então,

$$pred4x4_L(m, n) = round\left(\frac{p(0, -1) + 2p(-1, -1) + p(-1, 0)}{4}\right),$$

senão, se $zVR = -2$ ou -3 então,

$$pred4x4_L(m, n) = round\left(\frac{p(m-1, -1) + 2p(m-2, -1) + p(m-3, -1)}{4}\right).$$

- *Modo 6* (Horizontal abaixo): esse modo deve ser considerado apenas se as amostras $p(-1, n)$, com $n = 0..3$ (A até D), e $p(m, -1)$, com $m = -1..3$ (M e I até L), estão disponíveis.

Para $m, n = 0..3$,

seja $zHD = 2m - n$,

se $zHD = 0, 2, 4$ ou 6 então,

$$pred4x4_L(m, n) = round\left(\frac{p(m - \lfloor n/2 \rfloor - 1, -1) + p(m - \lfloor n/2 \rfloor, -1)}{2}\right),$$

senão, se $zHD = 1, 3, 5$ então,

$$pred4x4_L(m, n) = round\left(\frac{p(m - \lfloor n/2 \rfloor - 2, -1) + 2p(m - \lfloor n/2 \rfloor - 1, -1) + p(m - \lfloor n/2 \rfloor, -1)}{4}\right),$$

senão, se $zHD = -1$ então,

$$pred4x4_L(m, n) = round\left(\frac{p(0, -1) + 2p(-1, -1) + p(-1, 0)}{4}\right),$$

senão, se $zHD = -2$ ou -3 então,

$$pred4x4_L(m, n) = round\left(\frac{p(-1, n - 1) + 2p(-1, n - 2) + p(-1, n - 3)}{4}\right).$$

- *Modo 7* (Vertical à esquerda): esse modo deve ser considerado apenas se as amostras $p(-1, n)$, com $n = 0..7$ (A até H), estão disponíveis.

Para $m, n = 0..3$,

se $m = 0$ ou 2 então,

$$pred4x4_L(m, n) = round\left(\frac{p(-1, n + \lfloor m/2 \rfloor) + p(-1, n + \lfloor m/2 \rfloor + 1)}{2}\right),$$

senão, se $m = 1$ ou 3 então,

$$pred4x4_L(m, n) = round\left(\frac{p(-1, n + \lfloor m/2 \rfloor) + 2p(-1, n + \lfloor m/2 \rfloor + 1) + p(-1, n + \lfloor m/2 \rfloor + 2)}{4}\right).$$

- *Modo 8* (Horizontal acima): esse modo deve ser considerado apenas se as amostras $p(m, -1)$, com $m = 0..3$ (I até L), estão disponíveis.

Para $m, n = 0..3$,

seja $zHU = n + 2m$,

se $zHU = 0, 2, 4$ então,

$$pred4x4_L(m, n) = round\left(\frac{p(m + \lfloor n/2 \rfloor, -1) + p(m + \lfloor n/2 \rfloor + 1, -1)}{2}\right),$$

senão, se $zHU = 1$, ou 3 então,

$$pred4x4_L(m, n) = \text{round} \left(\frac{p(m + \lfloor n/2 \rfloor, -1) + 2p(m + \lfloor n/2 \rfloor + 1, -1) + p(m + \lfloor n/2 \rfloor + 2, -1)}{4} \right),$$

senão, se $zHU = 5$ então,

$$pred4x4_L(m, n) = \text{round} \left(\frac{p(2, -1) + 3p(3, -1)}{4} \right),$$

senão, se $zHU > 5$ então,

$$pred4x4_L(m, n) = p(3, -1).$$

Os modos de predição $I4$ são apropriados à codificação de regiões com significativo nível de detalhe. Algum tempo após a publicação do H.264/AVC, o perfil *Fidelity Range Extensions* (FRExt) [84] foi incorporado ao padrão. Nesse perfil, a predição do sinal com tamanho de bloco 8×8 pixels ($I8$) foi introduzida com o objetivo de se melhorar a eficiência em taxa e distorção do codificador.

3.3.2 Modos de Predição $I8$

Para a predição 8×8 há igualmente 9 modos, à semelhança da predição $I4$, o que acaba aumentando a complexidade computacional do codificador [91]. As amostras $pred8x8_L(m, n)$ do bloco atual são preditas a partir das amostras filtradas $p'(m, n)$ (A' até Z'), conforme ilustrado na Figura 3.11. A predição $I8$ é precedida por uma etapa de filtragem das amostras de referência $p(m, n)$ (A até Z), conforme descrito mais adiante. Como exemplo, a Figura 3.12 mostra um bloco 8×8 a ser predito e suas respectivas predições $I8$.

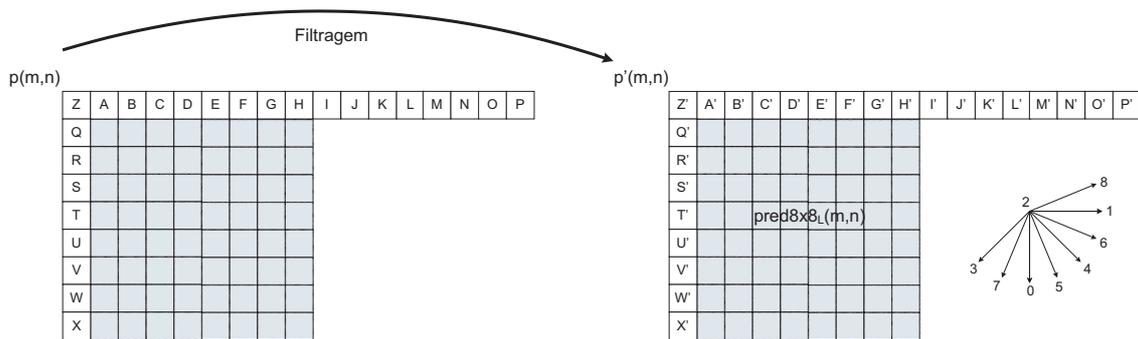


Figura 3.11: Predição intra para blocos de luminância com tamanho 8×8 pixels. Oito “direções de predição” e modo DC.

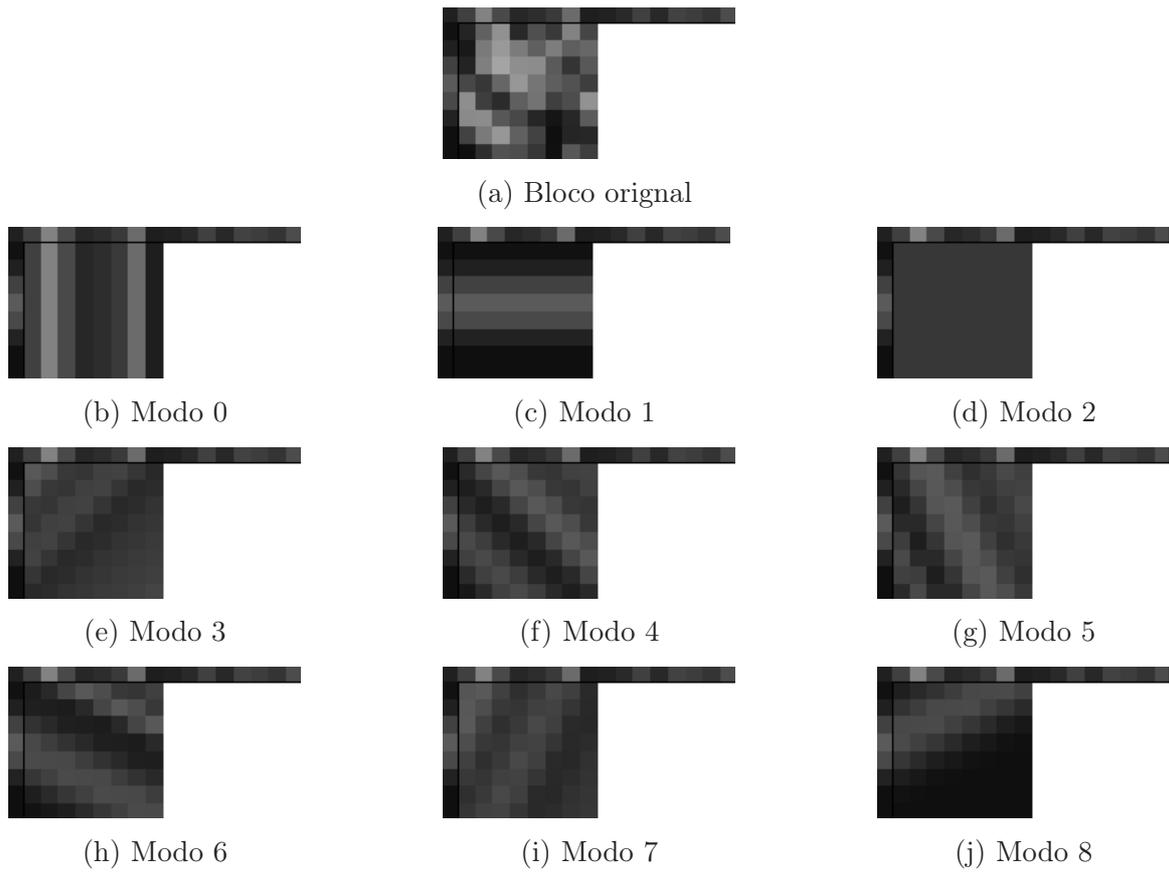


Figura 3.12: Exemplo de predições *I8*: (a) bloco original a ser predito; (b) a (j) predições geradas a partir dos modos de predição 0 a 8, respectivamente.

- Filtragem das amostras de referência $p(m, n)$

1. Se as amostras $p(-1, n)$, com $n = 0..7$ (A até H), estão disponíveis, então,

– Se $p(-1, -1)$ está disponível, então

$$p'(-1, 0) = \text{round} \left(\frac{p(-1, -1) + 2p(-1, 0) + p(-1, 1)}{4} \right),$$

senão, se $p(-1, -1)$ não está disponível, então

$$p'(-1, 0) = \text{round} \left(\frac{3p(-1, 0) + p(-1, 1)}{4} \right).$$

– Para $n = 1..7$,

$$p'(-1, n) = \text{round} \left(\frac{p(-1, n-1) + 2p(-1, n) + p(-1, n+1)}{4} \right).$$

2. Se as amostras $p(-1, n)$, com $n = 7..15$ (H até P), estão disponíveis, então,

– Para $n = 8..14$,

$$p'(-1, n) = \text{round} \left(\frac{p(-1, n-1) + 2p(-1, n) + p(-1, n+1)}{4} \right).$$

– Para $n = 15$,

$$p'(-1, n) = \text{round} \left(\frac{p(-1, n-1) + 3p(-1, n)}{4} \right).$$

3. Quando a amostra $p(-1, -1)$ estiver disponível, então,

– Se apenas um entre $p(-1, 0)$ e $p(0, -1)$ não está disponível, então, se $p(-1, 0)$ está disponível e $p(0, -1)$ não está disponível, então,

$$p'(-1, -1) = \text{round} \left(\frac{3p(-1, -1) + p(-1, 0)}{4} \right),$$

senão, se $p(0, -1)$ está disponível e $p(-1, 0)$ não está disponível,

$$p'(-1, -1) = \text{round} \left(\frac{3p(-1, -1) + p(0, -1)}{4} \right).$$

– Se $p(-1, 0)$ e $p(0, -1)$ estão disponíveis, então,

$$p'(-1, -1) = \text{round} \left(\frac{p(-1, 0) + 2p(-1, -1) + p(0, -1)}{4} \right).$$

4. Se as amostras $p(m, -1)$, com $n = 0..7$ (Q até X), estão disponíveis, então,

– Se $p(-1, -1)$ está disponível, então

$$p'(0, -1) = \text{round} \left(\frac{p(-1, -1) + 2p(0, -1) + p(1, -1)}{4} \right),$$

senão, se $p(-1, -1)$ não está disponível, então

$$p'(0, -1) = \text{round} \left(\frac{3p(0, -1) + p(1, -1)}{4} \right).$$

– Para $m = 1..6$,

$$p'(m, -1) = \text{round} \left(\frac{p(m-1, -1) + 2p(m, -1) + p(m+1, -1)}{4} \right).$$

– Para $m = 7$,

$$p'(m, -1) = \text{round} \left(\frac{p(m-1, -1) + 3p(m, -1)}{4} \right).$$

A seguir serão descritos os nove modos de predição I8.

- *Modo 0* (Vertical): esse modo deve ser considerado apenas se as amostras $p'(-1, n)$, com $n = 0..7$ (A' até H'), estão disponíveis.

Para $m, n = 0..7$,

$$\text{pred8x8}_L(m, n) = p'(-1, n).$$

- *Modo 1* (Horizontal): esse modo deve ser considerado apenas se as amostras $p'(m, -1)$, com $m = 0..7$ (Q' até X'), estão disponíveis.

Para $m, n = 0..7$,

$$\text{pred8x8}_L(m, n) = p'(m, -1).$$

- *Modo 2* (DC): esse modo é sempre considerado.

Para $m, n = 0..7$,

Se as amostras, $p'(-1, n)$, com $n = 0..7$ (A' até H'), e $p'(m, -1)$, com $m = 0..7$ (Q' até X'), estão disponíveis, então,

$$\text{pred8x8}_L(m, n) = \text{round} \left(\frac{\sum_{j=0}^7 p'(-1, j) + \sum_{i=0}^7 p'(i, -1)}{16} \right).$$

senão, se alguma amostra $p'(-1, n)$, com $n = 0..7$ (A' até H'), não está disponível e as amostras $p'(m, -1)$, com $m = 0..7$ (Q' até X'), estão disponíveis, então,

$$\text{pred8x8}_L(m, n) = \text{round} \left(\frac{\sum_{i=0}^7 p'(i, -1)}{8} \right),$$

senão, se alguma amostra $p'(m, -1)$, com $m = 0..7$ (Q' até X'), não está disponível e as amostras $p'(-1, n)$, com $n = 0..7$ (A' até H'), estão disponíveis, então,

$$\text{pred8x8}_L(m, n) = \text{round} \left(\frac{\sum_{j=0}^7 p'(-1, j)}{8} \right),$$

senão, se alguma amostra $p'(-1, n)$, com $n = 0..7$ (A' até H'), não está disponível e alguma amostra $p'(m, -1)$, com $m = 0..3$ (Q' até X'), também não está disponível, então,

$$pred8x8_L(m, n) = 128.$$

- *Modo 3* (Diagonal esquerda abaixo): esse modo deve ser considerado apenas se as amostras $p'(-1, n)$, com $n = 0..15$ (A' até P'), estão disponíveis.

Para $m, n = 0..7$,

se $m = 7$ e $n = 7$ então,

$$pred8x8_L(m, n) = round\left(\frac{p'(-1, 14) + 3p'(-1, 15)}{4}\right),$$

senão,

$$pred8x8_L(m, n) = round\left(\frac{p'(-1, m+n) + 2p'(-1, m+n+1) + p'(-1, m+n+2)}{4}\right).$$

- *Modo 4* (Diagonal direita abaixo): esse modo deve ser considerado apenas se as amostras $p'(-1, n)$, com $n = 0..7$ (A' até H'), e $p'(m, -1)$, com $m = -1..7$ (Z' e Q' até X'), estão disponíveis.

Para $m, n = 0..7$,

se $n > m$ então,

$$pred8x8_L(m, n) = round\left(\frac{p'(-1, n-m-2) + 2p'(-1, n-m-1) + p'(-1, n-m)}{4}\right),$$

senão, se $n < m$ então,

$$pred8x8_L(m, n) = round\left(\frac{p'(m-n-2, -1) + 2p'(m-n-1, -1) + p'(m-n, -1)}{4}\right),$$

senão, se $n = m$ então,

$$pred8x8_L(m, n) = round\left(\frac{p'(-1, 0) + 2p'(-1, -1) + p'(0, -1)}{4}\right).$$

- *Modo 5* (Vertical à direita): esse modo deve ser considerado apenas se as amostras $p'(-1, n)$, com $n = 0..7$ (A' até H'), e $p'(m, -1)$, com $m = -1..7$ (Z' e Q' até X'), estão disponíveis.

Para $m, n = 0..7$,

seja $zVR = 2n - m$,

se $zVR = 0, 2, 4, 6, 8, 10, 12, \text{ ou } 14$ então,

$$pred8x8_L(m, n) = round\left(\frac{p'(-1, n - \lfloor m/2 \rfloor - 1) + p'(-1, n - \lfloor m/2 \rfloor)}{2}\right),$$

senão, se $zVR = 1, 3, 5, 7, 9, 11, \text{ ou } 13$ então,

$$\begin{aligned} & \text{pred8x8}_L(m, n) = \\ & \text{round} \left(\frac{p'(-1, n - \lfloor m/2 \rfloor - 2) + 2p'(-1, n - \lfloor m/2 \rfloor - 1) + p'(-1, n - \lfloor m/2 \rfloor)}{4} \right), \end{aligned}$$

senão, se $zVR = -1$ então,

$$\text{pred8x8}_L(m, n) = \text{round} \left(\frac{p'(0, -1) + 2p'(-1, -1) + p'(-1, 0)}{4} \right),$$

senão, se $zVR = -2, -3, -4, -5, -6, \text{ ou } -7$ então,

$$\begin{aligned} & \text{pred8x8}_L(m, n) = \\ & \text{round} \left(\frac{p'(m - 2n - 1, -1) + 2p'(m - 2n - 2, -1) + p'(m - 2n - 3, -1)}{4} \right). \end{aligned}$$

- *Modo 6* (Horizontal abaixo): esse modo deve ser considerado apenas se as amostras $p'(-1, n)$, com $n = 0..7$ (A' até H'), e $p'(m, -1)$, com $m = -1..7$ (Z' e Q' até X'), estão disponíveis.

Para $m, n = 0..7$,

seja $zHD = 2m - n$,

se $zHD = 0, 2, 4, 6, 8, 10, 12, \text{ ou } 14$ então,

$$\text{pred8x8}_L(m, n) = \text{round} \left(\frac{p'(m - \lfloor n/2 \rfloor - 1, -1) + p'(m - \lfloor n/2 \rfloor, -1)}{2} \right),$$

senão, se $zHD = 1, 3, 5, 7, 9, 11, \text{ ou } 13$ então,

$$\begin{aligned} & \text{pred8x8}_L(m, n) = \\ & \text{round} \left(\frac{p'(m - \lfloor n/2 \rfloor - 2, -1) + 2p'(m - \lfloor n/2 \rfloor - 1, -1) + p'(m - \lfloor n/2 \rfloor, -1)}{4} \right), \end{aligned}$$

senão, se $zHD = -1$ então,

$$\text{pred8x8}_L(m, n) = \text{round} \left(\frac{p'(0, -1) + 2p'(-1, -1) + p'(-1, 0)}{4} \right),$$

senão, se $zHD = -2, -3, -4, -5, -6, \text{ ou } -7$ então,

$$\begin{aligned} & \text{pred8x8}_L(m, n) = \\ & \text{round} \left(\frac{p'(-1, n - 2m - 1) + 2p'(-1, n - 2m - 2) + p'(-1, n - 2m - 3)}{4} \right). \end{aligned}$$

- *Modo 7* (Vertical à esquerda): esse modo deve ser considerado apenas se as amostras $p'(-1, n)$, com $n = 0..15$ (A' até P'), estão disponíveis.

Para $m, n = 0..7$,

se $m = 0, 2, 4, \text{ ou } 6$ então,

$$\text{pred8x8}_L(m, n) = \text{round} \left(\frac{p'(-1, n + \lfloor m/2 \rfloor) + p'(-1, n + \lfloor m/2 \rfloor + 1)}{2} \right),$$

senão, se $m = 1, 3, 5, \text{ ou } 7$ então,

$$\text{pred8x8}_L(m, n) = \text{round} \left(\frac{p'(-1, n + \lfloor m/2 \rfloor) + 2p'(-1, n + \lfloor m/2 \rfloor + 1) + p'(-1, n + \lfloor m/2 \rfloor + 2)}{4} \right).$$

- *Modo 8* (Horizontal acima): esse modo deve ser considerado apenas se as amostras $p'(m, -1)$, com $m = 0..7$ (Q' até X'), estão disponíveis.

Para $m, n = 0..7$,

seja $zHU = n + 2m$,

se $zHU = 0, 2, 4, 6, 8, 10, \text{ ou } 12$ então,

$$\text{pred8x8}_L(m, n) = \text{round} \left(\frac{p'(m + \lfloor n/2 \rfloor, -1) + p'(m + \lfloor n/2 \rfloor + 1, -1)}{2} \right),$$

senão, se $zHU = 1, 3, 5, 7, 9, \text{ ou } 11$ então,

$$\text{pred8x8}_L(m, n) = \text{round} \left(\frac{p'(m + \lfloor n/2 \rfloor, -1) + 2p'(m + \lfloor n/2 \rfloor + 1, -1) + p'(m + \lfloor n/2 \rfloor + 2, -1)}{4} \right),$$

senão, se $zHU = 13$ então,

$$\text{pred8x8}_L(m, n) = \text{round} \left(\frac{p'(6, -1) + 3p'(7, -1)}{4} \right),$$

senão, se $zHU > 13$ então,

$$\text{pred8x8}_L(m, n) = p'(7, -1).$$

3.3.3 Modos de Predição I16

Os quatro modos de predição para blocos de luminância 16×16 *pixels* são mostrados na Figura. 3.13. As amostras $\text{pred16x16}_L(m, n)$ do bloco atual são preditas a partir das amostras $p(m, n)$ (H e V) dos blocos vizinhos, conforme detalhado mais adiante. Como exemplo, a Figura 3.14 mostra um bloco 16×16 a ser predito e suas respectivas predições I16.

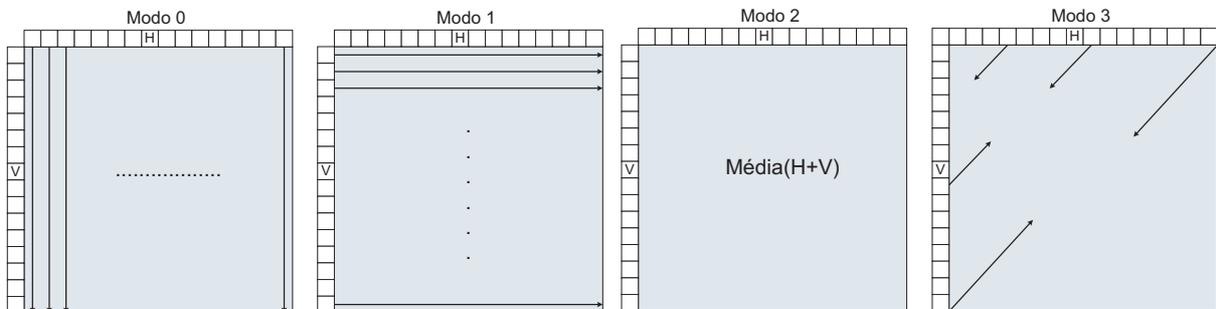


Figura 3.13: Predição intra para blocos de luminância com tamanho 16×16 *pixels*.

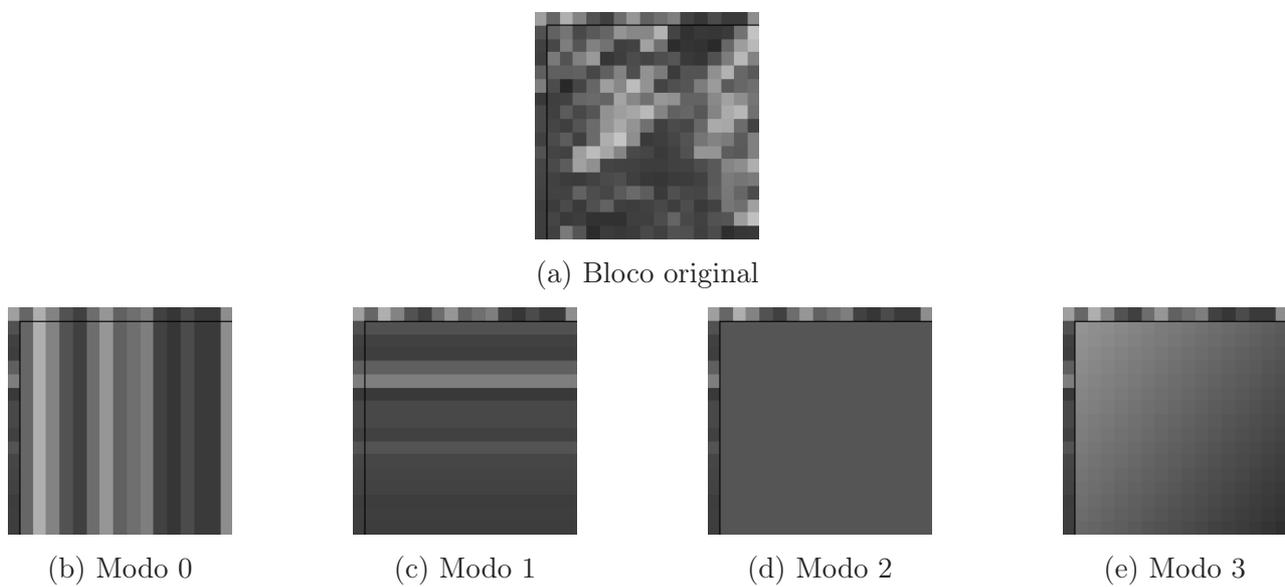


Figura 3.14: Exemplo de predições $I16$: (a) bloco original a ser predito; (b) a (e) predições geradas a partir dos modos de predição 0 a 3, respectivamente.

- *Modo 0* (Vertical): esse modo deve ser considerado apenas se as amostras $p(-1, n)$, com $n = 0..15$ (H), estão disponíveis.

Para $m, n = 0..15$,

$$pred16x16_L(m, n) = p(-1, n).$$

- *Modo 1* (Horizontal): esse modo deve ser considerado apenas se as amostras $p(m, -1)$, com $n = 0..15$ (V), estão disponíveis.

Para $m, n = 0..15$,

$$pred16x16_L(m, n) = p(m, -1).$$

- *Modo 2* (DC): esse modo é sempre considerado.

Para $m, n = 0..15$,

Se as amostras $p(-1, n)$, com $n = 0..15$ (H), e $p(m, -1)$, com $m = 0..15$ (V), estão disponíveis, então,

$$pred16x16_L(m, n) = round \left(\frac{\sum_{j=0}^{15} p(-1, j) + \sum_{i=0}^{15} p(i, -1)}{32} \right).$$

senão, se alguma amostra $p(-1, n)$, com $n = 0..15$ (H), não está disponível e as amostras $p(m, -1)$, com $m = 0..15$ (V), estão disponíveis, então,

$$pred16x16_L(m, n) = round \left(\frac{\sum_{i=0}^{15} p(i, -1)}{16} \right),$$

senão, se alguma amostra $p(m, -1)$, com $m = 0..15$ (V), não está disponível e as amostras $p(-1, n)$, com $n = 0..3$ (H), estão disponíveis, então,

$$pred16x16_L(m, n) = round \left(\frac{\sum_{j=0}^{15} p(-1, j)}{8} \right),$$

senão, se alguma amostra $p(-1, n)$, com $n = 0..15$ (H), não está disponível e alguma amostra $p(m, -1)$, com $m = 0..15$ (V), também não está disponível, então,

$$pred16x16_L(m, n) = 128.$$

- *Modo 3* (Plano): esse modo deve ser considerado apenas se as amostras $p(-1, n)$, com $n = -1..15$, e $p(m, -1)$, com $m = 0..15$, estão disponíveis.

Para $m, n = 0..15$,

$$pred16x16_L(m, n) = clip(round((a + b(n - 7) + c(m - 7))/32)).$$

com

$$a = 16(p(15, -1) + p(-1, 15))$$

$$b = round(5H/64)$$

$$c = round(5V/64),$$

e

$$H = \sum_{j=0}^7 (j+1)(p(-1, 8+j) - p(-1, 6-j)),$$

$$V = \sum_{i=0}^7 (i+1)(p(8+i, -1) - p(6-i, -1)).$$

A função *clip* é dada por,

$$clip(x) = max(0, min(x, 255)), \quad (3.42)$$

onde $max(x, y)$ e $min(x, y)$ retornam o máximo e o mínimo valor entre x e y , respectivamente.

A Figura 3.15 mostra exemplos de quadros preditos e residuais obtidos a partir das predições $I16$, $I8$ e $I4$, para a imagem “Lena”.

A seguir será apresentada uma técnica de otimização que contribui para o alto desempenho do H.264/AVC.

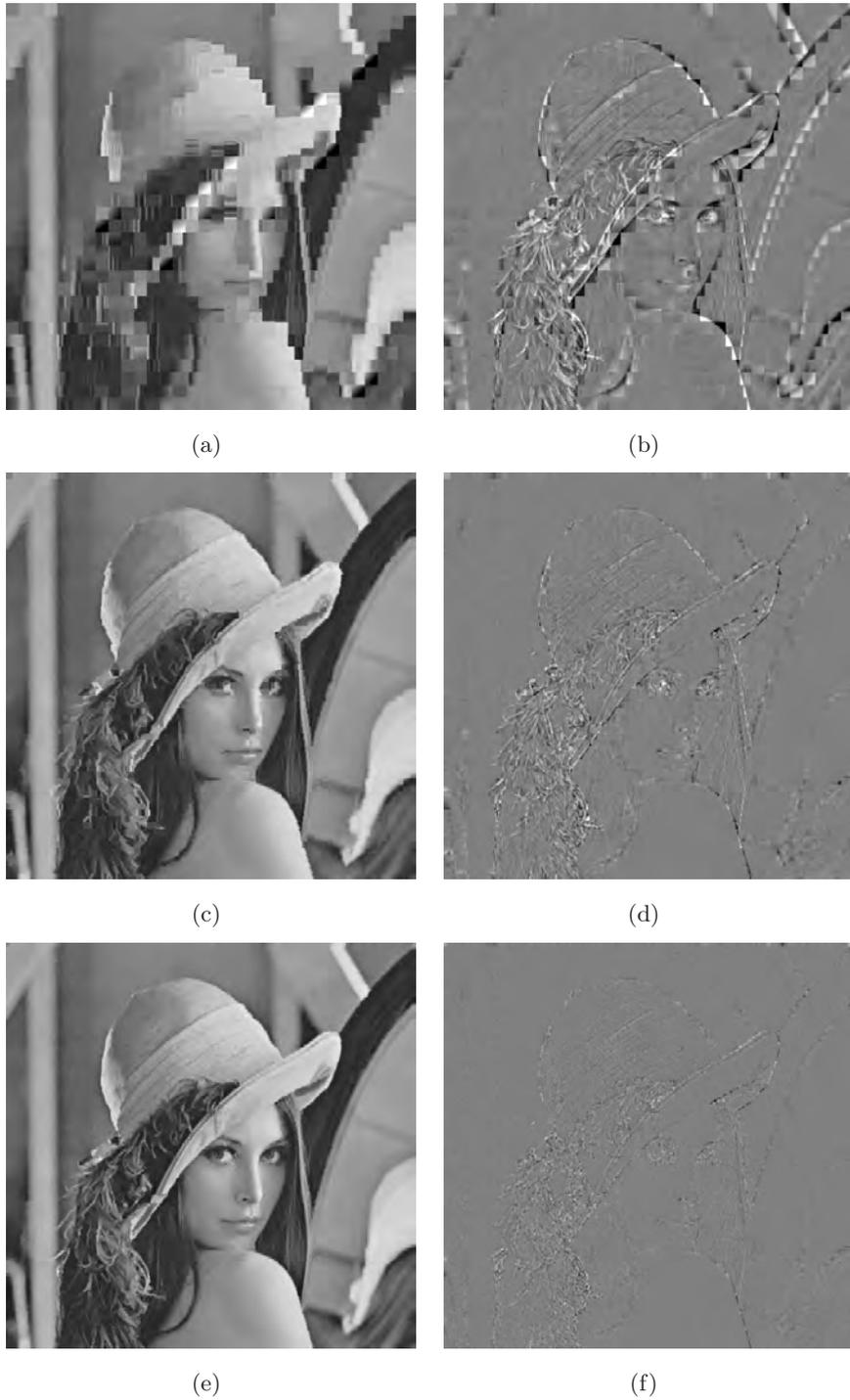


Figura 3.15: Exemplos de quadros preditos (a, c, e) e residuais (b, d, f) obtidos a partir das predições: I_{16} (a, b); I_8 (c, d); e I_4 (e, f), para a imagem “Lena”.

3.4 Otimização Taxa-Distorção Lagrangeana

Para atingir o mais alto grau de eficiência, o H.264/AVC utiliza uma técnica não normativa denominada *Otimização Taxa-Distorção Lagrangeana* [92] para decidir qual modo de predição deve ser escolhido. De uma maneira geral, os macroblocos são codificados utilizando-se todos os possíveis modos e escolhem-se aqueles que resultam no menor custo taxa-distorção J local. O algoritmo de otimização que codifica a componente de luminância de um macrobloco em um quadro intra pode ser descrito da seguinte maneira:

1. *Cálculo do multiplicador de Lagrange*: dado o parâmetro de quantização QP , que pode variar de 0 a 51, o multiplicador de Lagrange λ é calculado por meio de,

$$\lambda(QP) = 0,85 \cdot 2^{(QP-12)/3}. \quad (3.43)$$

2. *Seleção do melhor modo de predição I4*: o custo $J_n(i)$ associado a cada modo i de predição I4 é calculado por,

$$J_n(i) = D_n(i) + \lambda R_n(i), \quad (3.44)$$

onde $D_n(i)$ e $R_n(i)$ representam a distorção e a taxa de *bits* associadas ao modo de predição i para o sub-bloco n . Considerando $n = 0..15$, determina-se,

$$\hat{i}_n = \underset{i}{\operatorname{argmin}} \{J_n(i)\}, \quad (3.45)$$

e o custo total J_{I4} para o macrobloco é calculado por,

$$J_{I4} = \sum_{n=0}^{16} J_n(\hat{i}_n). \quad (3.46)$$

3. *Seleção do melhor modo de predição I8*: o custo $J_n(i)$ associado a cada modo i de predição I8 é calculado para cada sub-bloco n . Considerando $n = 0..3$, determina-se \hat{i}_n por meio da Equação 3.45 e o custo total J_{I8} para o macrobloco é calculado por,

$$J_{I8} = \sum_{n=0}^4 J_n(\hat{i}_n). \quad (3.47)$$

4. *Seleção do melhor modo de predição I16*: o custo $J(i)$ associado a cada modo i de predição I16 é calculado para o macrobloco. Determina-se,

$$\hat{i} = \underset{i}{\operatorname{argmin}} \{J(i)\}, \quad (3.48)$$

e o custo J_{I16} para o macrobloco é calculado por,

$$J_{I16} = J(\hat{i}). \quad (3.49)$$

5. *Seleção entre I4, I8 e I16*: selecionam-se os modos de predição que resultam no menor entre os custos J_{I4} , J_{I8} e J_{I16} .

Pode-se, ainda, realizar a busca acima descrita para um intervalo de QPs , selecionando-se o QP e os modos que resultam no menor custo. No caso em que apenas um único valor de QP é considerado, o número máximo de computações N da função de custo J , por macrobloco, é dado por,

$$N = M4 \cdot 16 + M8 \cdot 4 + M16, \quad (3.50)$$

onde $M4$, $M8$ e $M16$ representam o número de modos das predições $I4$, $I8$ e $I16$, respectivamente. Ou seja, por macrobloco computa-se $(9 \times 16 + 9 \times 4 + 4) = 188$ cálculos de taxa-distorção [93]. Se um quadro for composto por 512×512 *pixels*, ou seja, 1024 macroblocos, o número de computações da função de custo para a imagem inteira seria $1024 \times 188 = 192512$. Isso pode parecer muito, mas na verdade há estudos indicando que a estimação de movimento do H.264/AVC é responsável por aproximadamente 90% da complexidade do codificador, enquanto a decisão de modo e a predição intra representam apenas 2% da complexidade total [94]. Mesmo assim, alguns trabalhos apresentam formas de se acelerar a decisão de modo para a compressão intra, como a contribuição de Li, Ngan e Wei [95], que propõe uma redução de complexidade de 75% a 82% para um aumento de taxa de 1,5% a 3,7% e redução de PSNR de 0,1 a 0,25 dB.

3.5 Codificação Aritmética Binária Adaptativa Baseada em Contexto

A *codificação aritmética binária adaptativa baseada em contexto* (ou, do inglês, *Context-based Adaptive Binary Arithmetic Coding* ou *CABAC*) [96] é uma das técnicas sugeridas pelo H.264/AVC para se realizar a codificação de entropia. O alto desempenho em compressão alcançado pelo CABAC é devido ao emprego conjunto de três técnicas básicas, a saber:

1. Codificação aritmética;
2. Seleção de um modelo de probabilidade que depende do contexto; e
3. Adaptação dos modelos de probabilidade de acordo com estatísticas locais.

A codificação ocorre em três etapas:

1. *Binarização*: o CABAC utiliza a *codificação aritmética binária*, ou seja, apenas 0's e 1's podem ser codificados. Valores não binários, como os coeficientes da transformada e os vetores de movimento, por exemplo, precisam ser inicialmente binarizados.
2. *Seleção do modelo de probabilidade (ou contexto)*: um modelo de probabilidade é selecionado de acordo com o contexto do *bit* a ser codificado. Para cada contexto os modelos de probabilidade armazenam as probabilidades de ocorrência de 0's e 1's.

3. *Codificação aritmética*: a codificação aritmética é realizada de acordo com o modelo de probabilidade escolhido no passo anterior.
4. *Atualização do modelo de probabilidade*: as frequências de ocorrência de 0's e 1's para o modelo selecionado são atualizadas. Por exemplo, se um determinado modelo de probabilidade é selecionado e o *bit* a ser codificado apresenta o valor 1, a frequência de 1's será incrementada e, na próxima vez que esse modelo for escolhido, a probabilidade de ocorrência do valor 1 será ligeiramente superior à vez anterior.

Os princípios gerais da codificação aritmética foram apresentados na Seção 2.2.1.2 do Capítulo 2. O conceito de adaptação ao contexto e o motivo que justifica a utilização dessa técnica foram expostos na Seção 2.2.1.3 do mesmo capítulo. Mais detalhes a respeito do esquema de binarização, do codificador aritmético e dos modelos de contexto empregados pelo H.264/AVC, o leitor pode encontrar na documentação técnica que define o padrão [9].

3.6 O H.264/AVC utilizado como um Codificador de Imagens

O H.264/AVC é um padrão de compressão de vídeo e não foi originalmente concebido para ser aplicado como uma ferramenta de compressão de imagens. Apesar disso, os avanços incorporados ao H.264/AVC não apenas estabelecem um novo patamar em compressão de vídeo, mas também fazem dele um excelente codificador de *imagens estáticas* [88, 97, 98]. Deve-se isso principalmente à predição intra-quadro (*I4*, *I8* e *I16*), à otimização taxa-distorção lagrangeana e à codificação aritmética binária adaptativa baseada em contexto, estudadas nas seções 3.3, 3.4 e 3.5, respectivamente.

Se o H.264/AVC for ajustado para codificar apenas um único quadro de vídeo, ele irá operar exclusivamente no modo intra, comportando-se como um codificador de imagens. Esse codificador, ilustrado no diagrama de blocos da Figura. 3.16, será chamado de AVC-I. A grande surpresa é que o AVC-I supera, em muitos casos, codificadores de imagens estáticas considerados o estado-da-arte, como o JPEG2000. Os ganhos do AVC-I sobre o JPEG2000 são tipicamente da ordem de 0,25dB a 0,5dB em PSNR para imagens em escala de cinza. Nas Figuras 3.17 e 3.18 são apresentadas curvas PSNR comparando esses dois codificadores na compressão das imagens “Lena”, “goldhill”, “cameraman”, “Barbara”, “peppers” e “baboon”.

Os resultados apresentados nas Figuras 3.17 e 3.18 por si só representam um ganho significativo. Porém, o AVC-I parece ter uma capacidade ainda maior em se adaptar a conteúdos heterogêneos. Para *documentos compostos* (mistura de imagens naturais, textos e gráficos), os ganhos em PSNR são ainda mais substanciais, podendo ultrapassar em alguns casos a marca de 4 dB com relação ao JPEG2000 [88]. As Figuras 3.19 e 3.20 mostram curvas PSNR que comparam os CODECs AVC-I e JPEG2000 na compressão de seis documentos compostos. Observe como em todos os casos o AVC-I supera objetivamente o JPEG2000. Quanto à complexidade, estudos indicam que o decodificador JPEG2000 é mais complexo que o decodificador AVC-I, tanto em termos de número de operações básicas, quanto em termos de memória [99]. Quanto a comparação entre a complexidade dos respectivos codificadores, a literatura parece carecer de uma análise detalhada.

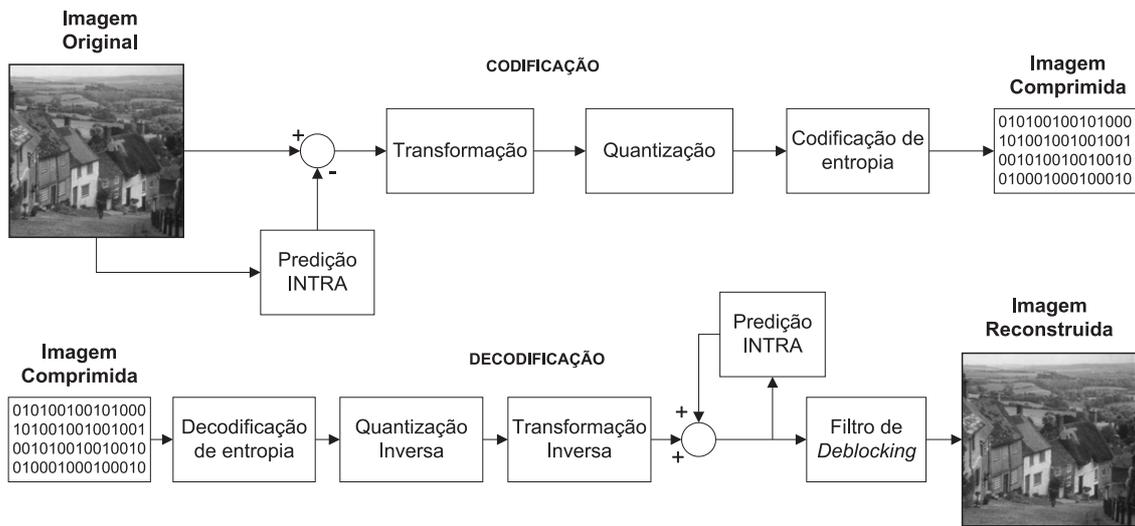


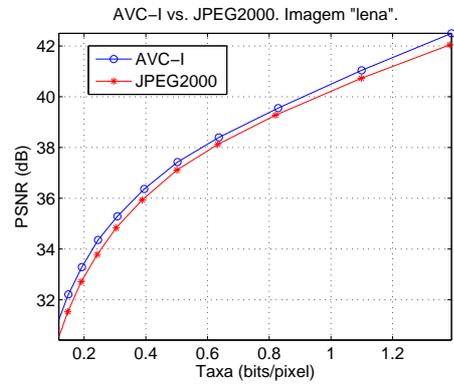
Figura 3.16: Diagrama de blocos do AVC-I: codificação e decodificação.

Sem dúvida, o H.264/AVC-I estabelece um novo patamar de desempenho em compressão e muito provavelmente irá influenciar futuros padrões de codificação de imagens estáticas. Além disso, uma vez que tal CODEC não foi originalmente concebido para ser aplicado exclusivamente à compressão de imagens estáticas, muitas melhorias podem ser propostas com o objetivo de se elevar ainda mais sua eficiência.

No próximo capítulo serão tratados alguns dos principais aspectos relacionados à compressão de documentos digitais.



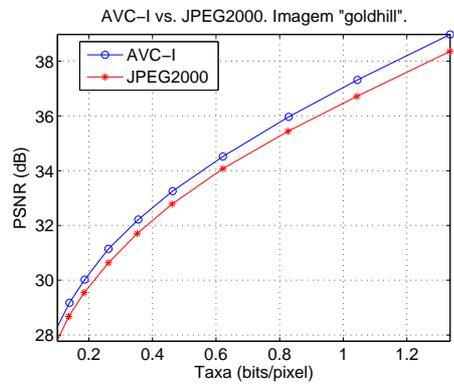
(a)



(b)



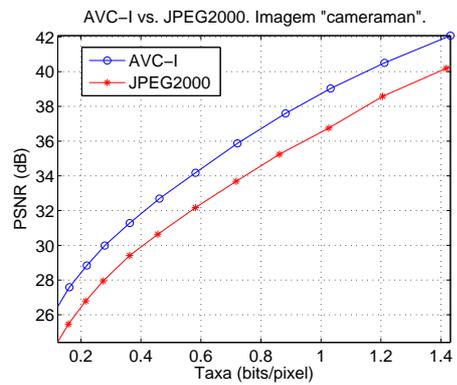
(c)



(d)



(e)

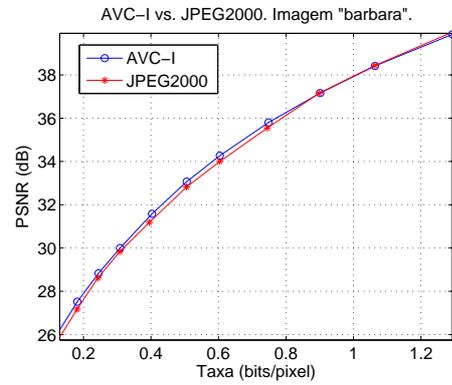


(f)

Figura 3.17: Curvas PSNR comparando o AVC-I e o JPEG2000 na compressão das imagens: (a, b) "Lena"; (c, d) "goldhill"; e (e, f) "cameraman".



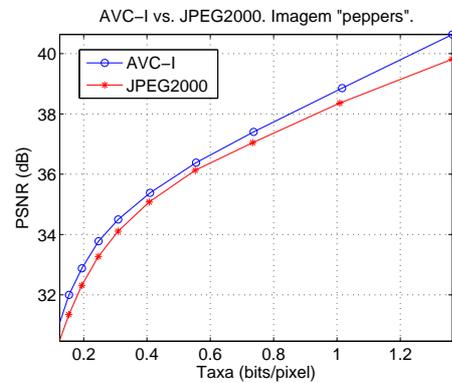
(a)



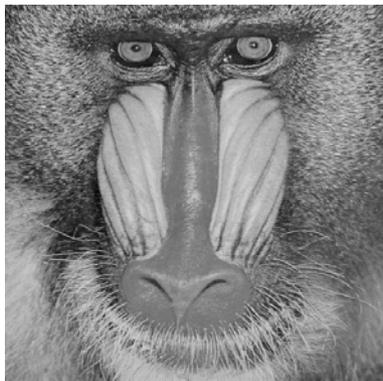
(b)



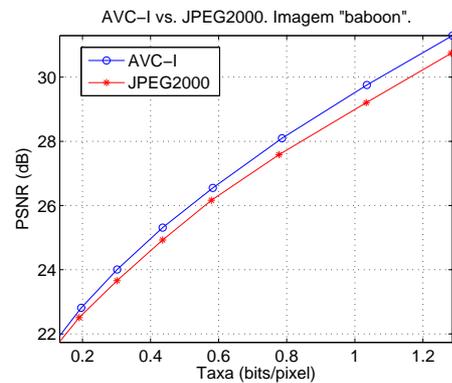
(c)



(d)



(e)

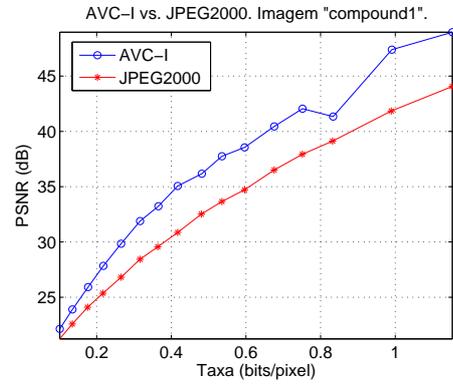


(f)

Figura 3.18: Curvas PSNR comparando o AVC-I e o JPEG2000 na compressão das imagens: (a, b) "barbara"; (c, d) "peppers"; e (e, f) "baboon".



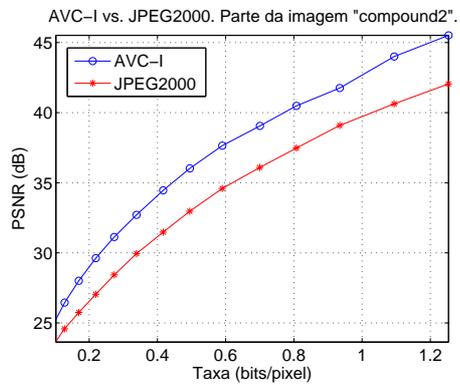
(a)



(b)



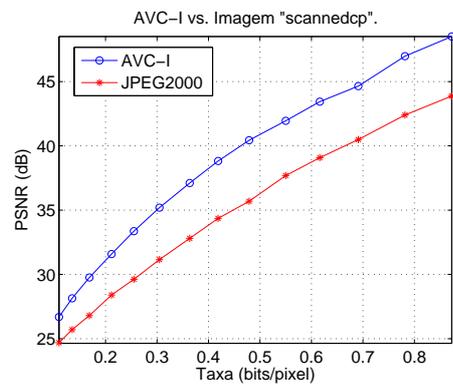
(c)



(d)

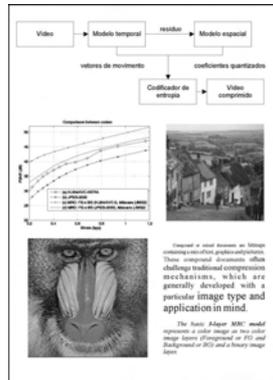


(e)

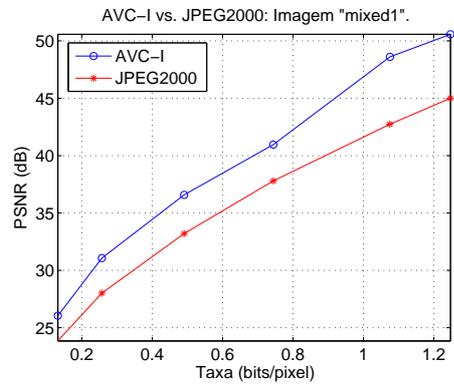


(f)

Figura 3.19: Curvas PSNR comparando o AVC-I e o JPEG2000 na compressão de documentos compostos: (a, b) “compound1”; (c, d) “compound2”; e (e, f) “scannedcp”.



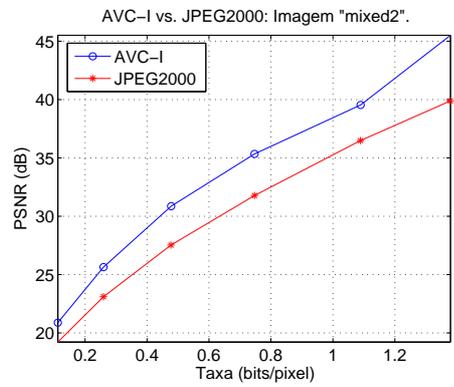
(a)



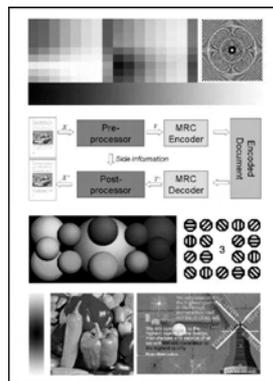
(b)



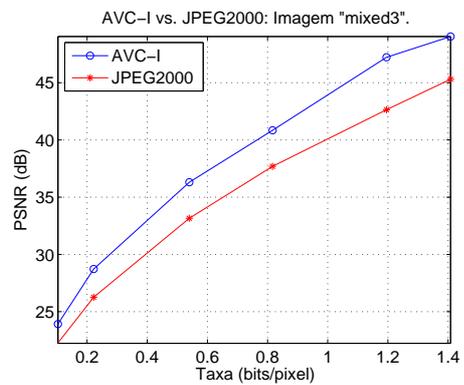
(c)



(d)



(e)



(f)

Figura 3.20: Curvas PSNR comparando o AVC-I e o JPEG2000 na compressão de documentos compostos: (a, b) “mixed1”; (c, d) “mixed2”; e (e, f) “mixed3”.

Capítulo 4

Compressão de Documentos Digitais

4.1 Considerações Iniciais

Um documento digital é tipicamente representado nas formas vetorial ou *raster*. Um documento representado na forma vetorial é composto por uma série de instruções que indicam onde cada objeto deve ser desenhado. Exemplos são os arquivos PS (*PostScript*) [100] e PDF (*Portable Document File*) [101]. Documentos na forma *raster*, por outro lado, são mapas de *bits*, ou seja, imagens. A vantagem da representação *raster* sobre a representação vetorial é que na representação *raster*, o documento, para ser visualizado, não precisa ser previamente construído a partir da interpretação de instruções, como no caso da representação vetorial [102]. Como documentos digitais geralmente são constituídos a partir de uma mistura de elementos textuais, gráficos e pictóricos ¹, muitas vezes também são chamados de *documentos compostos*. O presente trabalho trata de documentos compostos representados na forma *raster*.

Algoritmos de compressão são desenvolvidos tendo-se em mente algum tipo particular de imagem e aplicação. Quando se comprime texto, é importante preservar as bordas e os formatos dos caracteres para que a leitura seja facilitada. O sistema visual humano, por outro lado, trabalha diferentemente para imagens em tons contínuos, melhor mascarando erros de alta frequência [103]. A Figura 4.1 mostra uma parte ampliada do documento “compound1”, codificado com o AVC-I a taxa de 0,32 bpp. Observe o *efeito de borda*² ao redor dos caracteres.

Documentos compostos são tipicamente codificados de duas maneiras: como uma imagem única, caso em que se utiliza um codificador único, ou como um conjunto de regiões decompostas em camadas, situação em que múltiplos codificadores podem ser utilizados. A Seção 4.2 apresenta conceitos relacionados à compressão baseada em um codificador único. Já a Seção 4.3 trata da compressão baseada em múltiplos codificadores.

¹Pictórico, do latim *pictore*, adjetivo: relativo à pintura. Tal adjetivo será utilizado no sentido figurado para especificar as regiões do documento formadas por imagens naturais.

²Artefatos presentes nas bordas dos caracteres.



Figura 4.1: Parte ampliada do documento “compound1”. Observe os efeitos de borda ao redor dos caracteres.

4.2 Compressão de Documentos baseada em Codificador Único

Alguns autores trabalharam no problema da compressão de documentos utilizando a abordagem de codificador único. Konstantinides e Tretter [104], por exemplo, propuseram a quantização adaptativa dentro do arcabouço das extensões JPEG [105]. Ramos e Queiroz [106] também utilizaram um único codificador JPEG para comprimir documentos compostos. Propuseram um processo baseado em limiarização que analisa, para cada bloco de 8×8 *pixels* da imagem, o custo taxa-distorção associado à codificação de cada coeficiente quantizado da DCT. Aqueles que não são eficientes em termos de taxa-distorção são zerados, tornando o processo de limiarização adaptativo ao conteúdo local da imagem.

Geralmente, na codificação por transformada com otimização taxa-distorção, o sinal é dividido em blocos x_n , cada bloco contribuindo com uma taxa de *bits* R_n para a taxa total R , ou seja,

$$R = \sum_n R_n. \quad (4.1)$$

A distorção D é uma função do erro de quantização $\hat{x}_n - x_n$. Utilizando-se funções de distorção, como o MSE, definido na Seção 2.2.2, a distorção global pode ser calculada a partir de distorções locais, ou seja,

$$D = \sum_n D_n, \quad (4.2)$$

onde D_n é a distorção para o n -ésimo bloco e é definida como,

$$D_n = [\hat{x}_n - x_n]^2. \quad (4.3)$$

A otimização taxa-distorção envolve a minimização de um função de custo,

$$J = D + \lambda R, \quad (4.4)$$

onde λ é um multiplicador de Lagrange. Assim,

$$J = \sum_n (D_n + \lambda R_n). \quad (4.5)$$

Pode-se, então, propor uma interpretação em que o significado da taxa varia com o conteúdo do bloco a ser codificado, ou seja,

$$R'_n = R_n u_n, \quad (4.6)$$

onde u_n é um fator de ponderação para a taxa do n -ésimo bloco. Desse modo, pode-se classificar os blocos de imagem em um número discreto de classes representativas e aplicar uma ponderação específica a cada classe. Ou seja,

$$J = \sum_n (D_n + \lambda_n R_n), \quad (4.7)$$

onde λ_n é interpretado como um fator de ponderação aplicado à taxa associada ao macrobloco n .

No H.264/AVC, a variação do parâmetro de quantização QP pode ser realizada a cada macrobloco, e o valor de λ é calculado a partir do valor de QP . Essas características serão exploradas pelo método proposto no Capítulo 5.

4.3 Compressão de Documentos baseada em Múltiplos Codificadores: o Modelo de Imageamento *Mixed Raster Content*

O modelo MRC (*Mixed Raster Content*) básico [10] representa um documento por meio de duas camadas de imagem em tons contínuos (*Foreground* ou FG e *Background* ou BG) e uma máscara binária que determina se um *pixel* pertence ao FG ou ao BG [102, 107], conforme ilustrado na Figura 4.2.

Partindo-se dessa idéia, o modelo pode ser estendido para o caso mais geral no qual N planos p_k são sequencialmente impressos sobre um único plano p_0 por meio de suas respectivas máscaras binárias m_k , para compor a imagem final, $x_N(i, j)$,

$$x_k(i, j) = \begin{cases} p_k & : k = 0 \\ m_k(i, j)p_k(i, j) + ((1 - m_k))x_{k-1}(i, j) & : 1 \leq k \leq N \end{cases} . \quad (4.8)$$

No presente trabalho, porém, será considerado o modelo MRC básico de três camadas, no qual o FG é impresso sobre o BG, através da máscara binária M , gerando a imagem $x(i, j)$, ou seja,

$$x(i, j) = M(i, j) \cdot BG(i, j) + (1 - M(i, j)) \cdot FG(i, j). \quad (4.9)$$

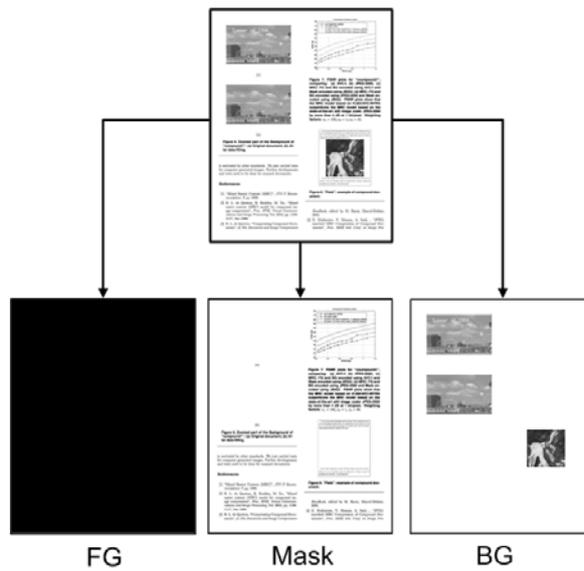


Figura 4.2: Ilustração do modelo de imageamento MRC. O documento original é representado utilizando-se três camadas: *Foreground* (FG), *Background* (BG) e máscara (*Mask*).

A Figura 4.3 mostra um documento x , uma possível máscara M e as componentes $M(i, j) \cdot BG(i, j)$ e $(1 - M(i, j)) \cdot FG(i, j)$.

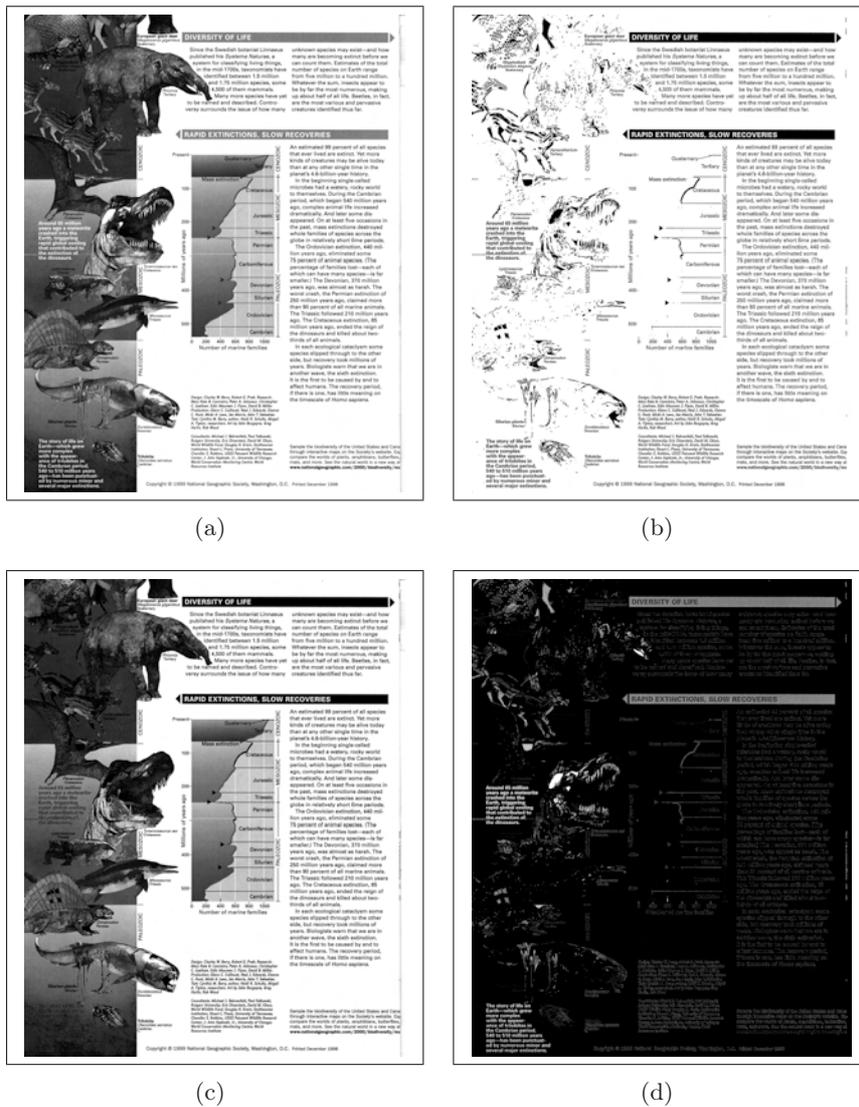


Figura 4.3: (a) Documento original x ; (b) máscara binária M ; e (c) $M(i, j) \cdot BG(i, j)$; e (d) $(1 - M(i, j)) \cdot FG(i, j)$. Na máscara binária a cor branca é representada pelo valor numérico 1, enquanto o preto é representado por 0. Isso garante que $x(i, j) = M(i, j) \cdot BG(i, j) + (1 - M(i, j)) \cdot FG(i, j)$.

Após a decomposição do documento, cada camada pode ser processada e comprimida por meio de algoritmos específicos. O processamento do FG e do BG pode incluir a mudança de resolução espacial e o preenchimento de regiões “irrelevantes” (*data-filling*). O algoritmo de compressão utilizado para uma determinada camada deve ser adequado ao seu conteúdo, possibilitando o aumento da compressão, enquanto a distorção é reduzida. Após codificadas, as camadas são empacotadas e enviadas ao decodificador. No decodificador, a imagem é recomposta por meio do processo inverso. A Figura 4.4 ilustra a codificação MRC acima descrita.

O ganho potencial de modelos como o MRC reside exatamente no fato de cada plano poder ser processado e comprimido por meio de um algoritmo específico, ajustado às características estatísticas dos dados a serem codificados. Em uma análise de taxa e distorção, espera-se que a combinação dos desempenhos observados na codificação de cada plano resulte em um desempenho global capaz de superar a abordagem baseada em um codificador único, conforme ilustrado na Figura 4.5. Nessa ilustração, os codificadores 1 a 3 são utilizados na compressão das camadas FG,

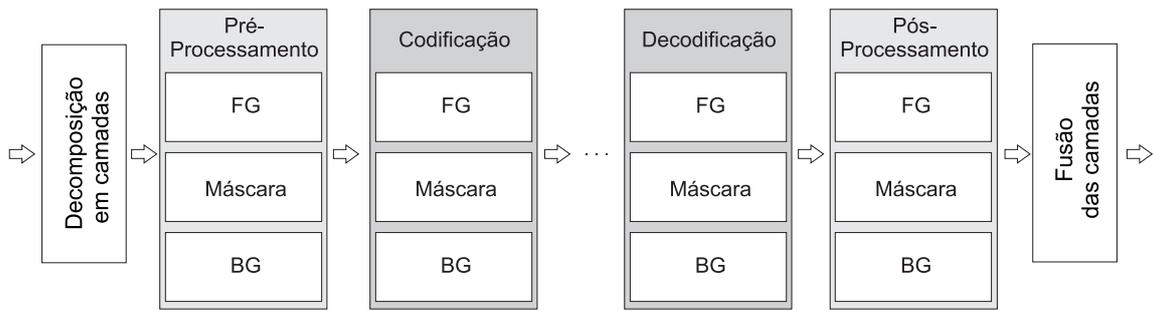


Figura 4.4: Diagrama de blocos ilustrando os procedimentos de decomposição em camadas, pré-processamento, codificação, decodificação, pós-processamento e fusão das camadas.

máscara e BG, respectivamente, o que gera um comportamento de taxa e distorção local para cada plano. Como tais codificadores foram ajustados de maneira a apresentarem a máxima eficiência na compressão de seus respectivos planos, espera-se que o desempenho de taxa e distorção global na codificação MRC seja superior à codificação que emprega apenas um único codificador, apesar do custo indireto em taxa associado à compressão de vários planos.

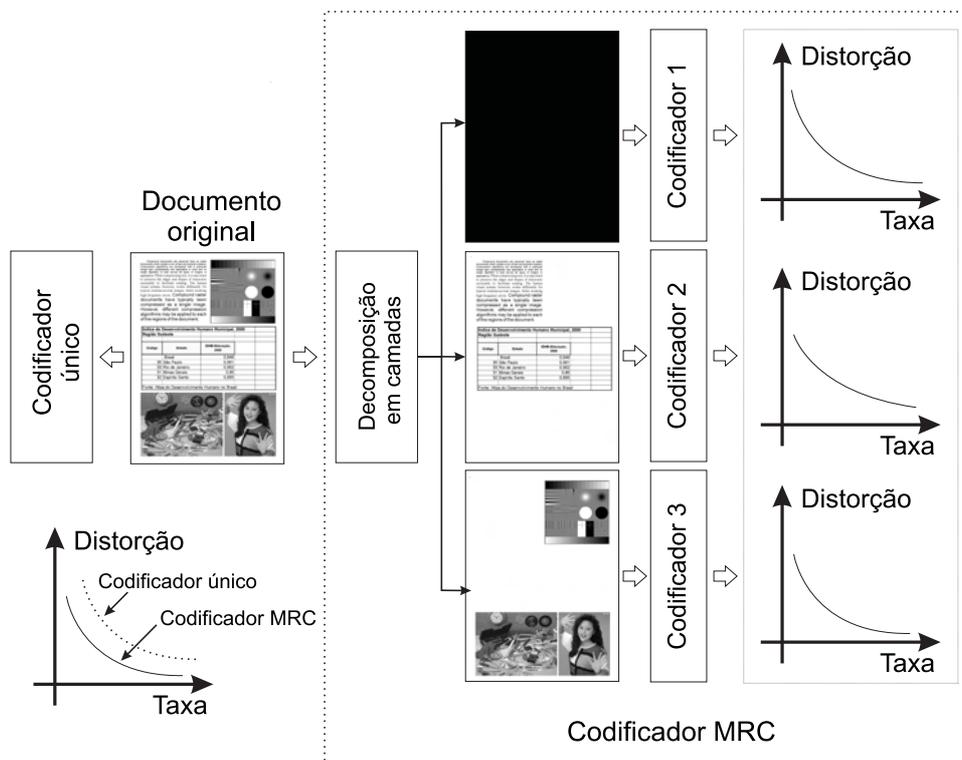


Figura 4.5: Cada plano do modelo MRC pode ser processado e comprimido por meio de algoritmos específicos, resultando em um comportamento global de taxa e distorção mais eficiente do que a abordagem baseada em um codificador único.

Pelo que foi exposto até aqui, é possível identificar três das principais variáveis que afetam o desempenho da compressão baseada no modelo MRC, que são:

1. O algoritmo de *decomposição (ou segmentação)* em camadas: dependendo do conteúdo de cada camada, o codificador apresentará um melhor ou pior desempenho em termos de taxa

e distorção. Além disso, a busca pelo desempenho ótimo pode resultar no aumento da complexidade computacional do codificador;

2. Os algoritmos de *pré/pós-processamento* das camadas: podem incluir alteração de resolução espacial dos planos e *data-filling* (preenchimento de regiões irrelevantes), procedimento discutido na Seção 4.3.2. Em ambos os casos, os dados das camadas FG e BG serão alterados, e isso afeta o desempenho do codificador; e
3. Os algoritmos de *compressão* aplicados às camadas: deve-se escolher, para cada plano, um algoritmo de compressão específico, ajustado à natureza dos dados a serem comprimidos.

A seguir, será abordada a segmentação em camadas.

4.3.1 Segmentação em Camadas

A primeira etapa da codificação MRC é a segmentação em camadas [102]. Vale ressaltar que o desempenho do codificador, em termos de taxa e distorção, é extremamente sensível a esse procedimento. Dentre os métodos de segmentação já propostos [108, 109, 110, 111, 112], há uma classe de algoritmos que se baseia no conceito de otimização taxa-distorção [113, 114, 115]. De uma maneira geral, tais algoritmos avaliam uma função de custo e escolhem a decomposição associada ao menor custo possível. Como o presente trabalho utiliza a PSNR em função da taxa na avaliação objetiva do desempenho dos algoritmos de compressão e, conceitualmente, a PSNR é inversamente proporcional à distorção, optou-se por empregar o paradigma de segmentação baseado em otimização taxa-distorção.

O algoritmo de segmentação descrito a seguir é uma variação do algoritmo baseado em limiarização por blocos proposto por Queiroz, Fan e Tran [113]. Sua versão original opera sobre bloco de 8×8 *pixels*. No intuito de adaptá-lo às características do AVC-I, generalizou-se o algoritmo de forma a torná-lo apto a trabalhar com blocos de qualquer tamanho.

Como o FG e o BG serão codificados por macroblocos (blocos de 16×16 *pixels*), deseja-se encontrar uma máscara de macrobloco $m_n(i, j)$ para cada um dos n macroblocos do documento, conforme a Equação 4.10,

$$m_n(i, j) = u(t_n - x_n(i, j) - 1), \quad (4.10)$$

onde t_n é o limiar do bloco, $x_n(i, j)$ representa o macrobloco original e $u(k)$ é a função degrau discreta, ou seja,

$$u(k) = \begin{cases} 1 & : k \geq 0 \\ 0 & : k < 0 \end{cases}. \quad (4.11)$$

Em um macrobloco há 256 *pixels* e, por isso, 256 é o número de possíveis limiares. Para cada macrobloco n , inicialmente seleciona-se um conjunto de $l \leq 256$ limiares t_n organizados em ordem crescente. Considere, por exemplo, o macrobloco apresentado na Figura 4.6 (a). Os possíveis limiares para esse macrobloco podem ser visualizados na Figura 4.6 (b).

73	75	90	95	88	108	73	72	66	67	69	94	85	96	72	70
82	108	69	75	80	84	78	89	70	86	60	76	81	87	77	101
85	91	85	91	73	97	72	68	70	71	73	101	72	91	73	77
85	84	86	72	84	84	89	76	70	81	83	78	79	71	93	95
75	80	90	84	100	67	71	45	50	80	94	82	98	72	92	74
100	78	95	72	65	44	29	15	39	59	83	77	84	79	97	78
79	81	86	79	54	15	7	11	36	82	85	81	85	81	74	80
76	84	66	78	36	11	7	10	33	82	62	78	78	90	76	92
65	62	38	49	23	11	7	11	25	42	36	75	74	94	68	79
72	72	14	8	7	7	6	8	7	8	11	40	71	85	73	100
75	58	18	11	11	5	4	5	7	15	24	50	70	79	75	74
70	72	70	45	31	5	5	8	41	65	75	57	74	69	87	80
45	74	79	68	57	5	8	6	31	73	88	76	111	68	89	66
68	71	84	61	44	5	12	12	42	70	84	72	74	79	96	75
41	75	63	77	40	12	8	12	31	79	77	79	93	85	72	80
64	90	72	87	31	8	12	10	44	93	65	74	65	82	81	101

(a)

4	10	18	31	40	49	59	64	69	74	79	84	89	94	100
5	11	23	33	41	50	60	65	70	75	80	85	90	95	101
6	12	24	36	42	54	61	66	71	76	81	86	91	96	108
7	14	25	38	44	57	62	67	72	77	82	87	92	97	111
8	15	29	39	45	58	63	68	73	78	83	88	93	98	112

(b)

Figura 4.6: (a) Macrobloco original; e (b) possíveis limiares: $l = 75$.

Em seguida, procura-se minimizar a seguinte função de custo,

$$J = \alpha_1 V_{BG} + \alpha_2 V_{FG} + \alpha_3 N_t, \quad (4.12)$$

onde α_i são pesos, e V_{BG} e V_{FG} são as variâncias dos *pixels* dos macroblocos do BG e do FG, respectivamente. N_t é o número de transições horizontais de $m_n(i, j)$ (a primeira coluna da máscara do macrobloco atual usa como referência a última coluna da máscara do macrobloco anterior). Valores de V_{BG} e V_{FG} mais elevados indicam uma menor compressão dos planos BG e FG, o que contribui para o aumento do custo. Em relação à máscara $m_n(i, j)$, quanto maior o número de transições N_t , menor a compressão, ou seja, maior o custo.

Para cada um dos l limiares t_n , obtêm-se $m_n(i, j)$, e dois conjuntos são definidos,

$$X_{FG} \equiv \{x_n(i, j) | m_n(i, j) = 0\} \quad (4.13)$$

$$X_{BG} \equiv \{x_n(i, j) | m_n(i, j) = 1\}.$$

Define-se n_{FG} e n_{BG} como o número de *pixels* dos conjuntos X_{FG} e X_{BG} , respectivamente. Obviamente, $n_{FG} + n_{BG} = 256$. As variâncias são calculadas pelas equações,

$$V_{FG} = \frac{\sum x_n(i, j)^2}{n_{FG}} - \left(\frac{\sum x_n(i, j)}{n_{FG}} \right)^2$$

$$V_{BG} = \frac{\sum x_n(i, j)^2}{n_{BG}} - \left(\frac{\sum x_n(i, j)}{n_{BG}} \right)^2, \quad (4.14)$$

as quais podem ser implementadas de forma eficiente.

No que se refere ao cálculo do número de transições horizontais N_t , considere, por exemplo, o limiar $t_n = 36$, para o macrobloco mostrado na Figura 4.7 (a). A máscara invertida $\bar{m}_n(i, j)$ relativa a esse limiar é mostrada na Figura 4.7 (b). A Figura 4.7 (c) representa $\bar{m}_{nd}(i, j)$, que é $\bar{m}_n(i, j)$ deslocada de uma coluna à direita. Já a Figura 4.7 (d) mostra $\bar{m}_{ndif}(i, j)$, que é o valor absoluto da diferença entre $\bar{m}_n(i, j)$ e $\bar{m}_{nd}(i, j)$. N_t é dado pelo número de *pixels* 1 observados em $\bar{m}_{ndif}(i, j)$, ou seja,

$$N_t = \sum_{i=0}^{15} \sum_{j=0}^{15} |\bar{m}_n(i, j) - \bar{m}_{nd}(i, j)|. \quad (4.15)$$

No exemplo em questão, para $t_n = 36$, N_t é igual a 22.

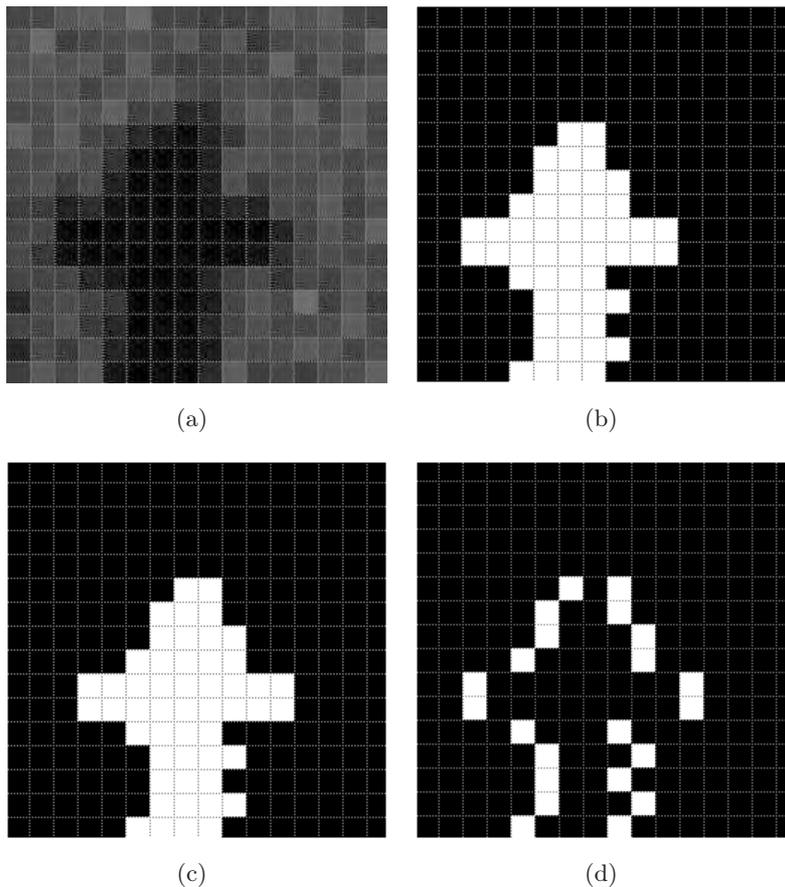


Figura 4.7: (a) Macrobloco original; (b) $\bar{m}_n(i, j)$: máscara invertida relativa ao limiar $t_n = 36$. (c) $\bar{m}_{nd}(i, j)$ máscara deslocada de uma coluna à direita; e (d) valor absoluto da diferença entre $\bar{m}_n(i, j)$ e $\bar{m}_{nd}(i, j)$, ou seja, $\bar{m}_{ndif}(i, j) = |\bar{m}_n(i, j) - \bar{m}_{nd}(i, j)|$.

Quanto aos pesos α_i , sem perda de generalidade, pode-se normalizar um deles (p. ex., $\alpha_2 = 1$) e a escolha dos outros dois pesos é realizada de forma empírica. O limiar t_n que resultar no menor custo J é escolhido, ou seja,

$$\hat{t}_n = \underset{t_n}{\operatorname{argmin}} \{J(t_n)\}. \quad (4.16)$$

A máscara $m_n(i, j)$ é, então, definida e a separação em camadas para o macrobloco é realizada.

Para o macrobloco mostrado na Figura 4.8 (a), se $\alpha_1 = \alpha_2 = \alpha_3 = 1$, a função de custo para os l valores de t_n é mostrada na Figura 4.8 (b), e o FG, a máscara e o BG referentes ao limiar de menor custo ($\hat{t}_n = 36$) podem ser visualizados nas Figuras 4.8 (c), (d) e (e). Observe, ainda, que as regiões brancas dos planos mostrados nas Figuras 4.8 (c) e (e) não são necessárias à recomposição do documento original e, por isso, serão chamadas de “irrelevantes”. A manipulação dessas regiões consiste basicamente em preenchê-las de forma a aumentar a taxa de compressão dos planos a que pertencem. Esse preenchimento é realizado na etapa de pré-processamento do modelo MRC e recebe o nome de *data-filling*.

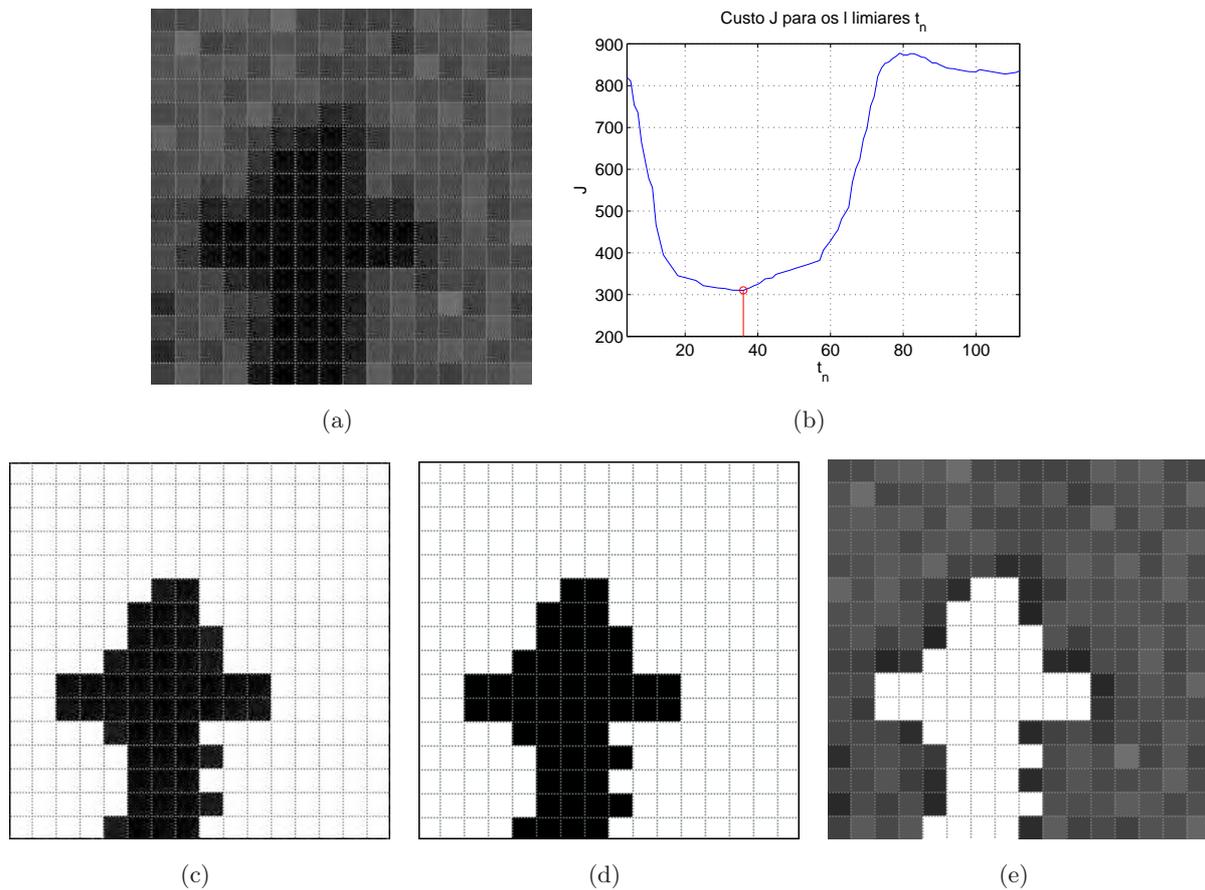


Figura 4.8: Exemplo de segmentação: (a) macrobloco original; (b) custo observado para cada um dos possíveis limiares t_n ; (c), (d) e (e) representam o FG, a máscara e o BG, respectivamente, para $t_n = 36$.

A escolha de um limiar de segmentação, conforme descrito anteriormente, é realizada para cada um dos n macroblocos do documento. A Figura 4.9 mostra a decomposição em camadas do documento “compound1”, antes do procedimento de *data-filling*.

4.3.2 Preenchimento de Regiões Irrelevantes (*Data-filling*)

Após a decomposição em camadas, *pixels* associados ao BG serão marcados como “irrelevantes” no FG. Da mesma forma, *pixels* associados ao FG serão marcados como “irrelevantes” no BG. Esses *pixels* podem ser substituídos por qualquer valor com o objetivo de se melhorar a compressão [102, 116, 117], conforme ilustrado na Figura 4.10. Existem vários métodos destinados a esse

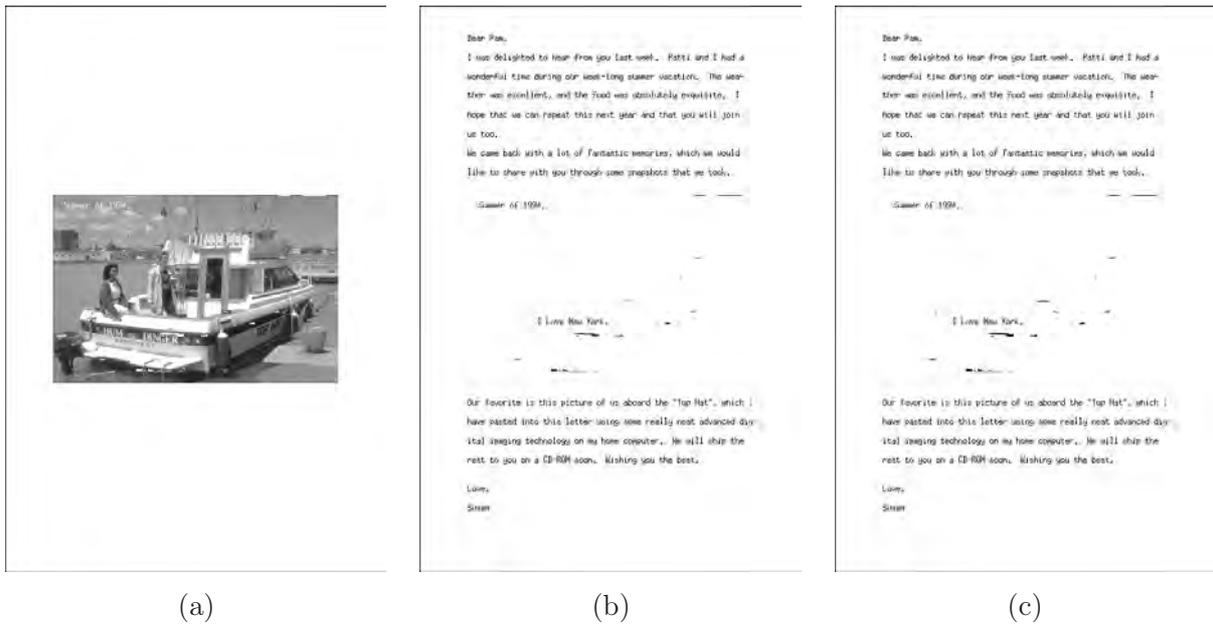


Figura 4.9: Decomposição em camadas do documento “compound1” antes do procedimento de *data-filling*: (a) BG; (b) máscara; e (c) FG.

fim. Tipicamente, as técnicas de *data-filling* procuram produzir transições suaves entre regiões relevantes e regiões irrelevantes, pois transições dessa natureza tendem a produzir menos *bits*, ao mesmo tempo em que geram menor distorção quando codificadores como o JPEG2000 e o AVC-I são empregados na compressão dos planos FG e BG.

4.3.2.1 Preenchimento por Valor Constante Igual à Média

A maneira mais simples de se realizar o *data-filling* é preencher as regiões irrelevantes com o valor da média dos *pixels* relevantes do mesmo plano. Para tanto, define-se inicialmente B e F como sendo as posições dos *pixels* onde a máscara indica BG e FG, respectivamente. Em seguida, calculam-se as médias,

$$\begin{aligned}
 m_{BG} &= \text{mean}(x(i, j) | (i, j) \in B) \\
 m_{FG} &= \text{mean}(x(i, j) | (i, j) \in F),
 \end{aligned}
 \tag{4.17}$$

onde $x(i, j)$ representa a imagem original e $\text{mean}(\cdot)$ é um operador que retorna a média aritmética do argumento. Por fim, as regiões irrelevantes são preenchidas segundo as equações abaixo,

$$\begin{aligned}
 BG_{df} &= M(i, j) \cdot BG(i, j) + m_{BG} \cdot (1 - M(i, j)) \\
 FG_{df} &= (1 - M(i, j)) \cdot FG(i, j) + m_{FG} \cdot (M(i, j)),
 \end{aligned}
 \tag{4.18}$$

onde BG_{df} e FG_{df} são as versões processadas de BG e FG , respectivamente. A Figura 4.11 mostra um trecho de imagem após o procedimento de *data-filling* descrito anteriormente.

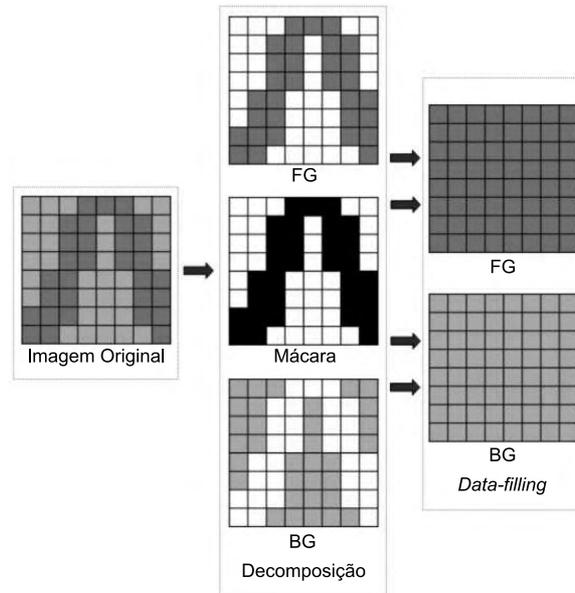


Figura 4.10: Diagrama que ilustra a decomposição MRC e o procedimento de *data-filling*.

4.3.2.2 Preenchimento Iterativo no Domínio Espacial

Uma outra solução para o problema do *data-filling* é o preenchimento iterativo de regiões irrelevantes, descrito a seguir.

Inicialmente o pré-processador recebe o macrobloco $x_n(i, j)$ e sua respectiva máscara binária $m_n(i, j)$. Em seguida a decomposição em camadas é realizada e os *pixels* do FG e do BG são marcados com o rótulo U , caso sejam relevantes, ou com o rótulo X , caso sejam irrelevantes. Se todos os *pixels* do plano recebem o rótulo X , conclui-se que toda a informação relevante está no plano oposto. Nessa situação o pré-processador retorna um macrobloco cujos *pixels* têm o valor da média do macrobloco anteriormente processado. Caso todos os *pixels* tenham recebido o rótulo U , o pré-processador retorna o macrobloco sem qualquer alteração, pois nessa situação há apenas regiões relevantes. Havendo uma mistura de rótulos U e X , em qualquer proporção, o pré-processador chama um procedimento que realiza o preenchimento iterativo das regiões X . Nesse procedimento, os *pixels* X que possuírem um ou mais vizinhos U em sua vizinhança horizontal e vertical têm seu valor substituído pela média dessa vizinhança. Isso é realizado de forma iterativa até que toda a região X tenha sido preenchida. A Figura 4.12 mostra as iterações desse algoritmo, quando aplicado ao plano BG de um macrobloco. Em cada etapa, as cores preta e branca indicam os *pixels* irrelevantes que estão sendo processados na etapa atual e os que serão processados em etapas futuras, respectivamente. Os demais níveis de cinza indicam *pixels* relevantes.

Na próxima seção será abordado um algoritmo de *data-filling* iterativo baseado na DCT.

4.3.2.3 Preenchimento Iterativo utilizando a DCT

O método de preenchimento de regiões irrelevantes descrito na presente seção envolve repetidas computações da DCT (*Discrete Cosine Transform*) até que um determinado critério de convergên-

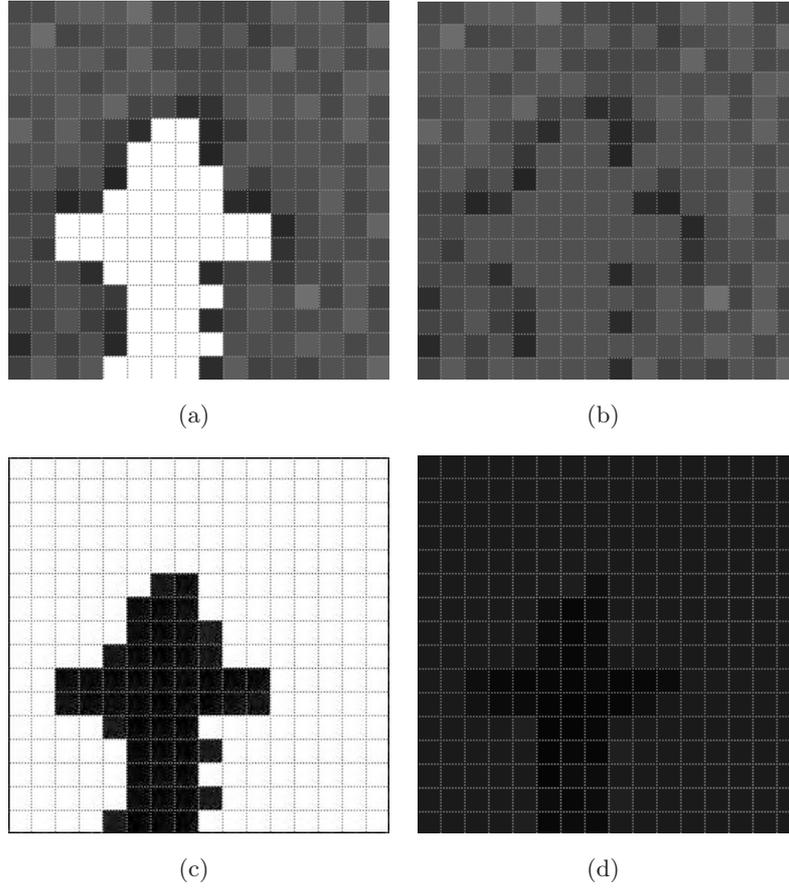


Figura 4.11: Exemplo de preenchimento por valor constante igual à média: (a) e (c) BG e FG; (b) e (d) BG_{df} e FG_{df} , versões processadas de BG e FG, respectivamente.

cia tenha sido satisfeito.

Seja I_0 a camada BG inicial do macrobloco, com os *pixels* irrelevantes substituídos por m_{BG} (média dos *pixels* relevantes do BG). Seja \tilde{I}_n a versão codificada e decodificada de I_n , utilizando-se um codificador a uma dada taxa de *bits* alvo. Se planejamos utilizar um codificador baseado na DCT, \tilde{I}_n pode ser aproximado por,

$$\tilde{I}_n = DCT^{-1}(\text{round}(DCT(I_n)/Q) * Q), \quad (4.19)$$

onde DCT e DCT^{-1} denotam a Transformada de Cosseno Discreta e sua respectiva inversa, $\text{round}(\cdot)$ é um operador que retorna o número inteiro mais próximo do argumento e Q é o passo de quantização dos coeficientes da transformada. Então, para $n = 0$ até $n = \nu$, onde ν limita o número de ciclos, computa-se,

$$I_{n+1}(i, j) = \begin{cases} I_n(i, j) & : (i, j) \in B \\ \tilde{I}_n(i, j) & : (i, j) \in F. \end{cases} \quad (4.20)$$

onde B e F são as posições dos *pixels* relevantes que pertencem ao BG e ao FG, respectivamente. O algoritmo é interrompido após ν ciclos ou quando,

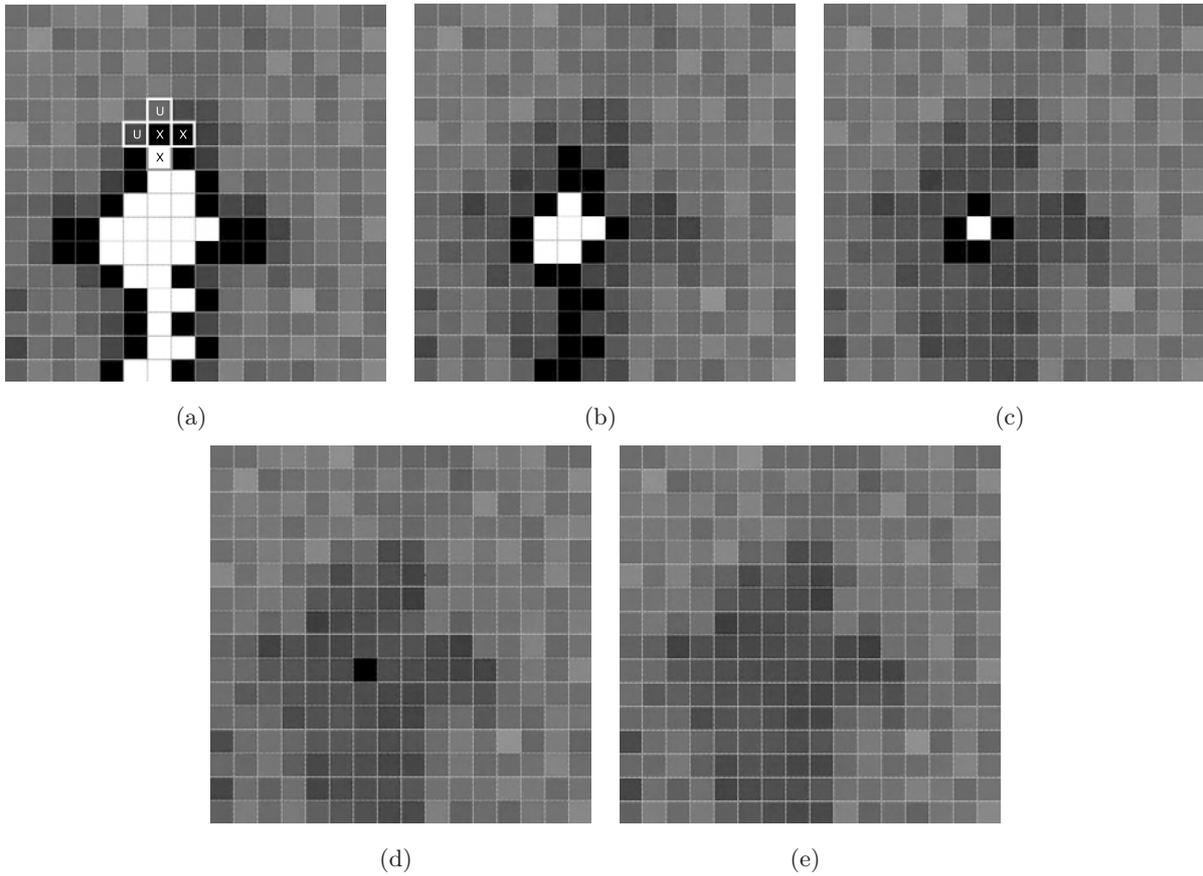


Figura 4.12: Exemplo de preenchimento iterativo no domínio espacial: em cada etapa, as cores preta e branca indicam os *pixels* irrelevantes que estão sendo processados na etapa atual e os que serão processados e etapas futuras, respectivamente.

$$\text{mean}(|I_n(i, j) - I_{n-1}(i, j)|) < \xi, \quad (i, j) \in F, \quad (4.21)$$

onde ξ é um limiar de tolerância. Em outras palavras, o algoritmo é interrompido quando o preenchimento das regiões irrelevantes converge. A Figura 4.13 mostra três iterações do algoritmo, para o BG de um macrobloco.

O mesmo processo é aplicado ao FG, substituindo-se a notação do BG pela do FG, e vice-versa. Alternativamente, pode-se utilizar a DWT (*Discrete Wavelet Transform*) no lugar da DCT e aplicar a transformada na imagem inteira em vez de aplicá-la macrobloco a macrobloco.

4.3.2.4 Preenchimento por Filtragem Segmentada

A idéia geral da técnica aqui exposta é preencher os *pixels* irrelevantes com um valor constante e procurar realizar a transição entre as regiões relevantes e irrelevantes por meio de filtragem.

Inicialmente, realiza-se um preenchimento conforme descrito na Seção 4.3.2.1, ou seja, os *pixels* X são substituídos pela média dos *pixels* U , gerando o plano $p(i, j)$. Em seguida $p(i, j)$ é filtrado por uma janela retangular (uniforme), por exemplo, gerando o plano filtrado $p_f(i, j)$. Por fim, os *pixels* U de $p_f(i, j)$ são restaurados a partir do plano original $p(i, j)$, conforme exemplificado

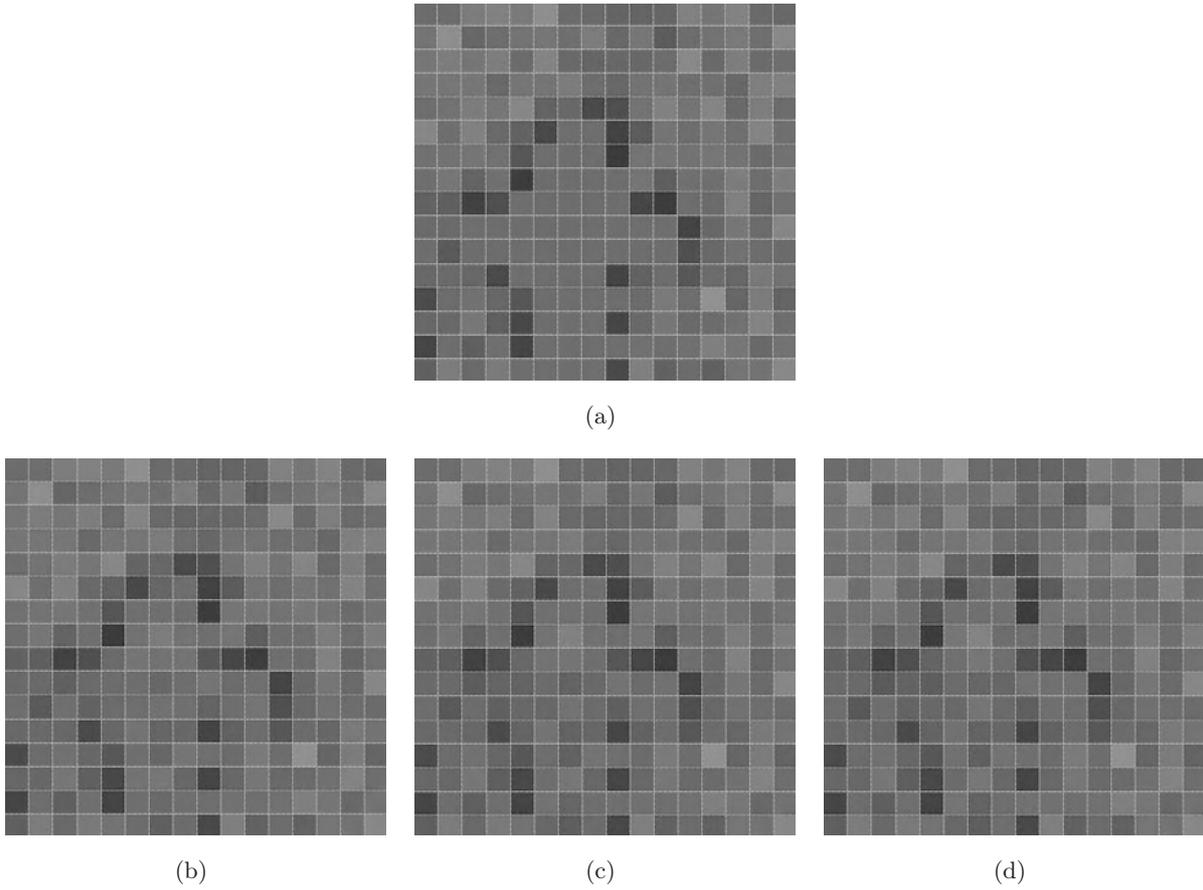


Figura 4.13: Exemplo de preenchimento iterativo utilizando a DCT: (a) plano inicial I_0 ; (b) iteração I_1 ; (c) iteração I_2 ; e (d) iteração I_3 .

nas Figuras 4.14.

É possível, porém, que, mesmo após a filtragem, algumas discontinuidades entre regiões U e X ainda permaneçam excessivamente acentuadas. Isso ocorre devido à simetria do processo de filtragem ao longo das transições. Uma solução para esse problema é a utilização da filtragem segmentada [116], onde pesos maiores são utilizados para ponderar *pixels* pertencentes à região relevante do plano. Ou seja, o plano filtrado $p_f(i, j)$ passa a ser dado por pela Equação 4.22.

$$p_f(i, j) = \frac{\sum_{u=-L}^L \sum_{v=-L}^L h_{i,j}(u, v) \cdot p(i + u, j + v)}{\sum_{u=-L}^L \sum_{v=-L}^L h_{i,j}(u, v)}, \quad (4.22)$$

onde $h_{i,j}(u, v)$ é um filtro variante ao deslocamento, com $2 \cdot L + 1$ linhas e $2 \cdot L + 1$ colunas. Dada a máscara $m(i, j)$, define-se,

$$h_{i,j}(u, v) = \begin{cases} 1 & : m(i + u, j + v) = 0, \text{ para } u, v = -L..L \\ K \cdot f(u, v) + 1 & : m(i + u, j + v) = 1, \text{ para } u, v = -L..L \end{cases}. \quad (4.23)$$

onde $f(u, v)$ é uma janela retangular, por exemplo. Uma melhor opção para $f(u, v)$, porém,

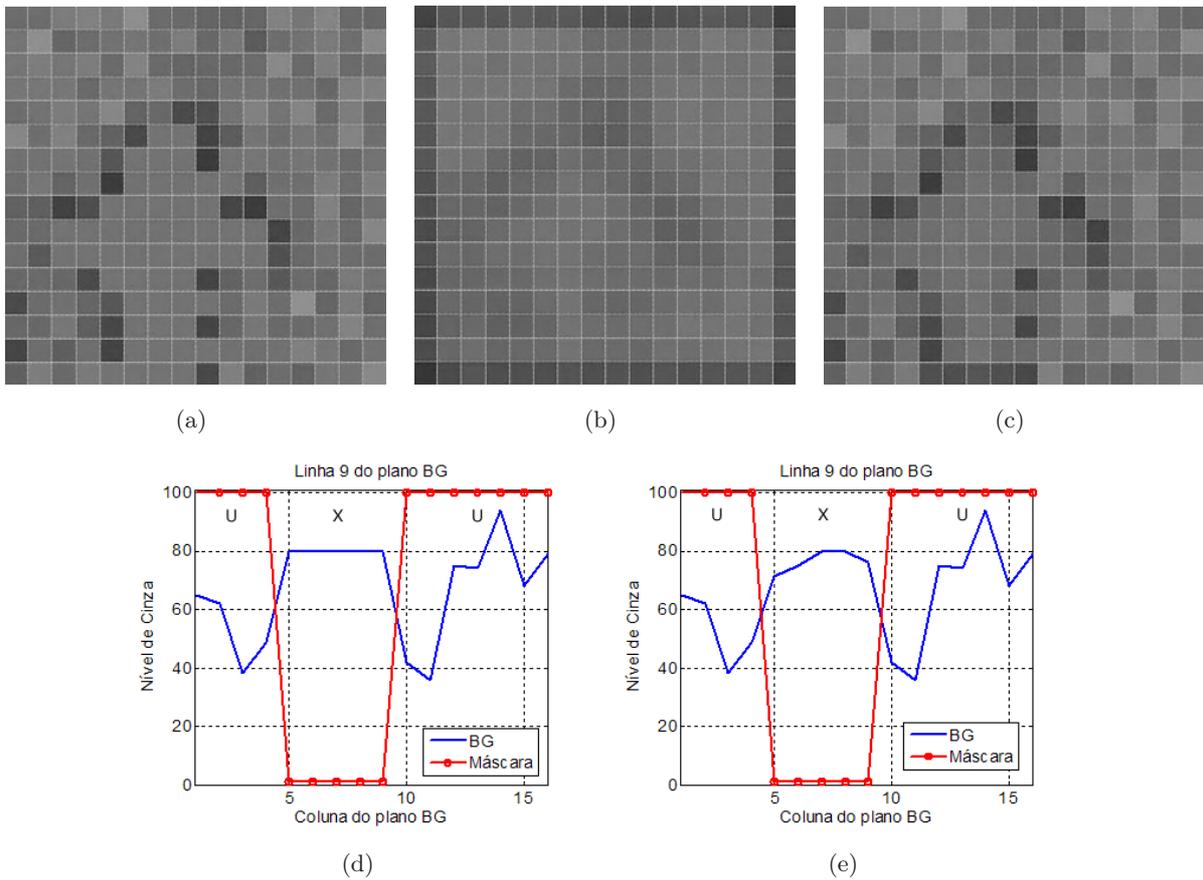


Figura 4.14: Exemplo de preenchimento por filtragem: (a) $p(i, j)$ (região X preenchida pela média da região U); (b) $p_f(i, j)$ (plano após filtragem por janela retangular 3×3); (c) plano $p_f(i, j)$ após restauração da região U ; (d) linha 9 da imagem (a); e (e) linha 9 da imagem (c). Observe como o processo de filtragem suavizou a transição entre as regiões X e U .

seria uma janela que desse menor ênfase a *pixels* relevantes mais distantes, como a triangular. As Equações 4.24 e 4.25 mostram exemplos de janela retangular $f_r(u, v)$ e triangular $f_t(u, v)$, respectivamente, ambas com 5 linhas e 5 colunas.

$$f_r(u, v) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (4.24)$$

$$f_t(u, v) = \begin{bmatrix} 1/9 & 2/9 & 1/3 & 2/9 & 1/9 \\ 2/9 & 4/9 & 2/3 & 4/9 & 2/9 \\ 1/3 & 2/3 & 1 & 2/3 & 1/3 \\ 2/9 & 4/9 & 2/3 & 4/9 & 2/9 \\ 1/9 & 2/9 & 1/3 & 2/9 & 1/9 \end{bmatrix} \quad (4.25)$$

Por fim, vale observar que quando $K = 0$, recai-se sobre o caso em que a filtragem não é segmentada. A Figura 4.15 ilustra a geração de uma das amostras $p_f(i, j)$, para $L = 1$, $K = 7$ e $f(u, v)$ retangular. A Figura 4.16 mostra o BG de um macrobloco antes e depois do *data-filling*.

A presente seção conclui a revisão da literatura técnica. Os quatro próximo capítulos serão destinados a apresentar os métodos propostos para compressão de documentos digitais.

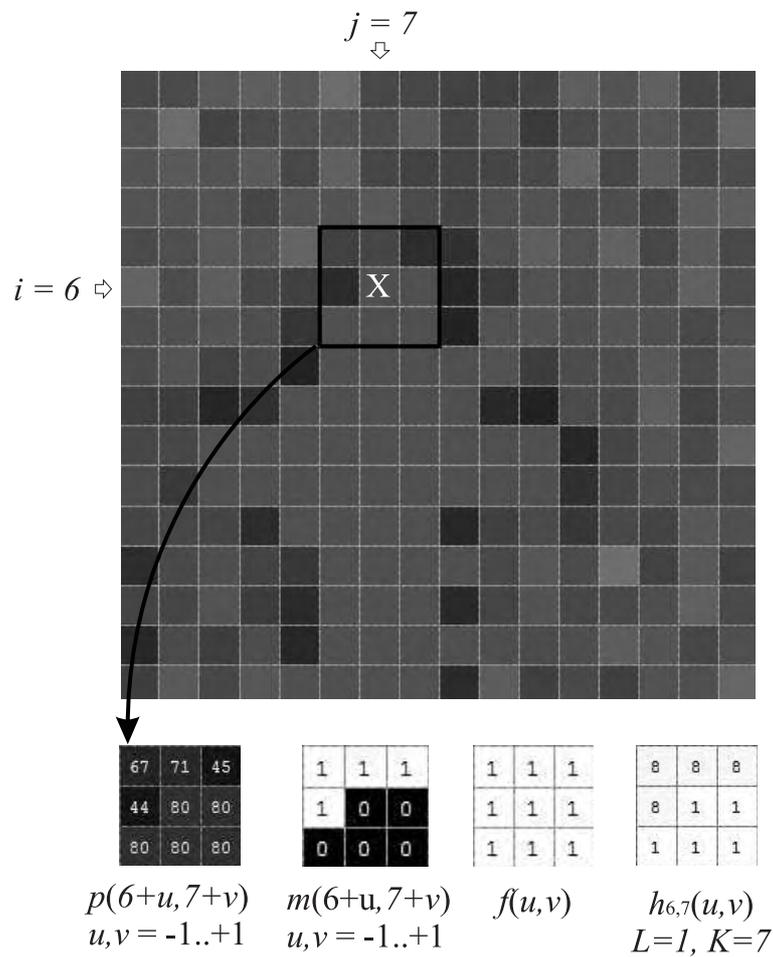
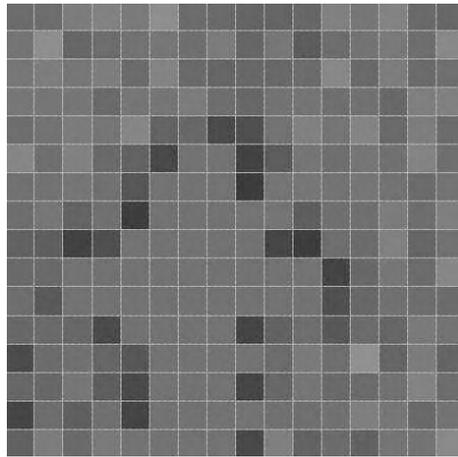
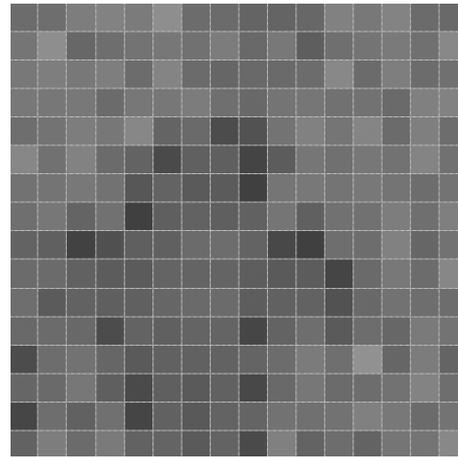


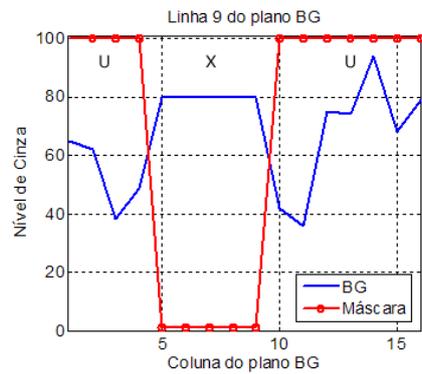
Figura 4.15: Exemplo de preenchimento utilizando filtragem segmentada. A amostra $X = p_f(6, 7)$ é determinada a partir da filtragem segmentada que utiliza o filtro variante ao deslocamento $h_{6,7}(u, v)$. No exemplo, $p_f(6, 7) = (8 \cdot (67 + 71 + 45 + 44) + 5 \cdot 80) / 37 = 60$. Para o melhor entendimento do exemplo, sugere-se a análise atenta das Equações 4.22 e 4.23.



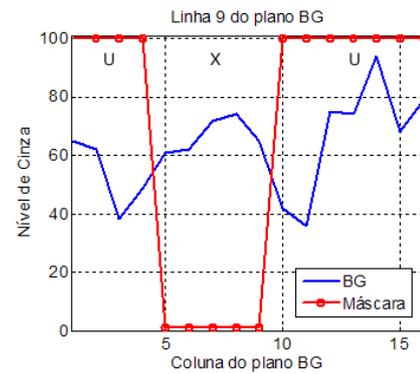
(a)



(b)



(c)



(d)

Figura 4.16: Exemplo de preenchimento por filtragem segmentada: (a) $p(i, j)$ (região X preenchida pela média da região U); (b) plano após filtragem segmentada por janela triangular 3×3 e restauração da região U ; (c) linha 9 da imagem (a); e (d) linha 9 da imagem (b). Observe como o processo de filtragem suavizou a transição entre as regiões X e U .

Capítulo 5

Método 1: Advanced Video Coding - Composto

5.1 Considerações Iniciais

O presente capítulo apresenta um método para compressão de documentos que utiliza a *abordagem de codificador único*. Tal método, que é uma das contribuições da presente tese, será denominado AVC-C (*Advanced Video Coding - Composto*), uma vez que se baseia no AVC-I (modo de operação intra do padrão H.264/AVC). De uma maneira geral, o codificador proposto ajusta-se dinamicamente à medida que a codificação do documento acontece, tratando de forma diferenciada as regiões pictóricas e de texto e gráfico.

5.2 Motivação e Contexto

É importante inserir o codificador proposto dentro do contexto adequado. Está sendo considerado um cenário distribuído, como a *Internet*, onde o documento é armazenado uma única vez e vários usuários o solicitam e decodificam em um momento diferente daquele em que ocorreu a codificação. Nesse cenário, o codificador não tem restrições de complexidade. De fato, como será mostrado posteriormente, um grande número de passagens é utilizado durante a codificação, fazendo com que o codificador proposto seja mais complexo do que aqueles que utilizam uma única passagem. Mesmo assim, a complexidade adicional não inviabiliza seu uso, uma vez que implementações rápidas podem ser utilizadas.

Assume-se, ainda, que o documento deva poder ser decodificado por decodificadores padrão, aos quais usuários comuns tenham acesso. O AVC-I não é um codificador originalmente proposto para ser aplicado à codificação de imagens estáticas, mas, uma vez que supera, em termos de taxa e distorção, codificadores considerados o estado-da-arte, como o JPEG2000, supõe-se um futuro em que tal codificador possa ser utilizado como base para a definição de um novo padrão de compressão de imagens.

5.3 Alocação de Taxa Baseada em Segmentação

O H.264/AVC apresenta duas características fundamentais que possibilitam a definição do codificador proposto:

1. O decodificador é capaz de lidar com a variação de QP , macrobloco a macrobloco; e
2. O valor do multiplicador de Lagrange λ , discutido no Capítulo 4, é calculado a partir do valor de QP , conforme a equação abaixo,

$$\lambda(QP) = 0,85 \cdot 2^{(QP-12)/3}. \quad (5.1)$$

O comportamento gráfico da Equação 5.1 pode ser visualizado na Figura 5.1. Para QP de 0 a 12, λ é menor que 1. Para QP de 13 a 51, λ passa a assumir valores maiores que 1. Nota-se, ainda, que a partir de $QP = 30$, aproximadamente, o crescimento da função torna-se mais acentuado.

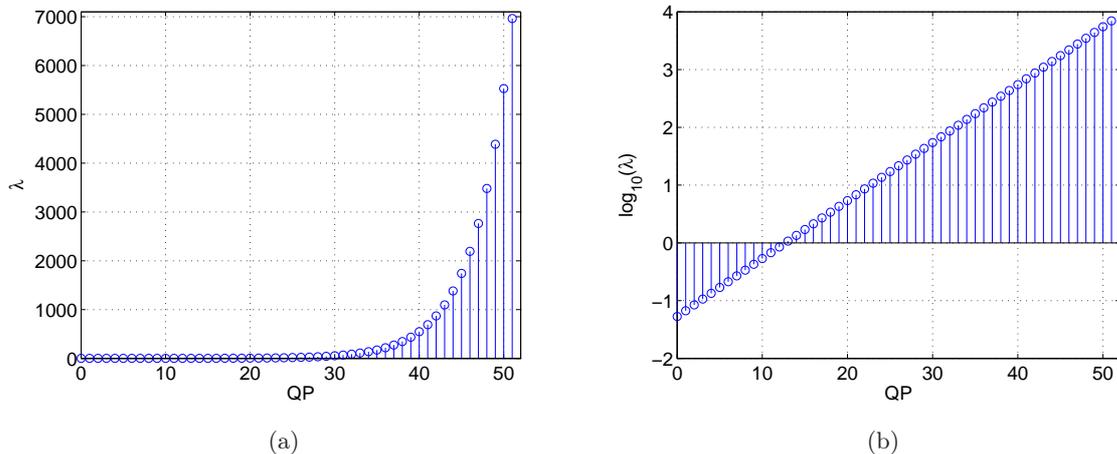


Figura 5.1: Comportamento gráfico da Equação 5.1. Menores valores de QP implicam menores valores de λ , e vice-versa. Para $0 \leq QP \leq 12$, $\lambda < 1$. Para $13 \leq QP \leq 51$, $\lambda > 1$. Em (a) e (b) λ é dado em escala linear e logarítmica, respectivamente.

No padrão há várias decisões que o codificador precisa tomar a cada macrobloco. Exemplos são a divisão dos macroblocos em blocos menores com o intuito de se realizar a estimação de movimento, a transformada a ser utilizada (4×4 ou 8×8) e o modo de predição intra. Tais decisões são todas realizadas por meio de uma análise de taxa e distorção. Considere novamente a função de custo definida conforme a Equação 4.4 ($J = D + \lambda R$), onde D e R representam a distorção e a taxa associada à codificação de um macrobloco, respectivamente. Essa equação determina que valores mais elevados de λ tendem a ajustar o codificador de forma a fazê-lo ser mais econômico em termos de taxa, dando menos importância à qualidade da imagem. No limite, quando $\lambda \rightarrow \infty$, a distorção torna-se até mesmo irrelevante, sendo a taxa o único fator a determinar a maneira como o macrobloco será codificado. Por outro lado, menores valores de λ fazem o codificador ser menos econômico em taxa, dando maior importância à qualidade da imagem. No limite, quando $\lambda \rightarrow 0$, a operação do codificador é determinada exclusivamente a partir do valor da distorção, sendo a

taxa irrelevante. A Figura 5.2 mostra um gráfico de D em função de R , onde esses dois pontos extremos de operação são ilustrados.

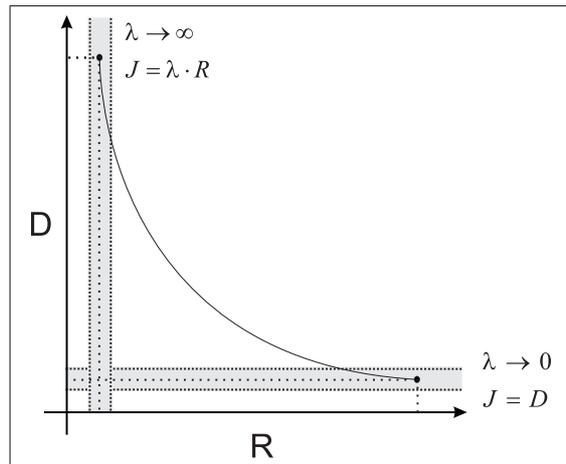


Figura 5.2: Relação entre a taxa R e a distorção D . Considerando a função de custo J , quando $\lambda \rightarrow \infty$, a distorção torna-se irrelevante e R é o único fator que determina a maneira como o macrobloco será codificado. Por outro lado, quando $\lambda \rightarrow 0$, a operação do codificador é determinada exclusivamente a partir do valor da distorção D , sendo a taxa irrelevante.

Da discussão acima, conclui-se que o ajuste do quantizador é a maneira mais eficiente de se controlar a taxa e a distorção, pois é possível alterar a computação da função de custo J simplesmente modificando-se o parâmetro de quantização QP para cada macrobloco n , ou seja,

$$J_n = D_n + R_n \cdot 0,85 \cdot 2^{(QP_n - 12)/3}. \quad (5.2)$$

O presente trabalho propõe a adaptação do QP com o intuito de tornar o codificador mais econômico em algumas regiões, em oposição a outras. Ou seja, em alguns macroblocos um menor número de coeficientes transformados é preservado, resultando na redução da taxa de *bits*. Essa economia de *bits* permite que em outros macroblocos um maior número de coeficientes transformados seja preservado, resultando no aumento da qualidade desses macroblocos. Essa proposta de adaptação evidencia a necessidade de se implementar uma etapa de classificação, a qual indicará ao codificador em quais macroblocos deve ocorrer economia de *bits* e em quais deve ocorrer melhoria de qualidade. Assume-se que o usuário é mais crítico em relação a degradações presentes nas bordas de texto do que nas demais regiões da imagem e, por isso, a classificação de macroblocos deve ser capaz de indicar ao codificador a localização de tais bordas. Desta maneira, a classificação de macroblocos deve ser precedida por uma etapa de segmentação de texto. Dado que o desenvolvimento de um segmentador não constitui o foco principal do método aqui proposto, pesquisou-se na literatura e utilizou-se, por conveniência, o algoritmo proposto por Fan [118]. Como exemplo, a Figura 5.3 (a) mostra um documento composto e a Figura 5.3 (b) mostra o resultado da segmentação.

O passo seguinte à segmentação é a classificação [119] dos macroblocos. A imagem binária, resultante da segmentação, é analisada e cada macrobloco é classificado em uma de três possíveis classes por meio do Algoritmo 1.

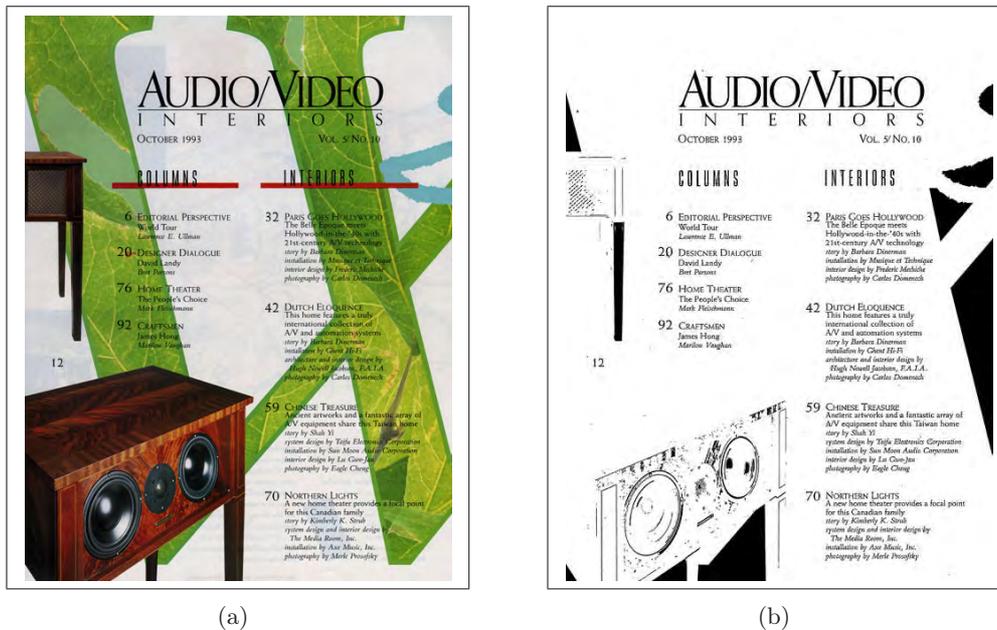


Figura 5.3: Segmentação: (a) exemplo de documento composto; (b) resultado da segmentação.

ALGORITMO 1

```

1  for  $n \leftarrow 1$  to  $NBlocos$  ▷ Para cada macrobloco.
2      do  $MediaMB \leftarrow media(macrobloco[n])$  ▷ Calcula a média do  $n$ -ésimo macrobloco.
3      if  $MediaMB = 0$ 
4          then  $TipoMB[n] \leftarrow 0$  ▷ O macrobloco  $n$  pertence à classe 0.
5      elseif  $MediaMB = 1$ 
6          then  $TipoMB[n] \leftarrow 1$  ▷ O macrobloco  $n$  pertence à classe 1.
7      else  $TipoMB[n] \leftarrow 2$  ▷ O macrobloco  $n$  pertence à classe 2.

```

As classes 0, 1 e 2, exemplificadas na Figura 5.4, são definidas a seguir.

- Classe 0 (*regiões pictóricas*): são compostas por pixels marcados com 1;
- Classe 1 (*regiões de interior de texto*): são compostas por pixels marcados com 0; e
- Classe 2 (*regiões de borda de texto*): apresentam uma mistura de pixels marcados com 0 e 1, em qualquer proporção.

A Figura 5.5 (a) mostra a máscara de classificação associada à imagem da Figura 5.3 (b). A Figura 5.5 (b) mostra uma parte ampliada da Figura 5.5 (a). Para facilitar a visualização e o entendimento do algoritmo, as classes 0, 1 e 2 foram representadas pelos níveis 0 (preto), 128 (cinza) e 255 (branco), respectivamente. A grade na Figura 5.5 (b) delimita os macroblocos. É importante observar que a segmentação de texto não é perfeita. Algumas regiões pictóricas acabam por serem classificadas como regiões de texto. Isso não é um problema maior, uma vez que o algoritmo proposto irá transferir *bits* de regiões pictóricas para regiões textuais. O efeito de uma segmentação imperfeita, como a verificada na Figura 5.5, é que algumas regiões pictóricas serão tratadas como borda de texto e terão sua qualidade desnecessariamente aumentada.

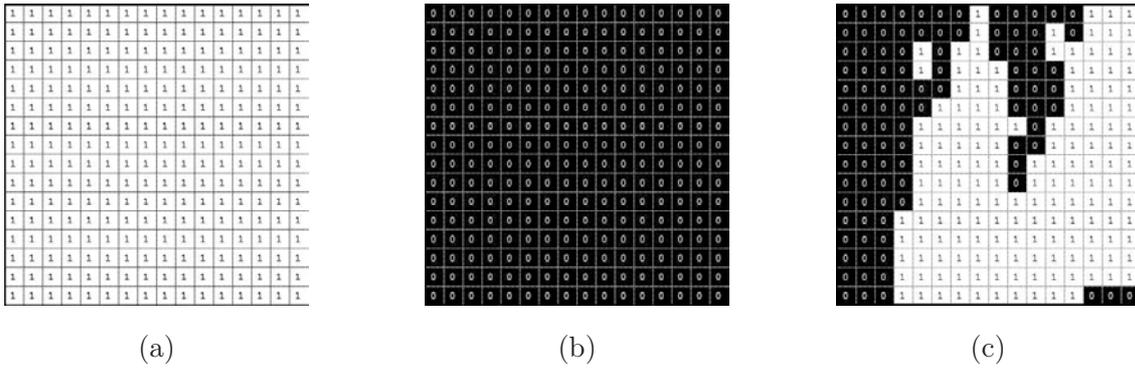


Figura 5.4: Exemplos das classes de macroblocos: (a) classe 0 (regiões pictóricas), (b) classe 1 (regiões de interior de texto) e (c) classe 2 (regiões de borda de texto).

A máscara de classificação é, então, passada a uma versão modificada do AVC-I, que adapta o valor de QP para cada macrobloco, de acordo com a classe a que pertencem. A idéia é economizar *bits* na codificação dos macroblocos pertencentes às classes 0 e 1 para aumentar a qualidade dos macroblocos pertencentes à classe 2. Para isso, macroblocos rotulados como 0 e 1 são codificados com um parâmetro de quantização QP , enquanto macroblocos rotulados como 2 são codificados com um outro parâmetro de quantização QP_{Texto} , sendo,

$$QP_{\text{Texto}} < QP. \tag{5.3}$$

Isso significa que a quantização irá zerar um maior número de coeficientes transformados em regiões pictóricas e de interior de texto do que em regiões onde ocorrem transições de caracteres. Esse algoritmo será chamado de *H.264/AVC-INTRA Composto*, ou simplesmente *AVC-C* [120], [121]. A Figura 5.6 ilustra a solução proposta. O objetivo é habilitar um único codificador H.264/AVC a comprimir eficientemente documentos compostos que possam ser decodificados por um decodificador H.264/AVC padrão.

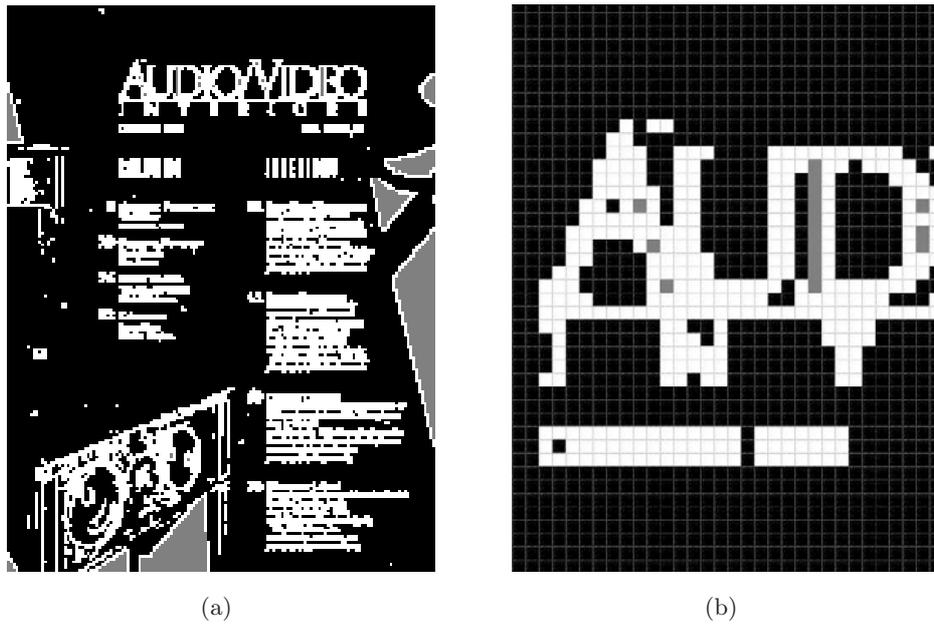


Figura 5.5: Máscara de classificação associada à imagem da Figura 5.3 (b). As classe 0 (pictóricas), 1 (interior de texto) e 2 (borda de texto) foram representadas pelos níveis 0 (preto), 128 (cinza) e 255 (branco), respectivamente.

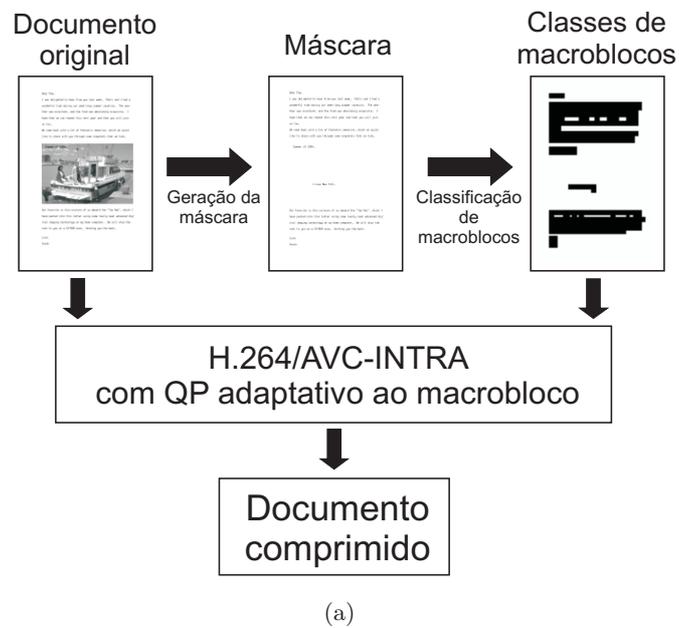


Figura 5.6: *H.264/AVC-INTRA - Composto* ou AVC-C: esquema do codificador proposto.

5.4 Algoritmo de Seleção de QP e QP_{Texto}

Alguns testes preliminares foram realizados com o intuito de se verificar o desempenho do codificador proposto. A Figura 5.7 mostra os resultados do AVC-C ajustado com (a) $QP = 0$ (melhor qualidade possível para as classes 0 e 1) e $QP_{\text{Texto}} = 51$ (pior qualidade possível para a classe 2) e (b) $QP = 51$ (pior qualidade possível para as classes 0 e 1) e $QP_{\text{Texto}} = 0$ (melhor qualidade possível para a classe 2).

Apesar de na Figura 5.7 (b) as bordas do texto terem sido preservadas, isso não é o suficiente. Deseja-se também manter a qualidade de regiões pictóricas em um nível razoável. Em outras palavras, deve-se reduzir a qualidade de regiões pictóricas com o objetivo de se melhorar a qualidade do texto, sem comprometer excessivamente a qualidade do documento como um todo. Os resultados apresentados nas Figuras 5.7 (a) e (b) apenas mostram que o AVC-C é capaz de se adaptar ao contexto, como esperado. Assim, o Algoritmo 2 foi proposto com a finalidade de se determinar a melhor escolha para QP e QP_{Texto} .

ALGORITMO 2

- 1 Codifica-se o documento com todas as possíveis combinações (QP, QP_{Texto}) , sendo $QP_{\text{Texto}} \leq QP$.
- 2 Uma taxa de *bits* alvo, R_a , é escolhida.
- 3 Uma variação de taxa δr ao redor de R_a é determinada.
- 4 Dentre todas as possíveis combinações de (QP, QP_{Texto}) ,
escolhe-se aquelas que apresentam taxas de *bits* contidas no intervalo $[R_a - \delta r, R_a + \delta r]$.
- 5 Dentre as combinações selecionadas, o máximo valor de PSNR, $PSNR_{\text{max}}$, é procurado.
- 6 Uma variação de PSNR, δq , é determinada,
e um valor mínimo de PSNR, $PSNR_{\text{min}} \leftarrow PSNR_{\text{max}} - \delta q$, é calculado.
- 7 Dentre as combinações selecionadas no passo 4,
aquelas com valores de PSNR maiores que $PSNR_{\text{min}}$ são escolhidas como candidatas.
- 8 A candidata que apresentar a maior valor $d = QP - QP_{\text{Texto}}$ é escolhida.

A seguir serão mostrados alguns resultados.



(a)



(b)

Figura 5.7: Teste preliminar realizado com o AVC-C: (a) $QP = 0$ (melhor qualidade possível para as classes 0 e 1) e $QP_{\text{Texto}} = 51$ (pior qualidade possível para a classe 2); e (b) $QP = 51$ (pior qualidade possível para as classes 0 e 1) e $QP_{\text{Texto}} = 0$ (melhor qualidade possível para a classe 2).

5.5 Resultados

O documento da Figura 5.8 foi comprimido com os codificadores AVC-C (QP adaptativo), AVC-I (QP fixo) e JPEG2000 (taxa de $bits$ fixa), utilizando-se diferentes configurações. Os resultados são mostrados nas Tabelas 5.1 (a) a (d). A Figura 5.9 mostra gráficos de desempenho taxa-distorção construídos a partir dos dados das tabelas. Note que a qualidade de regiões de texto pode ser consideravelmente aumentada enquanto a qualidade global não é demasiadamente afetada. A diferença entre os codificadores torna-se mais evidente quando o texto é visualizado em detalhe. As Figuras 5.10 (a) a 5.13 (a) mostram um exemplo de região textual ampliada do documento original “DSPG” e suas versões codificadas e reconstruídas por meio dos codificadores AVC-C, AVC-I e JPEG2000. Uma parte ampliada da região pictórica do mesmo documento também pode ser visualizada nas Figuras 5.10 (b) a 5.13 (b). Observe como a codificação utilizando o AVC-I e o JPEG2000 geram ruído ao redor do texto. Tal ruído não se apresenta na versão do documento codificada por meio do AVC-C.



Figura 5.8: Documento composto “DSPG”.

Tabela 5.1: Codificação de “DSPG”.

(a) AVC-C: $QP = 21$ e $QP_{\text{Texto}} = 03$. AVC-I: $QP = 20$. JPEG2000: $Taxa = 1,00$ bpp.

Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	43,54	43,24	67,10	1,05
AVC-I	44,11	44,00	46,76	1,01
JPEG2000	42,79	42,67	44,84	1,00

(b) AVC-C: $QP = 28$ e $QP_{\text{Texto}} = 02$. AVC-I: $QP = 25$. JPEG2000: $Taxa = 0,50$ bpp.

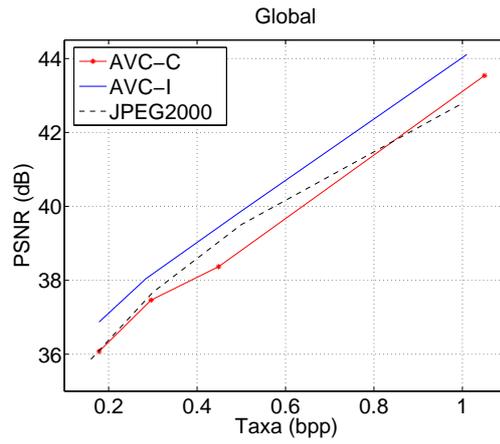
Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	38,37	38,07	70,56	0,45
AVC-I	39,81	39,87	39,13	0,49
JPEG2000	39,51	39,37	42,09	0,50

(c) AVC-C: $QP = 30$ e $QP_{\text{Texto}} = 11$. AVC-I: $QP = 28$. JPEG2000: $Taxa = 0,30$ bpp.

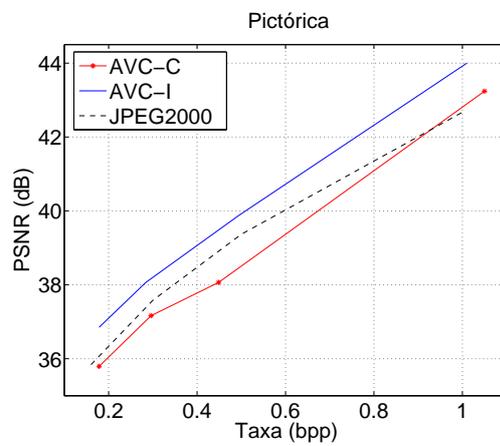
Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	37,46	37,17	54,36	0,30
AVC-I	38,04	38,07	37,71	0,28
JPEG2000	37,67	37,58	39,24	0,30

(d) AVC-C: $QP = 34$ e $QP_{\text{Texto}} = 17$. AVC-I: $QP = 31$. JPEG2000: $Taxa = 0,16$ bpp.

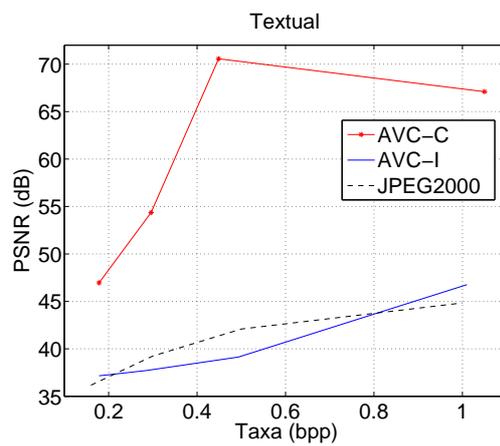
Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	36,08	35,80	46,96	0,17
AVC-I	36,87	36,85	37,16	0,18
JPEG2000	35,86	35,84	36,17	0,16



(a)



(b)

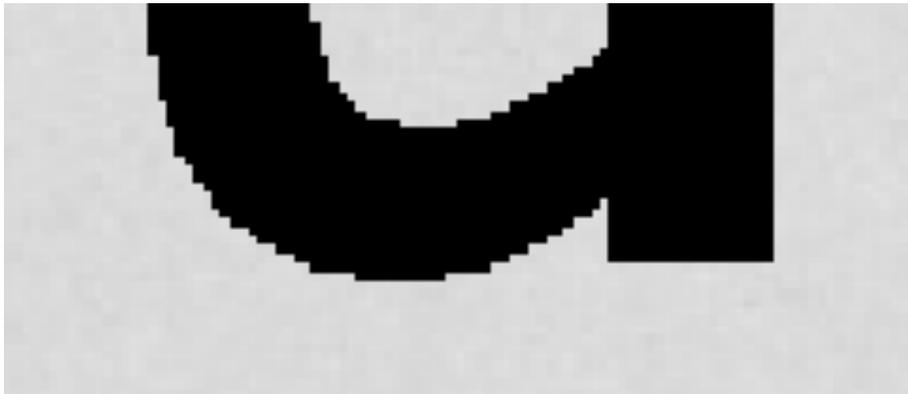


(c)

Figura 5.9: Comparação objetiva entre os codificadores AVC-C, AVC-I e JPEG2000 para o documento “DSPG”: (a) PSNR global; (b) PSNR de regiões pictóricas; e (c) PSNR de regiões de texto. Note que a qualidade de regiões de texto pode ser consideravelmente aumentada enquanto a qualidade global não é demasiadamente afetada.



(a)



(b)

Figura 5.10: Comparação subjetiva entre codificadores: (a) exemplo de região pictórica do documento original “DSPG”; e (b) exemplo de região textual do mesmo documento.



(a)



(b)

Figura 5.11: Comparação subjetiva entre codificadores. Documento “DSPG” codificado e reconstruído com o AVC-C a aproximadamente 0,16 bpp: (a) região pictórica; e (b) região textual.



(a)



(b)

Figura 5.12: Comparação subjetiva entre codificadores. Documento “DSPG” codificado e reconstruído com o AVC-I a aproximadamente 0,16 bpp: (a) região pictórica; e (b) região textual.



(a)



(b)

Figura 5.13: Comparação subjetiva entre codificadores. Documento “DSPG” codificado e reconstruído com o JPEG2000 a aproximadamente 0,16 bpp: (a) região pictórica; e (b) região textual.

As imagens das Figuras 5.14, 5.16, 5.18 e 5.20 também foram comprimidas com os codificadores AVC-C (QP adaptativo), AVC-I (QP fixo) e JPEG2000 (taxa de *bits* fixa), utilizando-se diferentes configurações. Os resultados são mostrados nas Tabelas 5.2, 5.3, 5.4 e 5.5. As Figuras 5.15, 5.17, 5.19 e 5.21 mostram os gráficos PSNR construídos a partir dos dados das tabelas.



Figura 5.14: Documento composto: “compound2”.

Tabela 5.2: Codificação de “compound2”.

(a) AVC-C: $QP = 27$ e $QP_{\text{Texto}} = 20$. AVC-I: $QP = 22$. JPEG2000: $Taxa = 1,07$ bpp.

Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	42,45	41,17	44,61	1,07
AVC-I	44,01	44,25	43,85	1,09
JPEG2000	40,39	41,1	39,82	1,07

(b) AVC-C: $QP = 38$ e $QP_{\text{Texto}} = 29$. AVC-I: $QP = 31$. JPEG2000: $Taxa = 0,57$ bpp.

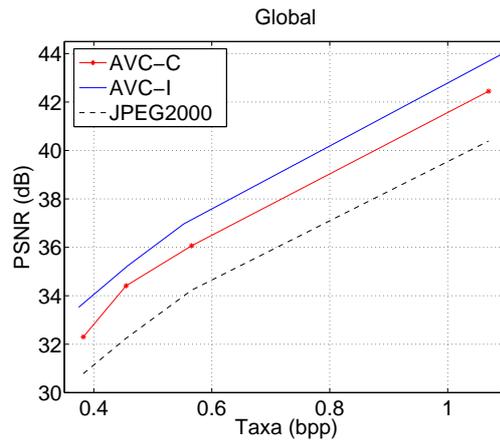
Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	36,06	35,36	37,11	0,57
AVC-I	36,97	38,9	35,62	0,55
JPEG2000	34,23	36,71	32,63	0,57

(c) AVC-C: $QP = 40$ e $QP_{\text{Texto}} = 32$. AVC-I: $QP = 33$. JPEG2000: $Taxa = 0,45$ bpp.

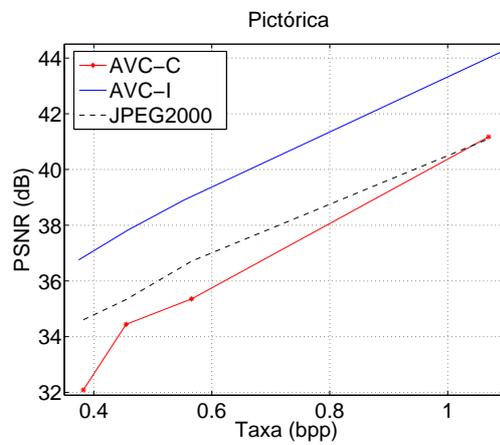
Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	34,41	34,44	34,48	0,45
AVC-I	35,23	37,81	33,58	0,46
JPEG2000	32,24	35,33	30,4	0,45

(d) AVC-C: $QP = 45$ e $QP_{\text{Texto}} = 34$. AVC-I: $QP = 35$. JPEG2000: $Taxa = 0,38$ bpp.

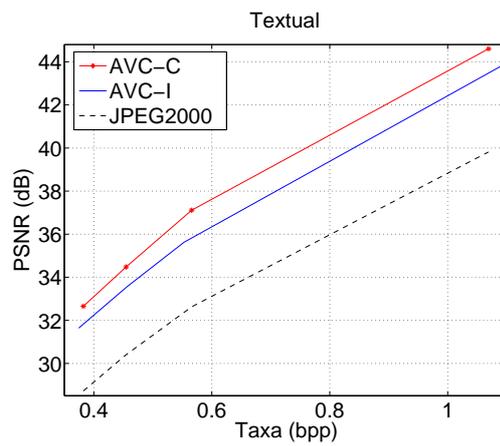
Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	32,3	32,09	32,65	0,38
AVC-I	33,53	36,75	31,64	0,37
JPEG2000	30,79	34,6	28,72	0,38



(a)



(b)



(c)

Figura 5.15: Comparação objetiva entre os codificadores AVC-C, AVC-I e JPEG2000 para o documento “compound2”: (a) PSNR global; (b) PSNR de regiões pictóricas; e (c) PSNR de regiões de texto.

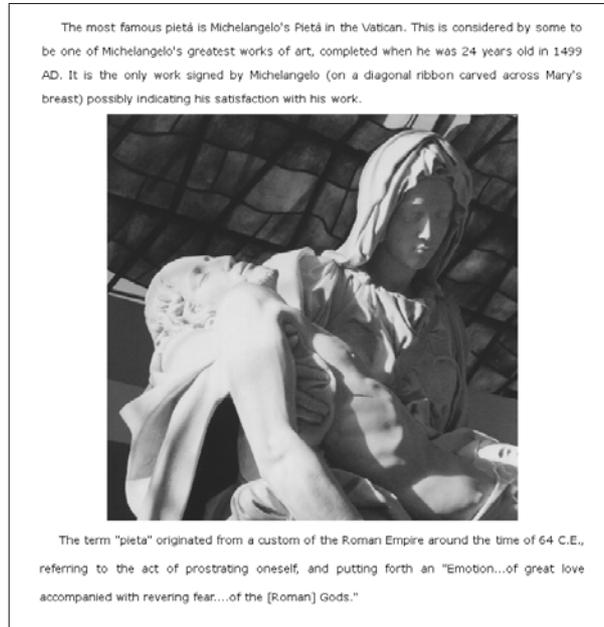


Figura 5.16: Documento composto: "pietà".

Tabela 5.3: Codificação de "pietà".

(a) AVC-C: $QP = 21$ e $QP_{Texto} = 18$. AVC-I: $QP = 20$. JPEG2000: $Taxa = 1,03$ bpp.

Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	45,34	44,44	51,66	1,05
AVC-I	45,88	45,11	50,42	1,03
JPEG2000	42,53	41,95	45,42	1,03

(b) AVC-C: $QP = 23$ e $QP_{Texto} = 20$. AVC-I: $QP = 22$. JPEG2000: $Taxa = 0,85$ bpp.

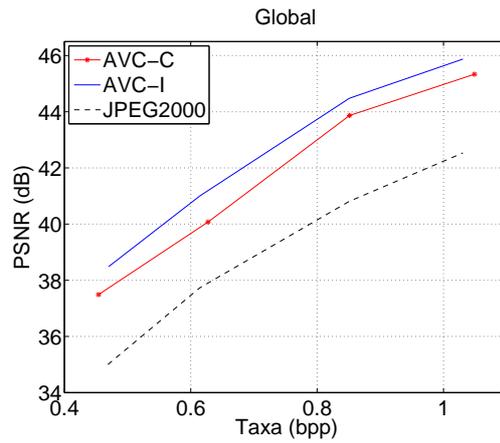
Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	43,87	42,97	50,35	0,85
AVC-I	44,48	43,73	48,8	0,85
JPEG2000	40,81	40,44	42,36	0,85

(c) AVC-C: $QP = 30$ e $QP_{Texto} = 20$. AVC-I: $QP = 26$. JPEG2000: $Taxa = 0,61$ bpp.

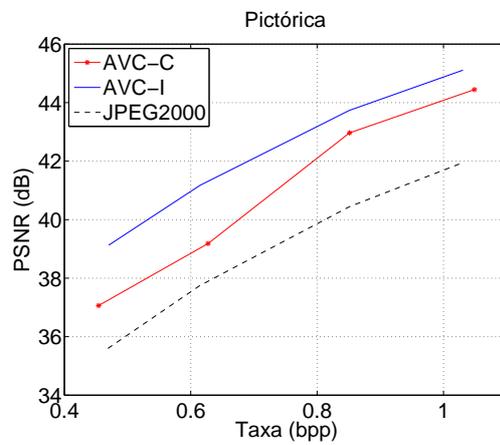
Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	40,07	39,18	46,4	0,63
AVC-I	41,00	41,17	40,49	0,61
JPEG2000	37,73	37,74	37,68	0,61

(d) AVC-C: $QP = 34$ e $QP_{Texto} = 26$. AVC-I: $QP = 30$. JPEG2000: $Taxa = 0,47$ bpp.

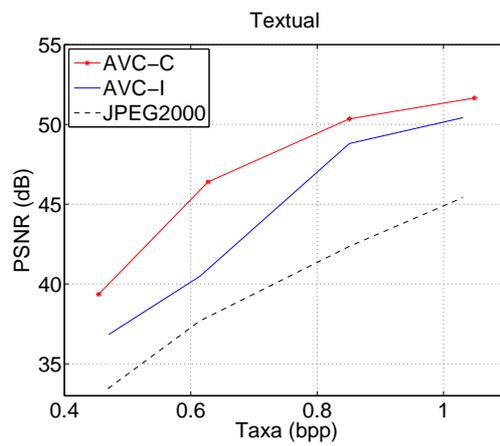
Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	37,49	37,06	39,35	0,45
AVC-I	38,49	39,13	36,84	0,47
JPEG2000	35,00	35,60	33,44	0,47



(a)



(b)



(c)

Figura 5.17: Comparação objetiva entre os codificadores AVC-C, AVC-I e JPEG2000 para o documento “pietà”: (a) PSNR global; (b) PSNR de regiões pictóricas; e (c) PSNR de regiões de texto.

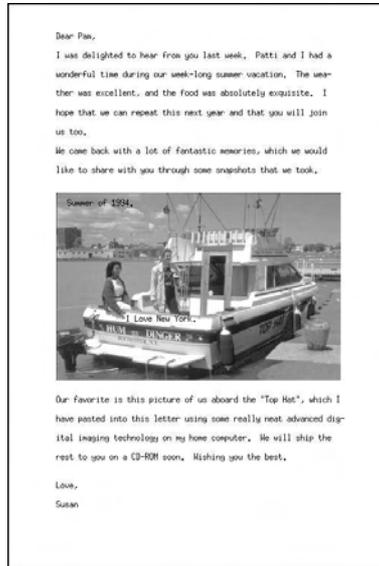


Figura 5.18: Documento composto: “compound1”.

Tabela 5.4: Codificação de “compound1”.

(a) AVC-C: $QP = 28$ e $QP_{T_{exto}} = 22$. AVC-I: $QP = 23$. JPEG2000: $Taxa = 0,89$ bpp.

Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	43,2	42,39	47,31	0,85
AVC-I	46,3	46,41	45,98	0,90
JPEG2000	40,01	40,4	39	0,89

(b) AVC-C: $QP = 30$ e $QP_{T_{exto}} = 23$. AVC-I: $QP = 26$. JPEG2000: $Taxa = 0,75$ bpp.

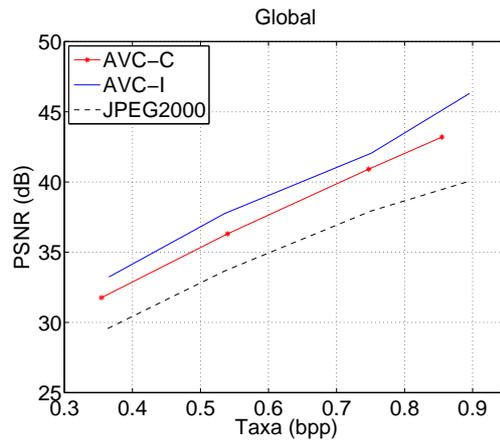
Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	40,91	40,73	41,49	0,75
AVC-I	42,05	43,94	38,92	0,75
JPEG2000	37,93	38,72	36,17	0,75

(c) AVC-C: $QP = 36$ e $QP_{T_{exto}} = 28$. AVC-I: $QP = 32$. JPEG2000: $tTaxa = 0,54$ bpp.

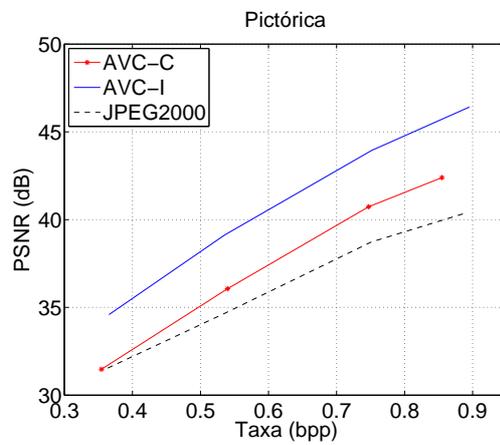
Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	36,31	36,07	37,13	0,54
AVC-I	37,75	39,12	35,16	0,54
JPEG2000	33,66	34,66	31,56	0,54

(d) AVC-C: $QP = 43$ e $QP_{T_{exto}} = 36$. AVC-I: $QP = 38$. JPEG2000: $Taxa = 0,36$ bpp.

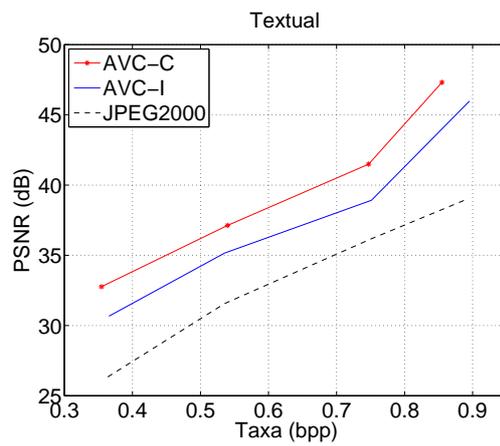
Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	31,76	31,48	32,77	0,35
AVC-I	33,24	34,59	30,67	0,37
JPEG2000	29,56	31,54	26,34	0,36



(a)



(b)



(c)

Figura 5.19: Comparação objetiva entre os codificadores AVC-C, AVC-I e JPEG2000 para o documento “pietà”: (a) PSNR global; (b) PSNR de regiões pictóricas; e (c) PSNR de regiões de texto.



Figura 5.20: Documento composto: “lennatexto”.

Tabela 5.5: Codificação de “lennatexto”.

(a) AVC-C: $QP = 31$ e $QP_{\text{Texto}} = 23$. AVC-I: $QP = 26$. JPEG2000: $Taxa = 0,91$ bpp.

Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	38,79	37,78	43,36	0,89
AVC-I	39,32	40,17	37,74	0,91
JPEG2000	36,85	37,16	36,16	0,91

(b) AVC-C: $QP = 34$ e $QP_{\text{Texto}} = 26$. AVC-I: $QP = 30$. JPEG2000: $Taxa = 0,71$ bpp.

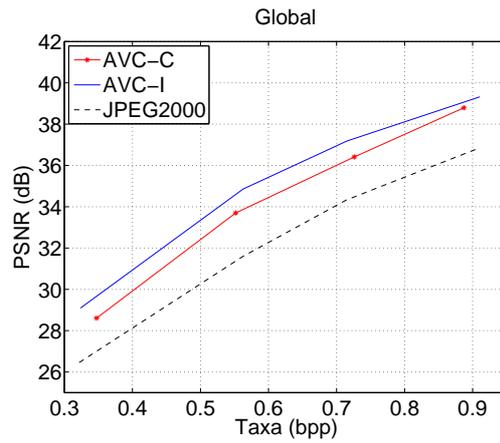
Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	36,41	35,98	37,72	0,73
AVC-I	37,18	38,18	35,39	0,72
JPEG2000	34,33	35,51	32,31	0,71

(c) AVC-C: $QP = 39$ e $QP_{\text{Texto}} = 32$. AVC-I: $QP = 34$. JPEG2000: $Taxa = 0,56$ bpp.

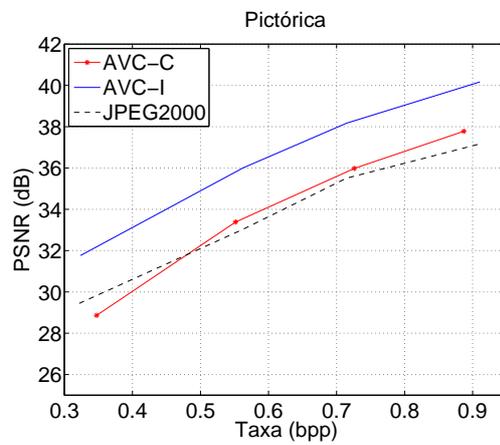
Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	33,7	33,39	34,59	0,55
AVC-I	34,86	36,01	32,88	0,56
JPEG2000	31,59	33,03	29,29	0,56

(d) AVC-C: $QP = 47$ e $QP_{\text{Texto}} = 40$. AVC-I: $QP = 42$. JPEG2000: $Taxa = 0,32$ bpp.

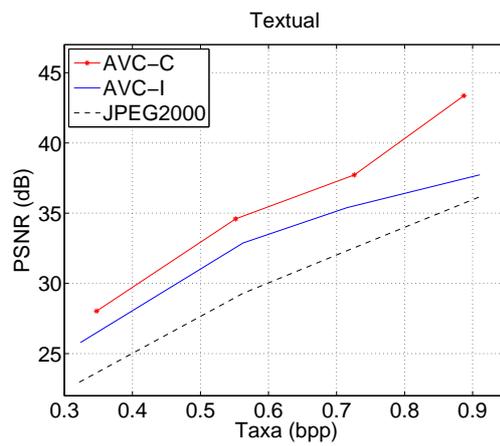
Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	28,61	28,86	28,02	0,35
AVC-I	29,09	31,76	25,79	0,32
JPEG2000	26,46	29,44	22,96	0,32



(a)



(b)



(c)

Figura 5.21: Comparação objetiva entre os codificadores AVC-C, AVC-I e JPEG2000 para o documento “lennatexto”: (a) PSNR global; (b) PSNR de regiões pictóricas; e (c) PSNR de regiões de texto.

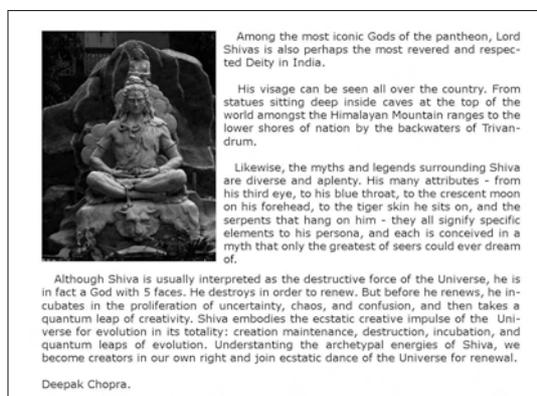


Figura 5.22: Documento composto: “shiva”.

Tabela 5.6: Codificação de “shiva”.

(a) AVC-C: $QP = 35$ e $QP_{T_{exto}} = 29$. AVC-I: $QP = 31$. JPEG2000: $Taxa = 0,99$ bpp.

Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	34,97	34,12	35,77	0,98
AVC-I	35,68	37,14	34,81	0,99
JPEG2000	31,46	31,98	31,09	0,99

(b) AVC-C: $QP = 42$ e $QP_{T_{exto}} = 35$. AVC-I: $QP = 37$. JPEG2000: $Taxa = 0,72$ bpp.

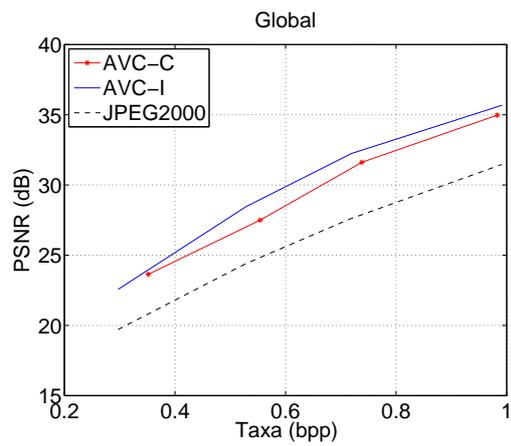
Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	31,61	30,12	33,26	0,74
AVC-I	32,24	33,08	31,68	0,72
JPEG2000	27,6	30,64	26,16	0,72

(c) AVC-C: $QP = 51$ e $QP_{T_{exto}} = 40$. AVC-I: $QP = 41$. JPEG2000: $Taxa = 0,53$ bpp.

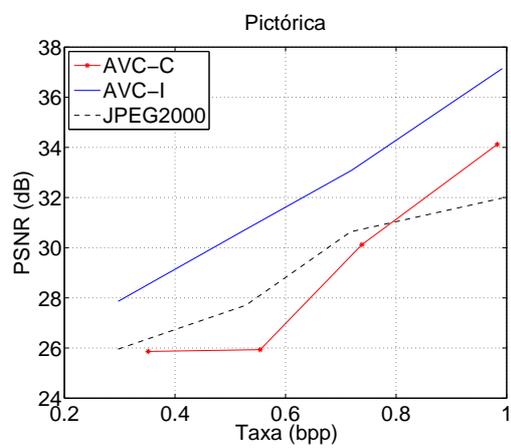
Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	27,5	25,93	29,29	0,55
AVC-I	28,46	30,74	27,26	0,53
JPEG2000	24,4	27,7	22,89	0,53

(d) AVC-C: $QP = 51$ e $QP_{T_{exto}} = 45$. AVC-I: $QP = 47$. JPEG2000: $Taxa = 0,30$ bpp.

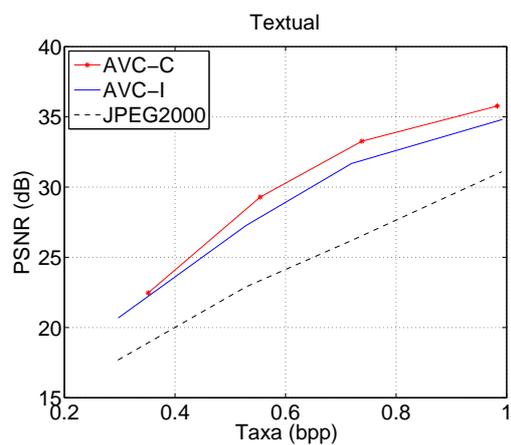
Resultado/ Algoritmo	PSNR Global (dB)	PSNR Pictórica (dB)	PSNR Textual (dB)	Taxa (bpp)
AVC-C	23,64	25,86	22,46	0,35
AVC-I	22,57	27,86	20,68	0,30
JPEG2000	19,71	25,95	17,68	0,30



(a)



(b)



(c)

Figura 5.23: Comparação objetiva entre os codificadores AVC-C, AVC-I e JPEG2000 para o documento “shiva”: (a) PSNR global; (b) PSNR de regiões pictóricas; e (c) PSNR de regiões de texto.

5.6 Codificação de Região de Interesse

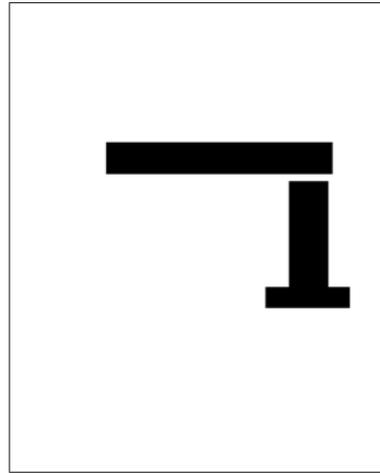
Alternativamente, o codificador proposto, AVC-C, pode ser utilizado para codificar uma região de interesse (ROI ou *Region of Interest*), funcionalidade que originalmente não é especificada no padrão H.264/AVC. Considere, por exemplo, a imagem da Figura 5.24 (a). Suponha que se deseje codificar a região em destaque com uma qualidade superior a das demais regiões. Se QP_{ROI} é o parâmetro de quantização aplicado à região em destaque e QP é o parâmetro de quantização aplicado ao resto da imagem, a codificação pode ser realizada passando-se ao AVC-C, além da imagem a ser codificada, a máscara mostrada na Figura 5.24 (b) e os parâmetros QP_{ROI} e QP , de tal forma que,

$$QP_{ROI} < QP. \quad (5.4)$$

A Figura 5.24 (c) mostra o resultado da codificação da imagem para $QP_{ROI} = 20$ e $QP = 51$. Nessas condições a taxa de bits observada foi de $0,37 \text{ bits/pixel}$. Note como a qualidade da região de interesse foi preservada, apesar da degradação geral observada. Se a imagem toda fosse codificada com $QP = 20$, com o objetivo de se preservar a região de interesse com a mesma qualidade, a taxa observada seria de $2,19 \text{ bits/pixel}$.



(a)



(b)



(c)

Figura 5.24: Codificação de região de interesse: (a) imagem original, com região de interesse em destaque; (b) máscara da região de interesse; e (c) image codificada com $QP_{ROI} = 20$ e $QP = 51$ ($0,37 \text{ bits/pixel}$). Note como a qualidade da região de interesse foi preservada, apesar da degradação geral da imagem.

Os experimentos mostraram que, para uma mesma taxa de *bits*, é possível aumentar a qualidade de regiões textuais enquanto a qualidade global da imagem não sofre danos excessivamente severos. As perdas em regiões pictóricas são tais que o AVC-C no geral supera objetivamente o JPEG2000. Apesar de não apresentar um ganho objetivo, quando comparado ao AVC-I, o algoritmo proposto permite codificar regiões de texto com alta qualidade, indicando uma possibilidade de ganho subjetivo superior, conforme exemplificado pelas Figuras 5.10 a 5.13.

Os resultados para os documentos 5.8 e 5.22 ilustram dois pontos de operação extremos do codificador. No documento 5.8 a quantidade de macroblocos de borda de texto é muito menor que a quantidade de macroblocos pictóricos, o que resulta num aumento acentuado da qualidade do texto. Já no documento 5.22, a situação se inverte. A quantidade de macroblocos de borda de texto é muito superior à quantidade de macroblocos pictóricos. Nessa situação não há como melhorar significativamente a qualidade do texto.

Além disso, o codificador AVC-C proposto é compatível com o decodificador AVC-I, ou seja, documentos codificados com o AVC-C podem ser decodificados por codificadores AVC-I sem qualquer prejuízo para o usuário. Uma outra característica interessante é que com AVC-C a compressão de regiões de interesse passa a estar habilitada dentro do arcabouço do H.264/AVC.

Capítulo 6

Método 2: Compressão MRC baseada em H.264/AVC-I e JBIG2 para Documentos Gerados Eletronicamente

6.1 Considerações Iniciais

O padrão MRC (*Mixed Raster Content*) [10] para compressão de documentos especifica uma representação multicamada para documentos compostos. Espera-se que melhores taxas de compressão possam ser alcançadas quando algoritmos de pré-processamento e padrões de compressão mais eficientes são utilizados [122, 123, 124]. O presente capítulo propõe um codificador baseado no modelo de imageamento MRC de três camadas. A segmentação em camadas é realizada conforme o algoritmo descrito na Seção 4.3.1 do Capítulo 4. A versão original desse algoritmo opera sobre blocos de tamanho 8×8 *pixels*. O presente trabalho generaliza seu uso, tornando-o apto a operar sobre blocos de tamanho qualquer. Quanto ao procedimento de *data-filling*, o método apresentado na Seção 6.2 do presente capítulo constitui uma outra contribuição dessa tese e leva em consideração que os planos FG e BG serão codificados com o AVC-I. Já a máscara binária é codificada com o JBIG2.

6.2 *Data-filling* Baseado nas Predições I16 do AVC-I

Uma vez que o AVC-I é utilizado na compressão das camadas FG e BG do documento, a presente seção propõe uma técnica de *data-filling* baseada nas predições I16 do referido codificador, descrita a seguir.

Seja BG_0 a camada BG inicial com seus *pixels* irrelevantes substituídos por m_{BG} (média dos *pixels* relevantes do BG). Considere BG'_0 como sendo a versão codificada e decodificada de BG_0 , utilizando-se o codificador AVC-I ajustado com o parâmetro de quantização QP desejado. Se BG_{pred} é a versão predita de BG'_0 (utilizando-se as predições I16 do AVC-I), então, BG após o procedimento de *data-filling* é dado por,

$$BG_{df}(i, j) = M(i, j) \cdot BG_0(i, j) + BG_{pred}(i, j) \cdot (1 - M(i, j)), \quad (6.1)$$

onde $M(i, j)$ representa a máscara binária obtida a partir do processo de segmentação em camadas. O FG é processado de forma análoga. Como exemplo, as Figuras 6.1 (a) a (e) mostram um exemplo de máscara binária, BG_0 , BG'_0 , BG_{pred} e BG_{df} para o BG de um macrobloco.

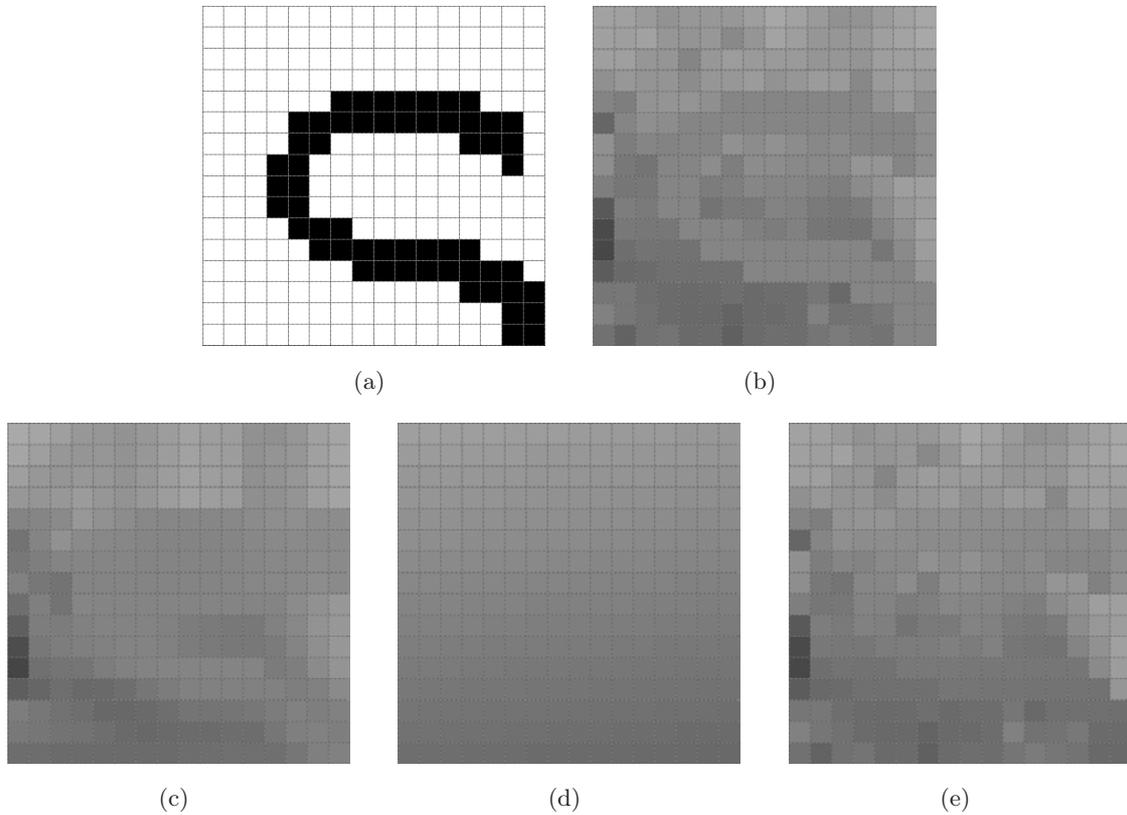


Figura 6.1: *Data-filling* baseado nas previsões $I16$ do AVC-I: (a) máscara binária; (b) BG_0 ; (c) BG'_0 ($QP = 30$); (d) BG_{pred} ; e (e) BG_{df} .

6.3 Codificação dos Planos

A última etapa do método proposto é a codificação dos planos. Propõe-se aplicar o AVC-I na compressão dos planos FG e BG, e o JBIG2 na compressão da máscara binária, conforme ilustrado na Fig 6.2.

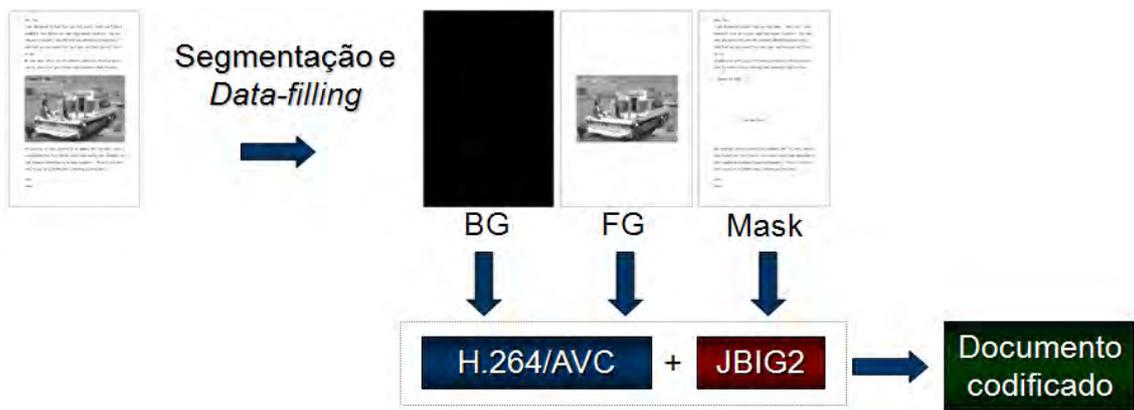


Figura 6.2: Codificador MRC proposto: propõe-se um codificador que utiliza o H.264/AVC na compressão dos planos FG e BG, e o JBIG2 na compressão da máscara binária.

6.4 Resultados

O documento “compound1”, mostrado na Figura 6.3 (a), foi comprimido com os codificadores AVC-I, JPEG2000 e o modelo MRC proposto. Na compressão MRC, a máscara binária mostrada na Figura 6.3 (b) foi codificada com o JBIG2 e as camadas FG e BG foram ambas codificadas com o JPEG2000 e com o AVC-I. A Figura 6.3 (c) mostra as curvas PSNR resultantes. Os resultados para esse documento mostram que o modelo MRC proposto, baseado no AVC-I, supera o modelo MRC baseado em JPEG2000 em mais de 5 dB a 1 bit/pixel . Em relação ao AVC-I e ao JPEG2000, o ganho é ainda mais expressivo, chegando a mais de 10 dB.

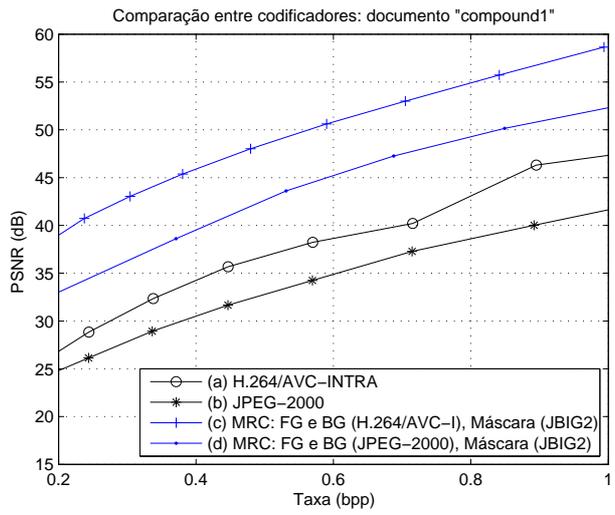
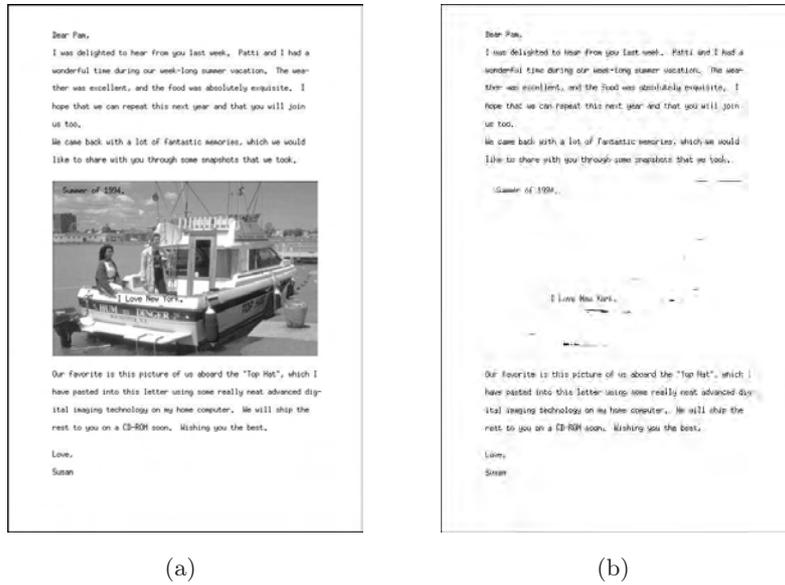
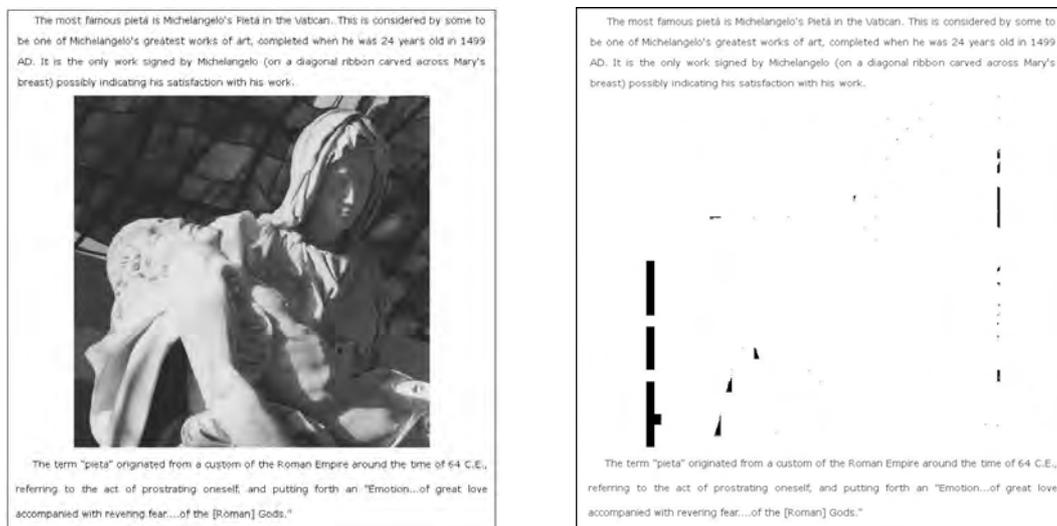


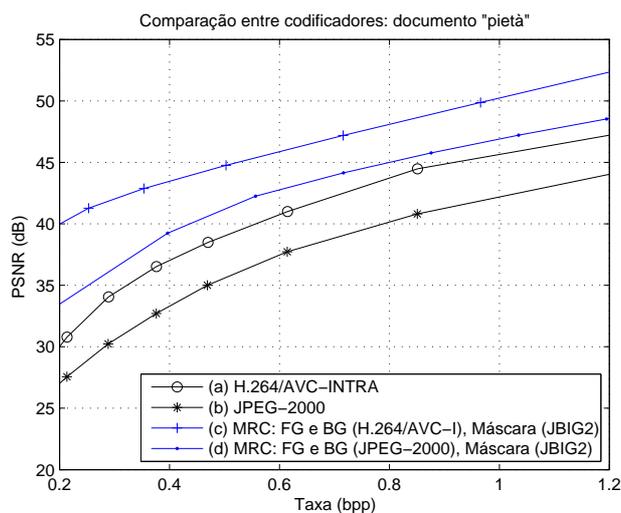
Figura 6.3: (a) Documento original: “compound1”; (b) máscara binária; (c) curvas PSNR para o documento “compound1”, comparando os codificadores AVC-I, JPEG2000, MRC: FG e BG codificados com AVC-I e máscara codificada com JBIG2, e MRC: FG e BG codificados com JPEG2000 e máscara codificada com JBIG2. Fatores de ponderação considerados no algoritmo de segmentação em camadas: $\alpha_1 = 100$, $\alpha_2 = 1$, $\alpha_3 = 40$.

Os documentos compostos originais mostrados nas Figuras 6.4 (a) a 6.9 (a) também foram comprimidos, utilizando-se os mesmos codificadores. Suas máscaras binárias e curvas PSNR podem ser observadas nas Figuras 6.4 (b) a 6.9 (b) e 6.4 (c) a 6.9 (c), respectivamente.



(a)

(b)



(c)

Figura 6.4: (a) Documento original: “pietà”; (b) máscara binária; (c) curvas PSNR para o documento: “pietà”. Fatores de ponderação considerados no algoritmo de segmentação em camadas: $\alpha_1 = 500$, $\alpha_2 = 1$, $\alpha_3 = 40$.

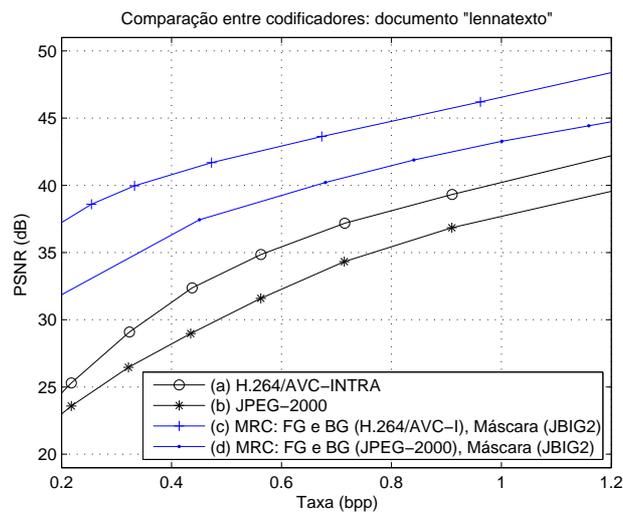
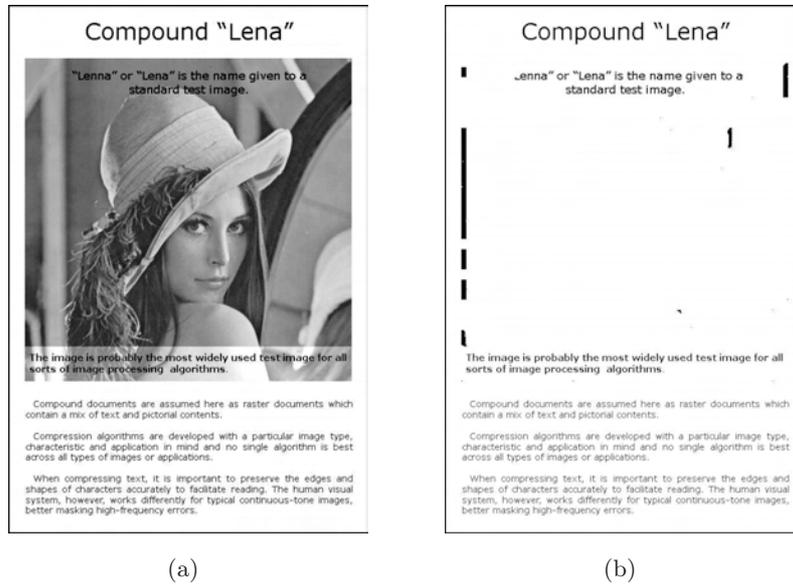
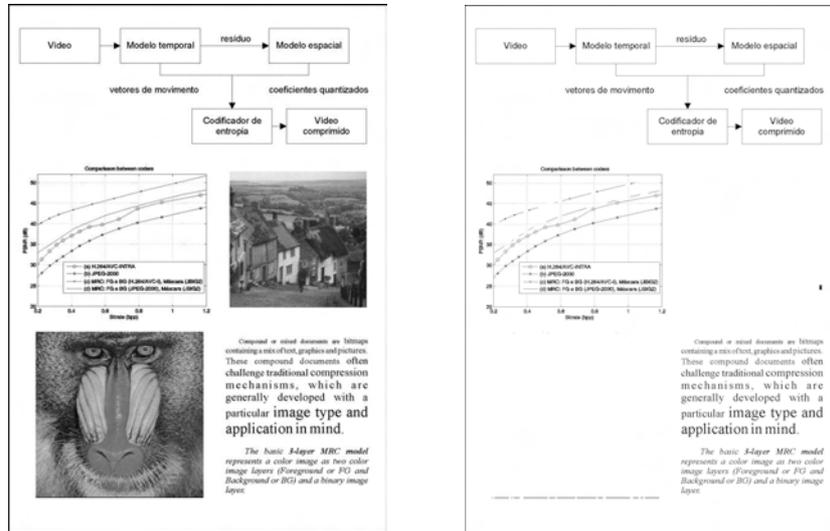
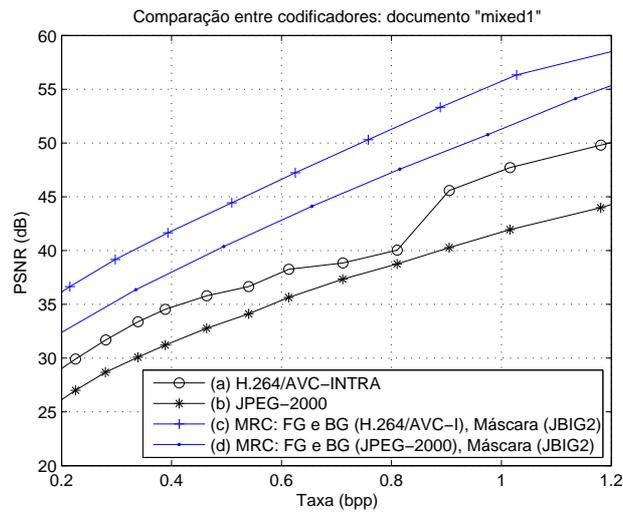


Figura 6.5: (a) Documento original: "lennatexto"; (b) máscara binária; (c) curvas PSNR para o documento: "lennatexto". Fatores de ponderação considerados no algoritmo de segmentação em camadas: $\alpha_1 = 300$, $\alpha_2 = 1$, $\alpha_3 = 40$.



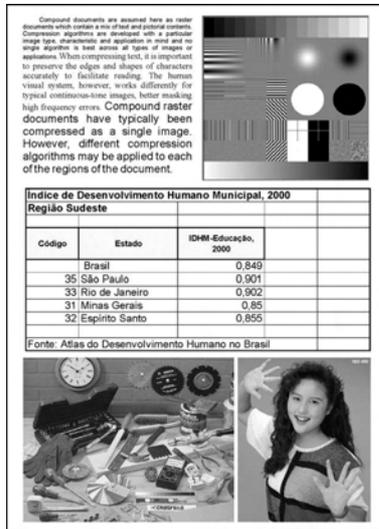
(a)

(b)

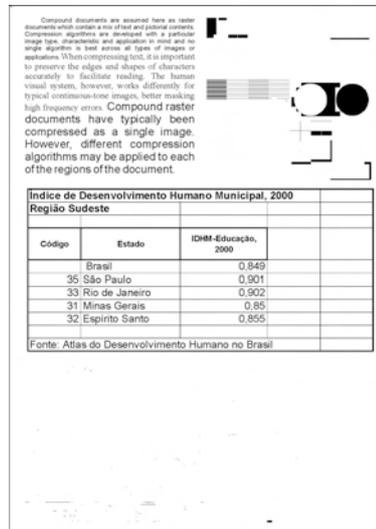


(c)

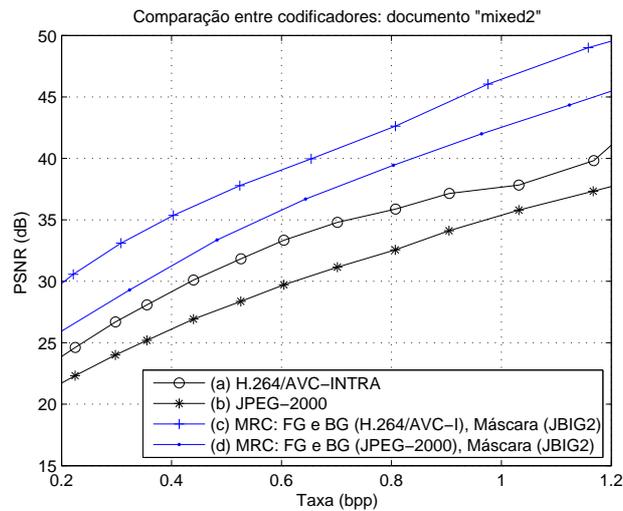
Figura 6.6: (a) Documento original: “mixed1”; (b) máscara binária; (c) curvas PSNR para o documento: “mixed1”. Fatores de ponderação considerados no algoritmo de segmentação em camadas: $\alpha_1 = 500$, $\alpha_2 = 1$, $\alpha_3 = 100$.



(a)

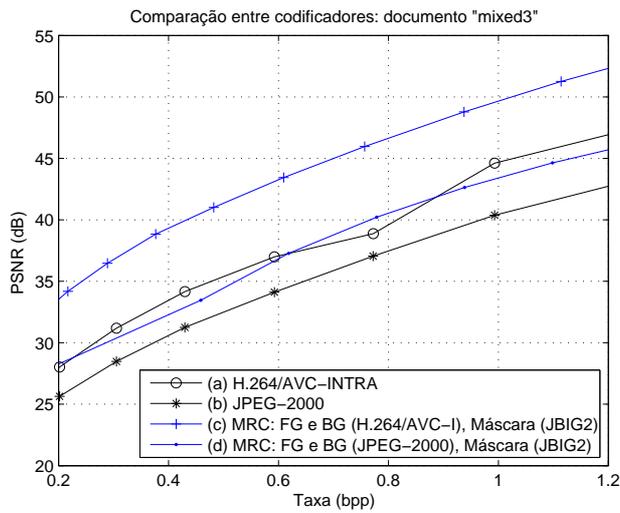
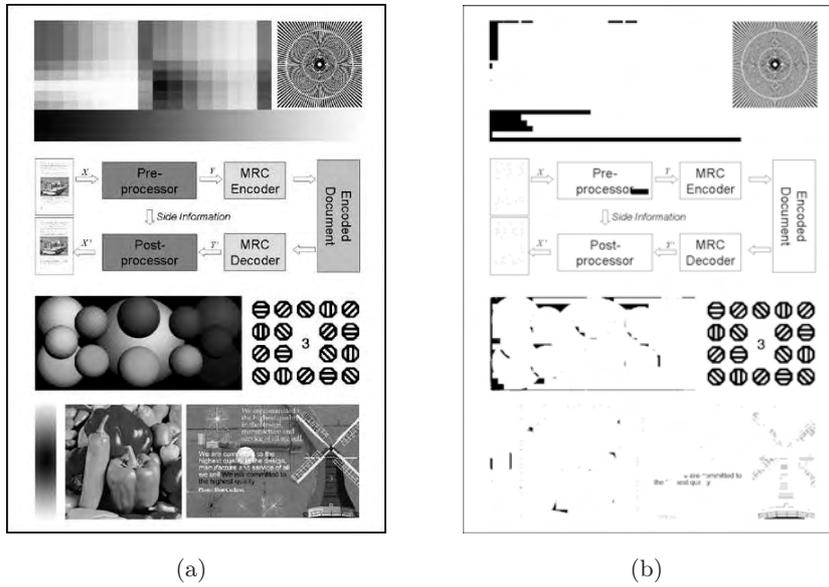


(b)



(c)

Figura 6.7: (a) Documento original: “mixed2”; (b) máscara binária; (c) curvas PSNR para o documento: “mixed2”. Fatores de ponderação considerados no algoritmo de segmentação em camadas: $\alpha_1 = 500$, $\alpha_2 = 1$, $\alpha_3 = 100$.



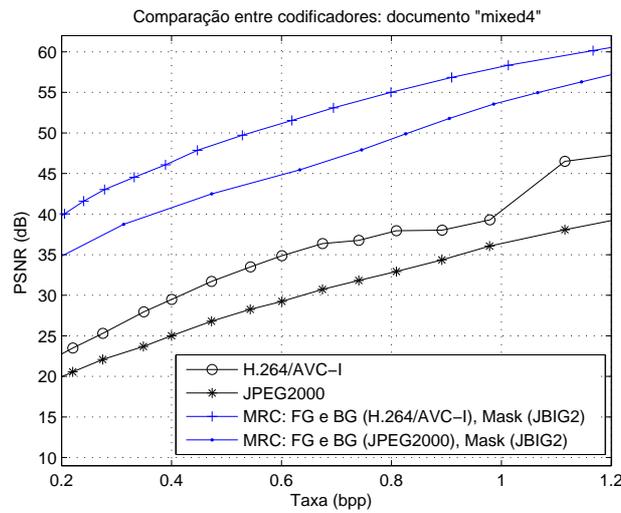
(c)

Figura 6.8: (a) Documento original: “mixed3”; (b) máscara binária; (c) curvas PSNR para o documento: “mixed3”. Fatores de ponderação considerados no algoritmo de segmentação em camadas: $\alpha_1 = 500$, $\alpha_2 = 1$, $\alpha_3 = 40$.



(a)

(b)



(c)

Figura 6.9: (a) Documento original: "mixed4"; (b) máscara binária; (c) curvas PSNR para o documento: "mixed4". Fatores de ponderação considerados no algoritmo de segmentação em camadas: $\alpha_1 = 500$, $\alpha_2 = 1$, $\alpha_3 = 40$.

As Figuras 6.10 (a) e (b) mostram partes ampliadas dos documentos originais “mixed2” e “mixed4”. Suas versões comprimidas e reconstruídas por meio dos CODEC’s JPEG2000 e AVC-I são mostradas nas Figuras 6.11 e 6.12, respectivamente. Os resultados gerados a partir dos codificadores MRC(AVC-I/JBIG2) e MRC(JPEG2000/JBIG2) são mostrados nas Figuras 6.13 e 6.14. Note que o esquema MRC(AVC-I/JBIG2) proposto elimina o efeito de borda ao redor de elementos textuais e gráficos, enquanto regiões pictoriais apresentam melhor qualidade. As taxas de *bits* alvo utilizadas na compressão foram $0,3 \text{ bits/pixel}$ para o documento “mixed2” e $0,2 \text{ bits/pixel}$ para o documento “mixed4”.



(a)

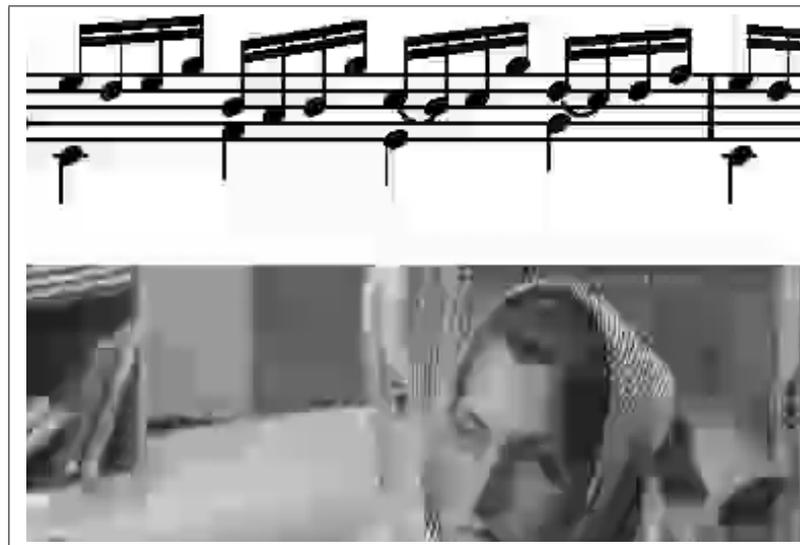


(b)

Figura 6.10: Comparação subjetiva entre codificadores. Parte ampliada dos documentos originais (a) “mixed2” e (b) “mixed4”.



(a)



(b)

Figura 6.11: Comparação subjetiva entre codificadores. Parte ampliada dos documentos codificados e reconstruídos com o AVC-I: (a) “mixed2” e (b) “mixed4”.



(a)



(b)

Figura 6.12: Comparação subjetiva entre codificadores. Parte ampliada dos documentos codificados com o JPEG2000: (a) “mixed2” e (b) “mixed4”.



(a)



(b)

Figura 6.13: Comparação subjetiva entre codificadores. Parte ampliada dos documentos codificados com o MRC (AVC-I/JBIG2), que é o modelo proposto: (a) “mixed2” e (b) “mixed4”.



(a)



(b)

Figura 6.14: Comparação subjetiva entre codificadores. Parte ampliada dos documentos codificados com o MRC (JPEG2000/JBIG2): (a) “mixed2” e (b) “mixed4”.

No presente capítulo foi proposto um codificador MRC de três camadas que utiliza o AVC-I na compressão do *Foreground* e do *Background*, e o JBIG2 na compressão da máscara binária. A compressão foi precedida por um pré-processamento que envolve a segmentação por limiarização de macroblocos (Seção 4.3.1) e o preenchimento de regiões irrelevantes baseada nas predições *I16* do H.264/AVC (Seção 6.2).

Conforme discutido no Capítulo 3, o AVC-I apresenta por si só uma capacidade extra em comprimir documentos compostos. Os ganhos em PSNR são substanciais, até mesmo ultrapassando a marca de 4 dB sobre o JPEG2000, em alguns casos. Apesar dessa capacidade extra do AVC-I, o modelo MRC baseado em múltiplos codificadores proposto oferece resultados que superam o AVC-I em até 10 dB a 1 *bit/pixel*. Além disso, os resultados obtidos também mostram que o modelo MRC sugerido, baseado no AVC-I, supera o modelo MRC baseado no JPEG2000 em até 5 dB a 1 *bit/pixel*. A análise em separado da qualidade objetiva de regiões de texto e de regiões pictóricas, como a realizada no Capítulo 5, torna-se secundária, uma vez que foi observado um aumento da qualidade objetiva do documento como um todo.

Quanto à qualidade subjetiva dos documentos codificados e reconstruídos, é possível observar que a solução proposta praticamente elimina o efeito de borda ao redor dos caracteres, ao mesmo tempo em que as regiões pictóricas apresentam qualidade superior.

Sem dúvida, o esquema proposto estabelece um novo patamar de desempenho objetivo e subjetivo no que se refere à compressão de documentos compostos. Há, porém, uma ressalva: o modelo é bastante eficiente para representar elementos com bordas abruptas sobre um fundo qualquer. No entanto, uma vez que a máscara é binária, torna-se difícil lidar com documentos digitalizados e elementos com bordas suaves. Assim, o Capítulo 7 propõe um método para habilitar a compressão de documentos digitalizados utilizando o modelo MRC baseado em H.264/AVC e JBIG2.

Capítulo 7

Método 3: Compressão MRC baseada em H.264/AVC-I e JBIG2 para Documentos Digitalizados

7.1 Considerações Iniciais

O presente capítulo propõe um método para compressão MRC de documentos digitalizados¹ [125]. Em documentos desse tipo, as bordas de caracteres e elementos gráficos não pertencem completamente ao FG ou ao BG, e a segmentação em planos gera uma fronteira pronunciada ao redor das transições de caracteres, aqui chamada de “halo”². A proposta é construir um *mapa de aguçamento* de bordas e parametrizar, no codificador, a suavidade original do documento. No decodificador, o mapa de aguçamento e os parâmetros de suavidade são utilizados na ressuavização das bordas, anteriormente aguçadas no codificador. Apesar de subjetivamente alguns usuários preferirem imagens com bordas aguçadas, assume-se que o documento codificado e reconstruído deve ser o mais próximo possível do documento digitalizado original, mantendo, portanto, as características de suavidade de suas bordas.

Em outras palavras, o que está sendo proposto, e que constitui uma contribuição da presente tese, é um método de pré e pós-processamento para as camadas MRC de documentos digitalizados, aliado à codificação baseada no H.264/AVC e no JBIG2. A Figura 7.1 descreve o arcabouço onde a solução está inserida. Inicialmente, o documento original digitalizado X é apresentado ao pré-processador, que devolve uma representação MRC de três camadas Y . Além disso, o pré-processador aguça as bordas, realiza o *data-filling*, constrói um mapa de aguçamento e parametriza a suavidade original das bordas. Ambos, o mapa de aguçamento e os parâmetros de suavidade são enviados ao decodificador como informação lateral. As camadas são, então, codificadas segundo

¹Na presente tese o termo “digitalizado” refere-se à condição de ter sido capturado para o meio digital com o auxílio de um *scanner*.

²Halo, do grego, *halos*, substantivo masculino: fenômeno caracterizado pela presença de um círculo luminoso que às vezes se nota ao redor de fontes luminosas. No sentido figurado é utilizado aqui como referência ao contorno pronunciado que se observa ao redor das fronteiras que definem os caracteres, quando o procedimento de segmentação em camadas é aplicado a documentos digitalizados.

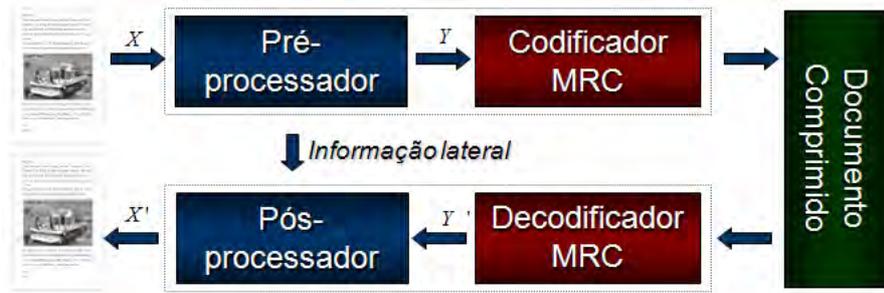


Figura 7.1: Ilustração das etapas de pré-processamento, codificação, decodificação e pós-processamento.

o modelo MRC ilustrado na Figura 6.2 da Seção 6.3 (Capítulo 6). No decodificador, as versões codificadas das camadas são decodificadas pelo decodificador MRC, resultando em Y' , que, junto com a informação lateral é utilizado pelo pós-processador para compor a versão reconstruída X' do documento. Visto que a decomposição em camadas não constitui o foco do presente capítulo, utilizou-se o algoritmo de segmentação proposto na Seção 4.3.1 do Capítulo 4.

7.2 O Problema com Transições Suaves

No modelo MRC, os elementos de texto têm, tipicamente, o mesmo formato que o observado na máscara. Quando a imagem é digitalizada, as bordas de texto e gráficos não são aguçadas, mas sim suaves. Uma vez que o plano seletor (máscara) é binário e as transições são suaves, não é possível transferir por completo o BG para um dos planos e o FG para outro plano. Uma ilustração é mostrada na Figura 7.2, onde as transições suaves são destacadas nas Figuras 7.2 (a) e (c), junto com a máscara (linha pontilhada). A máscara determina o que é BG e o que é FG. Note que a transição está incluída nos dois planos. Assim, se no FG as partes correspondentes ao BG forem substituídas pela média do FG, como uma forma simplificada de *data-filling* (ver Seção 4.3.2.1 do Capítulo 4), alguns “picos indesejados” aparecem, como ilustrado na Figura 7.2 (b). A Figura 7.2 (d) mostra o mesmo efeito no BG. Observa-se, então, o aparecimento de um halo ao redor do texto, conforme ilustrado nas Figuras 7.3 (a) e (b), o que acaba sendo bastante prejudicial para a taxa de compressão. Além disso, uma vez que tal efeito está contido em regiões relevantes, não é possível removê-lo via técnicas de *data-filling*.

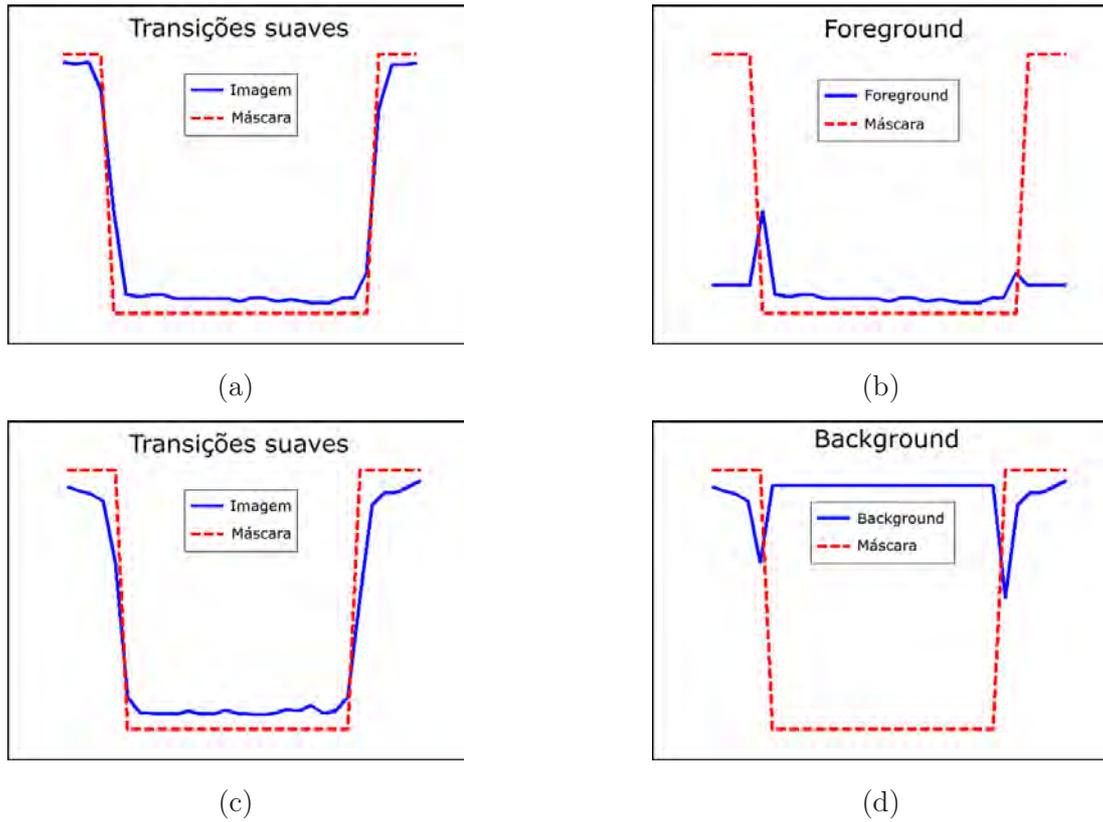


Figura 7.2: Exemplos em uma dimensão do efeito da segmentação em transições suaves. Em (a) e (c) duas transições e suas respectivas máscaras, indicadas em linhas pontilhadas, podem ser observadas. No planos (b) FG e (d) BG, as respectivas regiões irrelevantes são substituídas por valores constantes, enquanto parte das transições não, causando os picos indesejáveis.

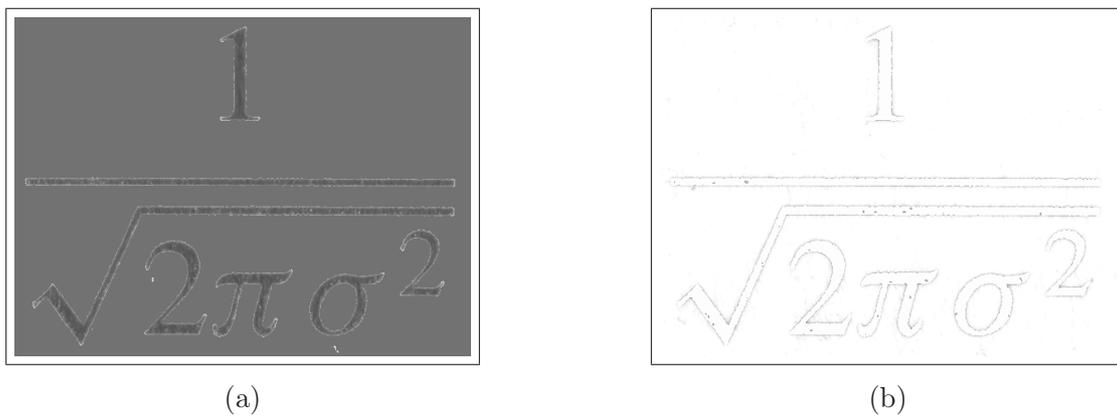


Figura 7.3: Picos indesejados: note o aparecimento de um contorno ao redor das bordas do texto (halo), tanto em (a) FG como em (b) BG.

7.3 Ajustamento e Ressuavização de Transições

Como o halo não pode ser removido via *data-filling*, os dados em si precisam ser alterados. De fato, alterá-los forçadamente é equivalente a modificar a imagem original. Assume-se que tal problema não está presente em toda a imagem, mas apenas onde a borda da máscara coincide com as transições da imagem. Assim, o primeiro passo é determinar as transições onde possivelmente o halo irá ocorrer, utilizando, sobre a máscara binária, operadores de Sobel [3], definidos como,

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}. \quad (7.1)$$

As bordas encontradas são, então, dilatadas [3] por um elemento estruturante de $d \times d$ *pixels*, de forma a delimitar uma vizinhança E ao redor das transições. Os *pixels* da imagem que coincidem com E são marcados como possíveis candidatos a processamento. O próximo passo é encontrar os *pixels* que contribuem para o efeito indesejado.

Sejam F e B os conjuntos de posições onde a máscara indica FG e BG, respectivamente. Computam-se as médias dos planos da seguinte maneira,

$$\begin{aligned} m_{FG} &= \text{mean}(x(i,j) | (i,j) \in F) \\ m_{BG} &= \text{mean}(x(i,j) | (i,j) \in B), \end{aligned} \quad (7.2)$$

onde $x(i,j)$ representa a imagem original. Marcam-se, em seguida, os *pixels* incluídos na região candidata E , cujos níveis de cinza estejam muito afastados da média do plano a que pertencem, ou seja,

$$\begin{aligned} C_{FG} &= \begin{cases} 0 & : |x(i,j) - m_{FG}| > \epsilon \mid (i,j) \in (F \cap E) \\ 1 & : \text{caso contrário} \end{cases} \\ C_{BG} &= \begin{cases} 0 & : |x(i,j) - m_{BG}| > \epsilon \mid (i,j) \in (B \cap E) \\ 1 & : \text{caso contrário} \end{cases}, \end{aligned} \quad (7.3)$$

onde ϵ é um limite de tolerância.

Depois, encontram-se os *pixels* marcados por C_{BG} cujos valores são inferiores a $(m_{FG} + \epsilon)$. Esses *pixels* são transferidos para o FG. Ou seja, *pixels* na região $(B \cap E)$ que estão distantes de m_{BG} mas que estão perto o suficiente de m_{FG} não serão pré/pós-processados. De forma idêntica, encontram-se os *pixels* marcados por C_{FG} cujos valores são superiores a $(m_{BG} - \epsilon)$. Esses *pixels* são transferidos para o BG. Em outras palavras, *pixels* que se encontram na região candidata $(F \cap E)$ e estão distantes de m_{FG} , mas estão próximos o suficiente de m_{BG} , não serão afetados pelo pré/pós-processador. A máscara M , C_{FG} , e C_{BG} são atualizados para acomodar essa transferência intercambiada de *pixels*, ilustrada na Figura 7.4.

Para a imagem da Figura 7.5 (a), e para $\epsilon = 16$ (dentro um total de 256 níveis de cinza), a máscara M , a vizinhança candidata E e o mapa de *pixels* a serem processados, isto é, $C =$

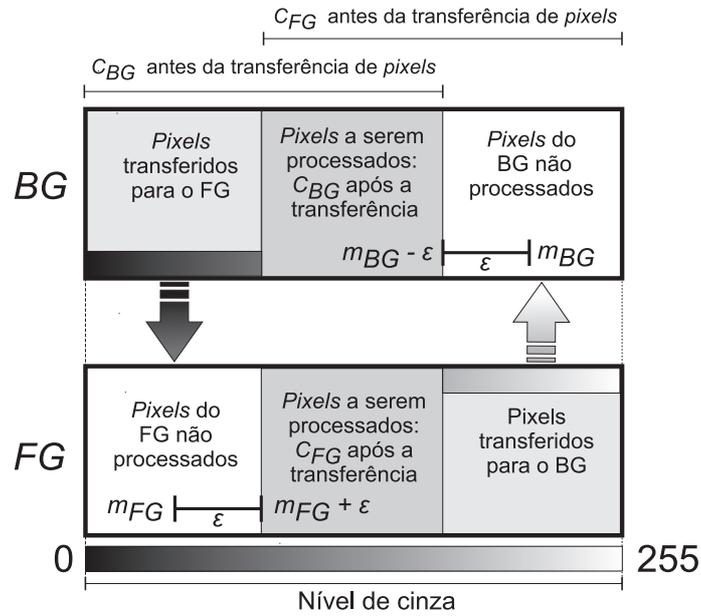
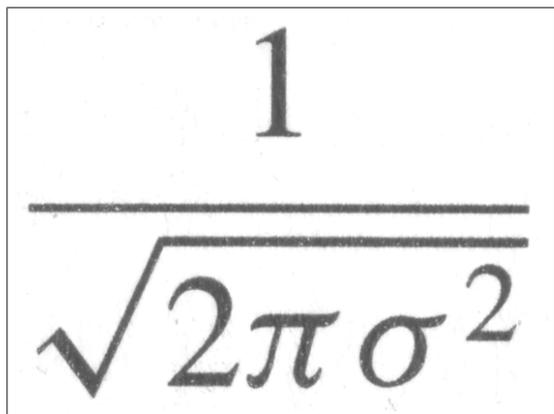


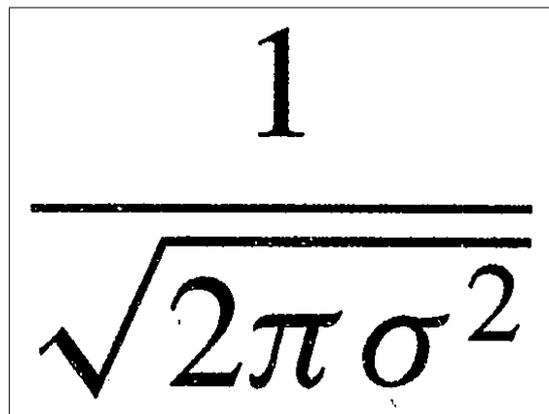
Figura 7.4: Processo de transferência intercambiada. A transferência tem por objetivo evitar que *pixels* sejam desnecessariamente processados.

$C_{FG} \cup C_{BG}$, podem ser visualizados nas Figuras 7.5 (b), (c) e (d), respectivamente. Com o objetivo de se eliminar os picos indesejados, os *pixels* em C_{FG} são substituídos por m_{FG} e os *pixels* em C_{BG} são substituídos por m_{BG} . O resultado desse processamento está ilustrado na Figura 7.6. Note que as transições, antes suaves, agora apresentam-se abruptas. Esse processo não é equivalente a aguçar a imagem como um todo. Apenas as regiões que são afetadas pelo halo são alteradas, o restante do FG e BG permanece intacto. A Figura 7.7 mostra o FG e o BG antes e após o processamento do halo e o procedimento de *data-filling*. Observe o aumento da qualidade dos planos.

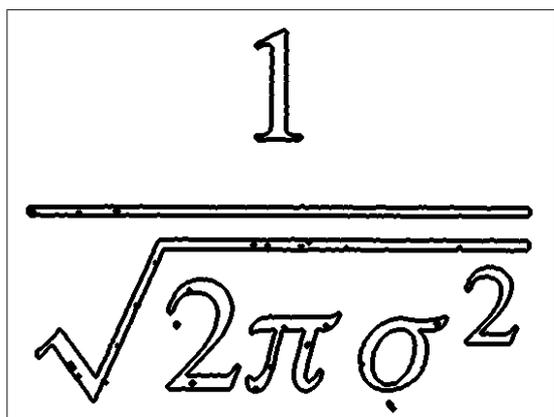
O mapa C pode ser enviado ao decodificador, como informação lateral, e os *pixels* que foram alterados podem ser recuperados via filtragem. A imagem resultante terá o aspecto daquela mostrada na Figura 7.8 (c), a qual teve a suavidade de suas bordas reconstruída a partir de um filtro gaussiano passa-baixas [3] de tamanho $h \times h$ e desvio padrão σ ($h = 15$ e $\sigma = 1,0$, nesse exemplo). Apenas para efeito de comparação, as Figura 7.8 (a) e (b) mostram a imagem original e a imagem aguçada por meio do pré-processamento sugerido, respectivamente. Observe que apenas as regiões que sofreram com o pré-processamento são suavizadas novamente. A informação lateral C também é codificada com o JBIG2, da mesma forma que a máscara binária M .



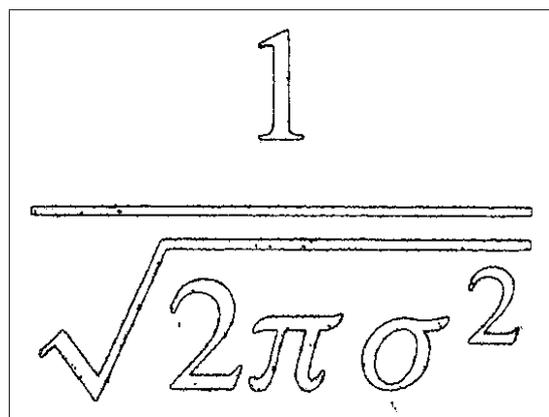
(a)



(b)



(c)



(d)

Figura 7.5: (a) Material digitalizado original; (b) máscara M ; (c) vizinhança candidata E ; (d) *pixels* a serem processados C .

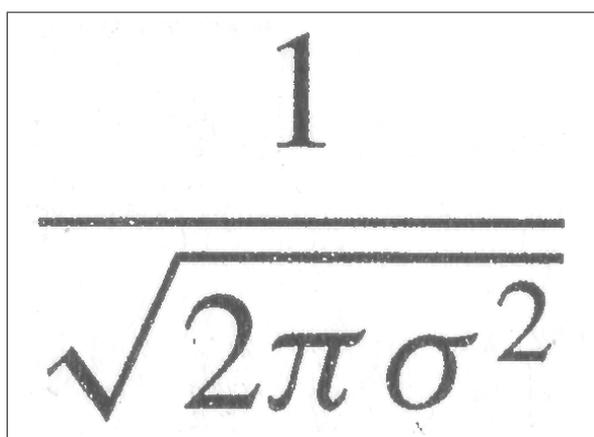
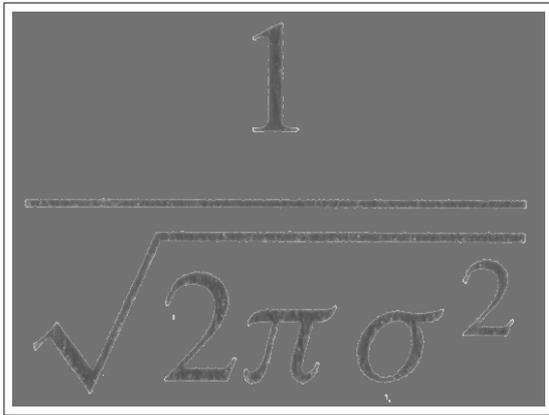
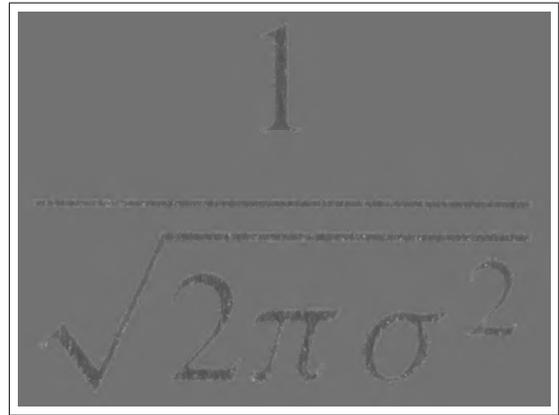


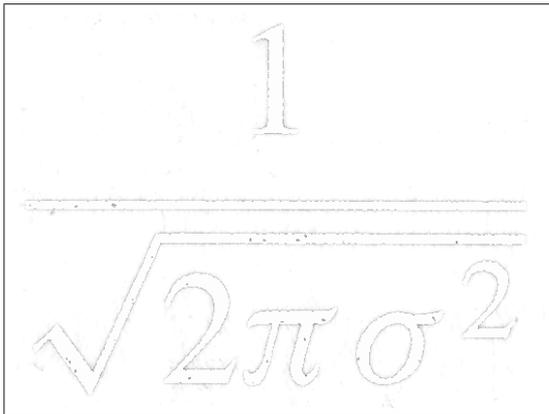
Figura 7.6: Imagem após pré-processamento. Note que as transições, antes suaves, agora apresentam-se aguçadas. O restante do FG e do BG permanece intacto.



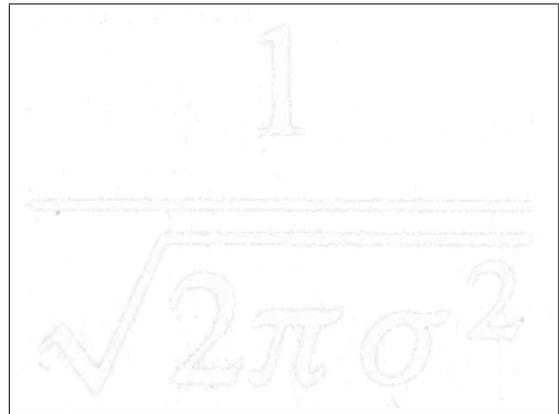
(a)



(b)



(c)



(d)

Figura 7.7: (a) FG e (c) BG originais; (b) FG e (d) BG processados. Note o contorno ao redor das bordas dos planos originais. O pré-processamento eleva a qualidade do FG e do BG, devido à remoção dos picos indesejados.

$$\frac{1}{\sqrt{2\pi\sigma^2}}$$

(a)

$$\frac{1}{\sqrt{2\pi\sigma^2}}$$

(b)

$$\frac{1}{\sqrt{2\pi\sigma^2}}$$

(c)

Figura 7.8: Pré/pós-processamento: (a) imagem original; (b) imagem aguçada; e (c) imagem ressuavizada.

7.4 Parametrização da Suavidade de Transições

As transições, aguçadas no codificador, precisam ser posteriormente ressuavizadas no decodificador. Assim, é necessário transmitir junto com o mapa C alguns parâmetros que permitam reconstruir a suavidade original das bordas dos caracteres e elementos gráficos. Na Seção 7.3, o valor de tolerância ϵ , a partir do qual se determina o mapa C , foi arbitrariamente ajustado. O mesmo ocorreu com os parâmetros do filtro gaussiano, h e σ . O desafio é encontrar os melhores valores para os parâmetros de pré-processamento ϵ , h e σ , a partir de uma abordagem baseada em taxa e distorção. Para isso, procura-se aproximar a solução, minimizando a seguinte função de custo,

$$J(\epsilon, h, \sigma) = D + \lambda \cdot R, \quad (7.4)$$

onde λ é um multiplicador de Lagrange, D é a distorção observada após os processos de pré-processamento, codificação, decodificação e pós-processamento das camadas MRC e R é a taxa de *bits* necessária à codificação do documento, definida como,

$$R = R_{FG} + R_{BG} + R_M + R_C + R_F, \quad (7.5)$$

onde R_{FG} , R_{BG} , R_M , R_C são as taxas parciais observadas na compressão de FG, BG, M e C , respectivamente, e R_F representa a quantidade de *bits* necessária para se realizar a sinalização do filtro de ressuavização. O algoritmo que determina o melhor valor para os parâmetros de pré-processamento ϵ , h e σ é descrito a seguir.

ALGORITMO 3

```

1   $h \leftarrow h_0$ ;
2   $\sigma \leftarrow \sigma_0$ ;
3  for  $\epsilon \leftarrow \epsilon_0$  to  $\epsilon_k$ 
4      do Gerar o mapa  $C$  utilizando  $\epsilon$ ;
5          Aguçar as bordas utilizando  $C$ ;
6          Rodar o algoritmo de data-filling;
7          Codificar e decodificar  $FG$ ,  $BG$  e  $M$  utilizando o modelo MRC proposto;
8          Codificar e decodificar  $C$  utilizando o JBIG2;
9          Armazenar a taxa  $R$  com a qual o documento foi codificado;
10         Filtrar as bordas utilizando um filtro gaussiano de parâmetros  $(h_0, \sigma_0)$ ;
11         Calcular a distorção  $D$ ;
12         Calcular e armazenar o custo  $J(\epsilon, h_0, \sigma_0)$ ;
13 Encontrar  $\hat{\epsilon}$  que minimiza  $J$ ;
14  $\epsilon_{best} \leftarrow \hat{\epsilon}$ ;
15 Gerar o mapa  $C_{best}$  utilizando  $\epsilon_{best}$ ;
16 Aguçar as bordas utilizando  $C_{best}$ ;
17 Rodar o algoritmo de data-filling;
18 Codificar e decodificar  $FG$ ,  $BG$  e  $M$  utilizando o modelo MRC proposto;
19 Codificar e decodificar  $C_{best}$  utilizando o JBIG2;
20 for  $h \leftarrow h_1$  to  $h_6$ 
21     do for  $\sigma \leftarrow \sigma_1$  to  $\sigma_8$ 
22         do Filtrar as bordas utilizando um filtro gaussiano de parâmetros  $(h, \sigma)$ ;
23             Calcular e armazenar a distorção  $D$ ;
24 Encontrar  $(\hat{h}, \hat{\sigma})$  que minimizam  $D$ ;
25  $h_{best} \leftarrow \hat{h}$ ;
26  $\sigma_{best} \leftarrow \hat{\sigma}$ ;

```

Quanto aos valores de h_0 e σ_0 , sugere-se utilizar 11 e 1.0, respectivamente. Os elementos h_1 a h_6 , citados na linha 20 do Algoritmo 3, pertencem ao conjunto H , definido por,

$$H = \{7, 9, 11, 13, 15, 17\}. \quad (7.6)$$

Já os elementos σ_1 a σ_8 , citados na linha 21, pertencem ao conjunto D_p definido por,

$$D_p = \{0, 7; 0, 8; 0, 9; 1, 0; 1, 1; 1, 2; 1, 3; 1, 4\}. \quad (7.7)$$

Esses dois conjuntos geram 48 filtros gaussianos diferentes. Assim, uma vez determinados os parâmetros h_{best} e σ_{best} , um código de 6 *bits* é enviado como informação lateral para indicar qual dos 32 filtros foi escolhido.

Uma vez que o FG e o BG são codificados por meio do AVC-I, faz-se necessário utilizar um parâmetro de quantização de projeto QP_D nos passos 7 e 18 do Algoritmo 3. O H.264/AVC

permite que o parâmetro de quantização QP varie de 0 a 51. Uma vez que a faixa de interesse situa-se em baixas taxas, sugere-se utilizar um valor alto para QP_D (acima de 30).

O MRC permite, ainda, a mudança da resolução espacial dos planos FG e BG. A presente tese propõe a variação do fator de escala S em 1, 1/2 e 1/4, e posterior seleção dos pontos de operação ótimos, determinados pelo Algoritmo 4 descrito a seguir.

ALGORITMO 4

```

1  for  $S \leftarrow 1, 1/2$  and  $1/4$ 
2      do for  $QP \leftarrow QP_0$  To  $QP_k$ 
3          do Codificar o documento;
4              Armazenar a taxa  $R$ ;
5              Calcular a distorção  $D$ ;
6              Armazenar o ponto taxa-distorção  $(R, D)$ ;
7  Ordenar os pontos  $(R, D)$  pela ordem crescente das taxas  $R$ ;
8   $N \leftarrow$  número de pontos  $(R, D)$ ;
9  for  $i \leftarrow 1$  To  $N$ 
10     do if  $D_i < D_{i-1}$ 
11         then Selecionar  $(R_i, D_i)$  como ponto ótimo;

```

Na próxima seção serão apresentados alguns resultados.

7.5 Resultados

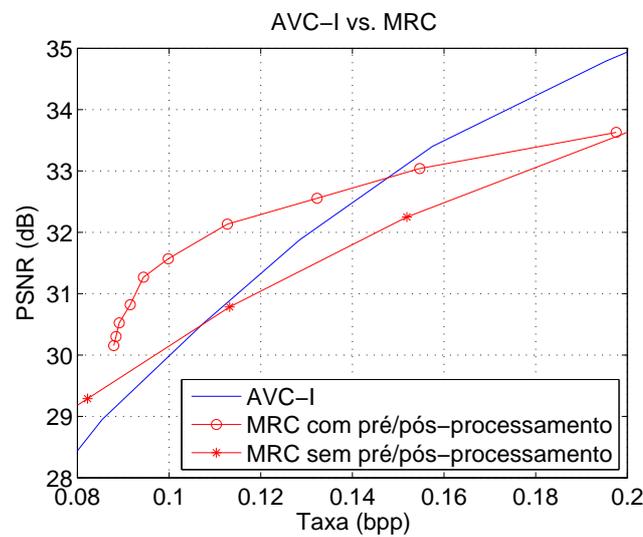
A Figura 7.9 (b) mostra um gráfico PSNR que compara os codificadores AVC-I e MRC (com e sem pré/pós-processamento), quando utilizados na codificação do documento da Figura 7.9 (a). Na geração desses resultados, a transformação de escala dos planos FG e BG ainda não foi considerada. Observe como o AVC-I e o MRC sem pré/pós-processamento são superados pelo MRC com o pré/pós-processamento proposto em mais de 1,0 dB a 0,1 *bit/pixel*. O intervalo, porém, onde a melhoria em relação ao AVC-I pode ser observada é bastante estreito (0,09 a 0,14 *bit/pixel*). Em relação ao MRC sem pré/pós-processamento esse intervalo é um pouco mais largo (0,09 a 0,2 *bit/pixel*). Isso ocorre devido à limitação em PSNR imposta pelo pré-processamento, que forçadamente altera os *pixels* do FG e do BG com o intuito de remover os picos indesejados. Além disso, a necessidade de se comprimir M e C sem perdas impõe uma limitação em taxa.

O gráfico da Figura 7.10 (a) mostra o efeito em taxa e distorção da transformação de escala dos planos FG e BG. Os fatores S utilizados foram 1 (sem transformação de escala), 1/2 (o número de linhas e colunas dos planos FG e BG é reduzido pela metade) e 1/4 (o número de linhas e colunas dos planos FG e BG é reduzido para um quarto). A idéia é tentar deslocar, mesmo que sutilmente, a curva de PSNR para cima e à esquerda, o que significa simultaneamente um alargamento do intervalo de taxa onde há ganho com relação ao AVC-I e um aumento da qualidade objetiva do documento. Dados os pontos que compõem as três curvas mostradas na Figura 7.10 (a), o Algoritmo 4, apresentado na seção anterior, seleciona aqueles que se mostram mais vantajosos em

termos de taxa e distorção. Dessa maneira, o comportamento final do codificador MRC proposto passa a ser representado pela curva da Figura 7.10 (b). As imagens das Figuras 7.11 (a) a 7.14 (a) também foram codificadas utilizando-se o método proposto. Suas respectivas curvas PSNR podem ser visualizadas nas Figuras 7.11 (b) a 7.14 (b).

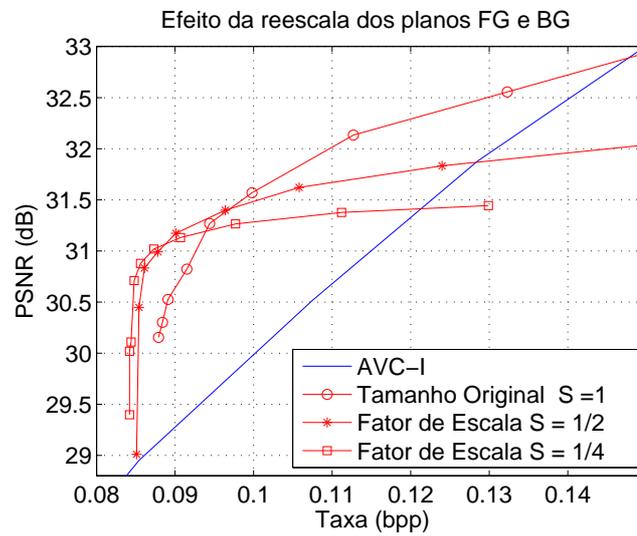


(a)

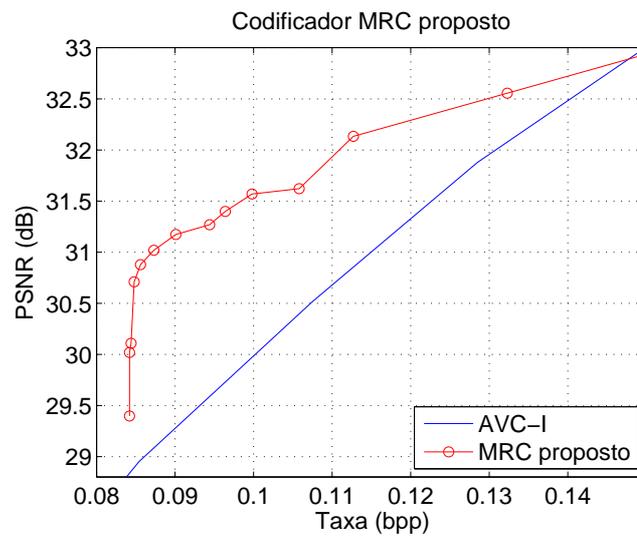


(b)

Figura 7.9: Comparação entre os codificadores AVC-I e MRC (com e sem pré/pós-processamento). Observe como o modelo MRC com o pré/pós-processamento proposto melhora a PSNR em mais de 1,0 dB em relação ao AVC-I e ao MRC sem pré/pós-processamento, a 0,1 *bit/pixel*.



(a)

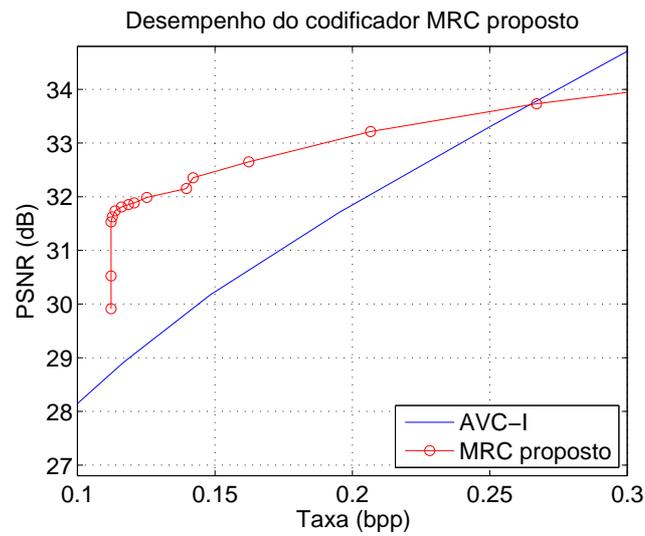


(b)

Figura 7.10: (a) Efeito da transformação de escala dos planos FG e BG. Os fatores de transformação de escala utilizados foram $S = 1, 1/2$ e $1/4$; e (b) desempenho final do codificador proposto.

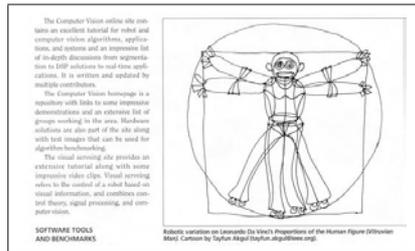


(a)

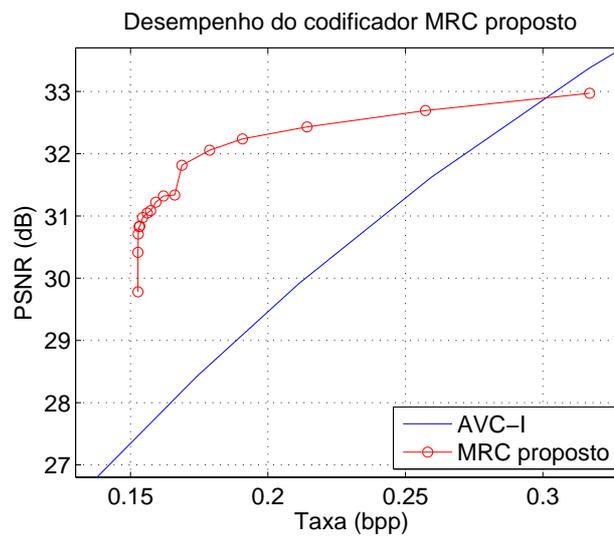


(b)

Figura 7.11: Desempenho do codificador MRC proposto, em comparação com o AVC-I: documento “horario”.



(a)

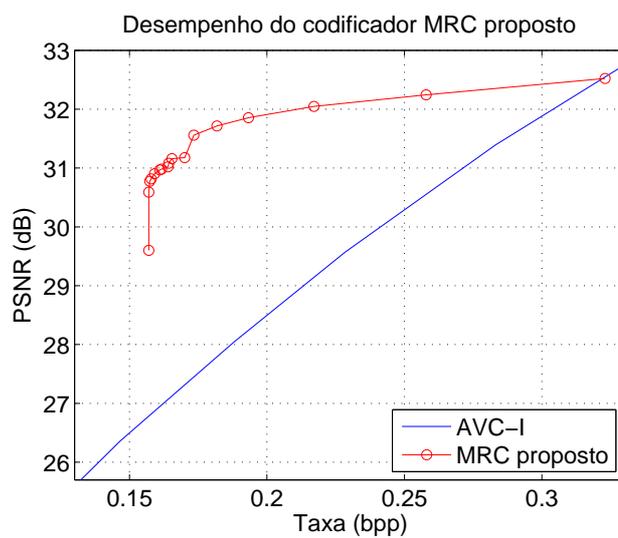


(b)

Figura 7.12: Desempenho do codificador MRC proposto, em comparação com o AVC-I: documento “vetruvian”.



(a)



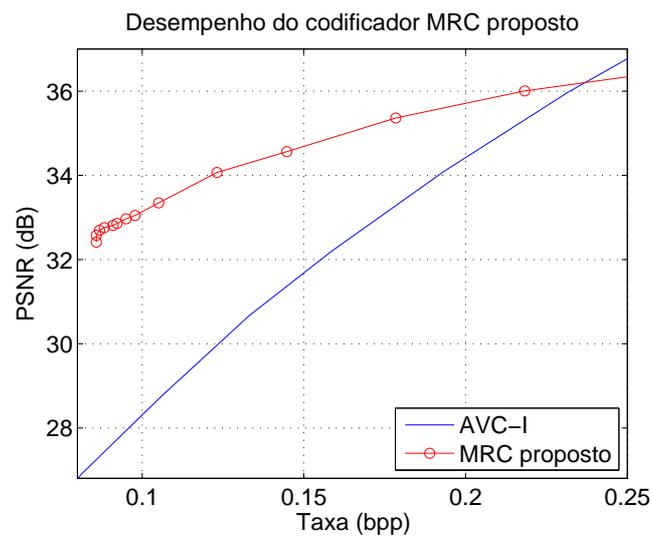
(b)

Figura 7.13: Desempenho do codificador MRC proposto em comparação com o AVC-I: documento “thinker”.

31 Tirando as dúvidas!
Principais dicas de segurança na UnB

- 1] Quem é o responsável pela segurança dos estacionamentos e pelo trânsito da UnB?
O DFTrans, o BPTTRAN e a Polícia Militar.
- 2] O que devo fazer se tiver meu veículo furtado ou roubado?
Se dirigir à Segurança do *Campus*, localizada no OCAS II – ao lado da quadra de esporte – e fazer a ocorrência. Posteriormente, dirigir-se à 2ª DP para registrar a ocorrência.
- 3] E se minha queixa for relacionada ao trânsito na UnB?
Depende da situação: pode ser o DETRAN ou, ainda, em outros casos, a PMDF, a Segurança da UnB ou o BPTTRAN.
- 4] Qual o alvo principal dos bandidos dentro do *Campus*?
Veículos, computadores pessoais, patrimônio da universidade, bolsas ou carteiras dentro de salas de aula ou espaços públicos. Portanto, certifique-se que fechou o seu veículo ao estacionar e não deixe os seus objetos pessoais largados em local público.

(a)



(b)

Figura 7.14: Desempenho do codificador proposto, em comparação com o AVC-I: documento “volta”.

A Figura 7.15 mostra uma região ampliada do documento “volta”, bem como suas versões codificadas e reconstruídas por meio dos codificadores AVC-I, JPEG2000 e MRC proposto. A taxa de *bits* observada foi de aproximadamente $0,1 \text{ bit/pixel}$, o que significa uma razão de compressão de 80 para 1 (ou 80:1). Note como o modelo proposto elimina o efeito de borda, presente nas imagens codificadas com o AVC-I e com o JPEG2000.

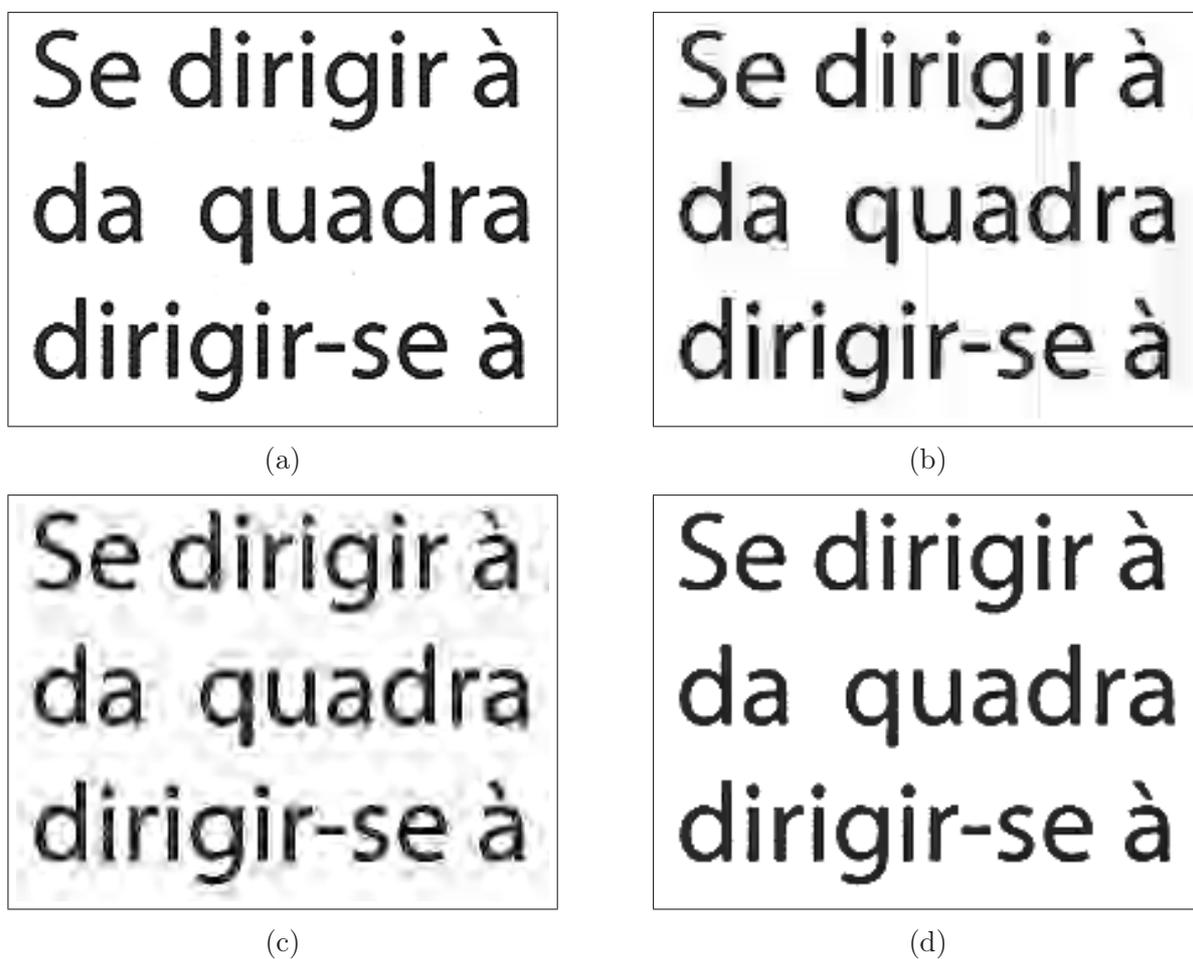
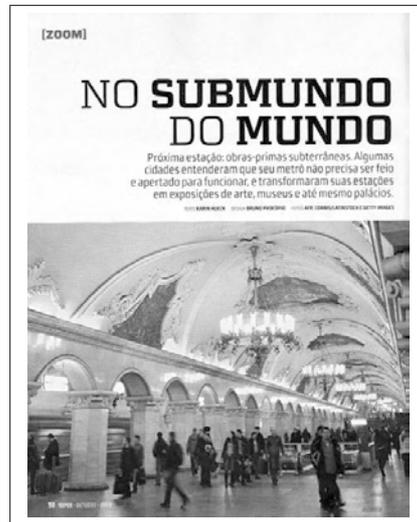
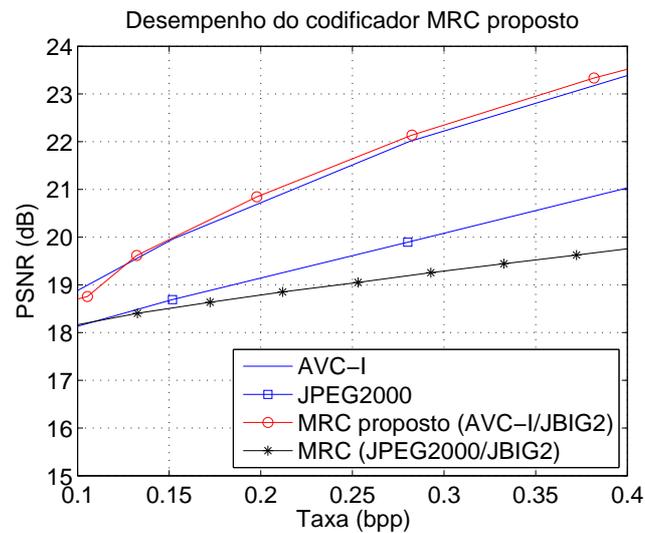


Figura 7.15: Comparação subjetiva entre codificadores. Parte ampliada do documento “volta” (Figura 7.14): (a) original; (b) codificado e reconstruído com o AVC-I (PSNR = 27,9314 dB); (c) codificado e reconstruído com o JPEG2000 (PSNR = 24,6271 dB); (d) codificado e reconstruído com o codificador MRC proposto (PSNR = 32,7260 dB). A taxa de *bits* observada foi aproximadamente $0,1 \text{ bit/pixel}$.

Para documentos mais complexos que os mostrados nas Figuras 7.11 a 7.14, duas foram as situações observadas. Na primeira situação, mais rara, ilustrada pela Figura 7.16, observou-se um desempenho objetivo ligeiramente superior ao AVC-I. Na segunda situação, mais frequente, ilustrada pelas Figuras 7.17 a 7.19, o modelo proposto não supera o AVC-I, mas ainda apresenta um desempenho objetivo superior ao codificador MRC baseado em JPEG2000.



(a)

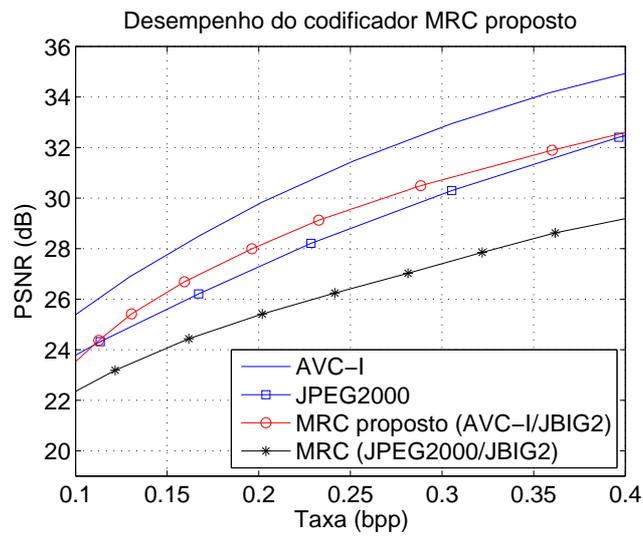


(b)

Figura 7.16: Desempenho do codificador MRC proposto, em comparação com os codificadores AVC-I, JPEG2000 e MRC baseado em JPEG2000: documento “super1”.



(a)

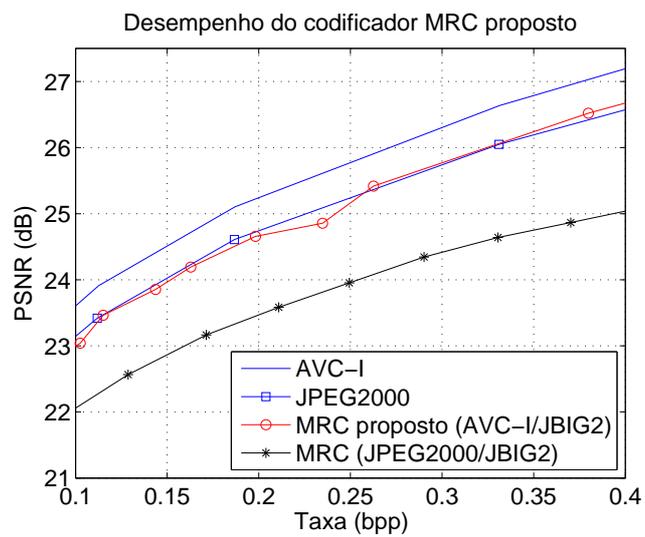


(b)

Figura 7.17: Desempenho do codificador MRC proposto, em comparação com os codificadores AVC-I, JPEG2000 e MRC baseado em JPEG2000: documento “IEEE1”.



(a)

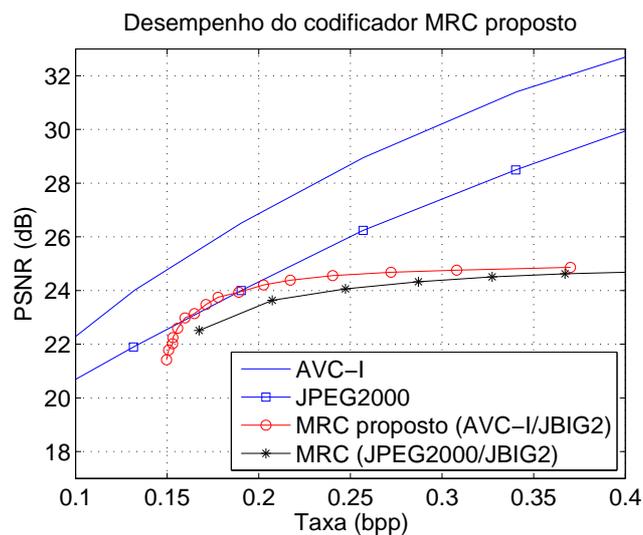


(b)

Figura 7.18: Desempenho do codificador MRC proposto, em comparação com os codificadores AVC-I, JPEG2000 e MRC baseado em JPEG2000: documento “IEEE2”.



(a)



(b)

Figura 7.19: Desempenho do codificador MRC proposto, em comparação com os codificadores AVC-I, JPEG2000 e MRC baseado em JPEG2000: documento “IEEE3”.

É importante lembrar que, em algumas situações, a representação MRC pode ser imposta como uma necessidade de projeto, mesmo que o desempenho em taxa e distorção resultante não seja superior ao apresentado por modelos que utilizam um único codificador. Em situações como essas, os principais fatores a serem avaliados são a capacidade de compressão do codificador e a qualidade subjetiva do documento reconstruído. As Figuras 7.20 a 7.23 mostram partes ampliadas do documento “IEEE3”, bem como suas versões codificadas e reconstruídas por meio dos codificadores AVC-I, JPEG2000 e MRC proposto. A taxa de *bits* observada foi de aproximadamente $0,19 \text{ bit/pixel}$. Note que o efeito de borda não se faz presente nos documentos codificados a partir da solução proposta.

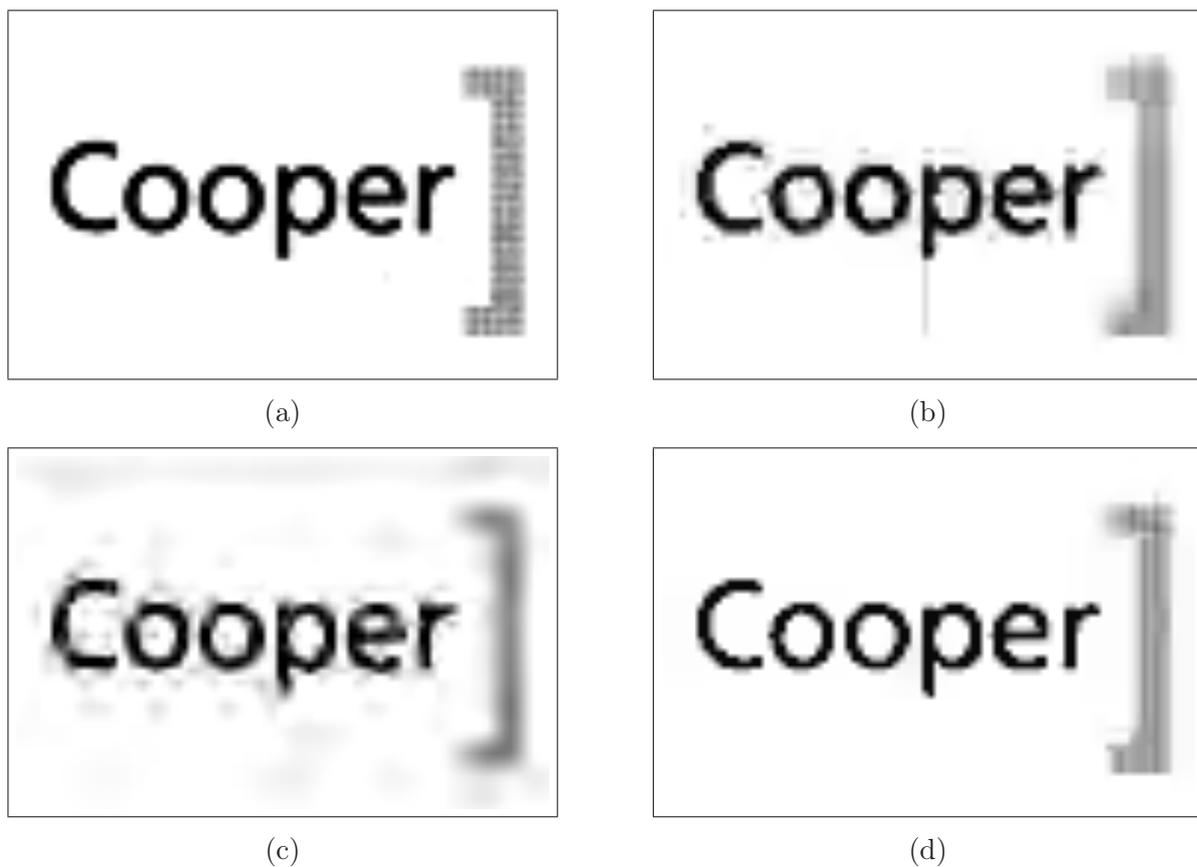


Figura 7.20: Comparação entre codificadores. Parte ampliada 1 do documento “IEEE3”: (a) original; (b) codificado e reconstruído com o AVC-I (PSNR = 26,5116 dB); (c) codificado e reconstruído com o JPEG2000 (PSNR = 23,9971 dB); (d) codificado e reconstruído com o codificador MRC proposto (PSNR = 24,2050 dB). A taxa de *bits* observada foi aproximadamente $0,19 \text{ bit/pixel}$.

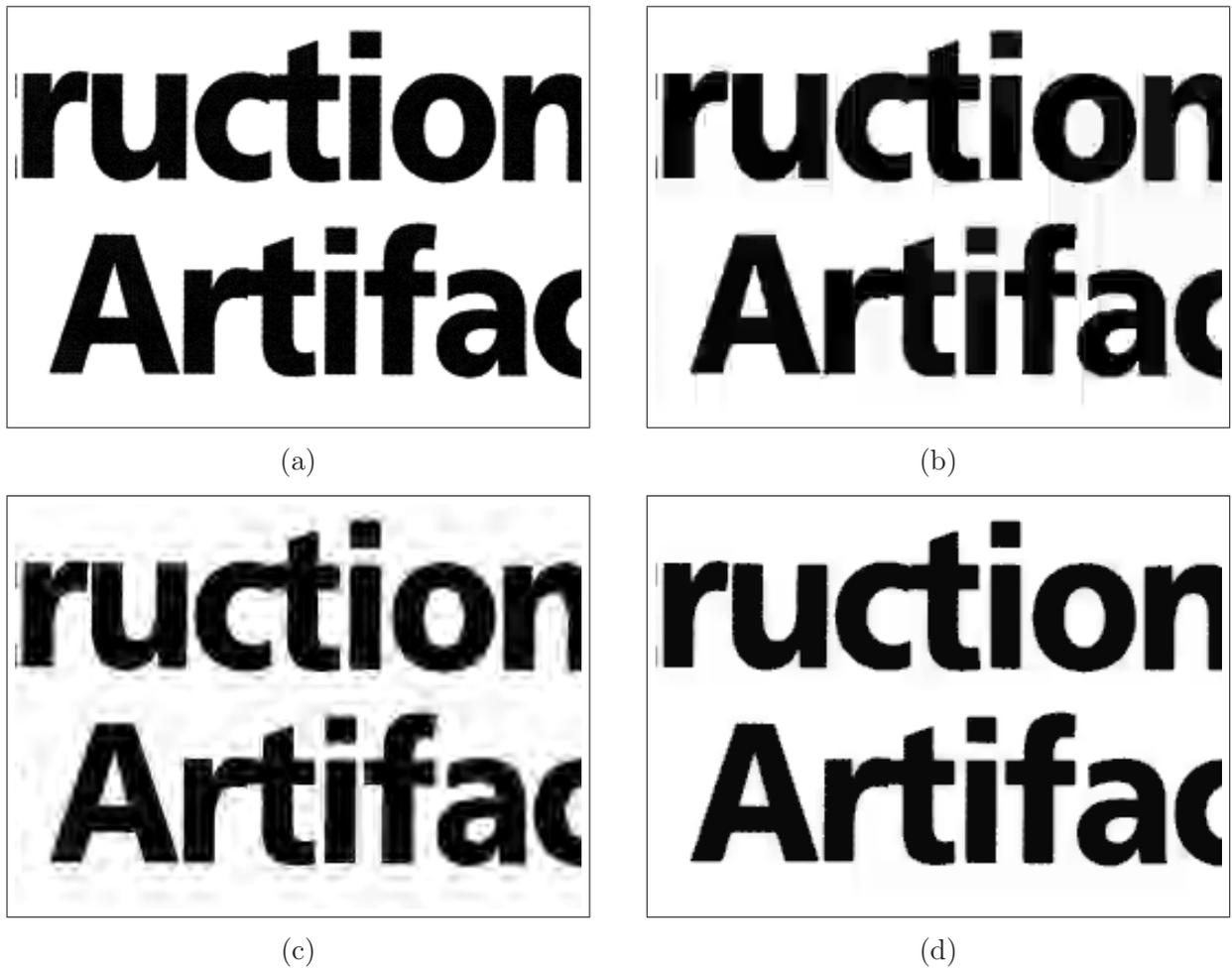


Figura 7.21: Comparação entre codificadores. Parte ampliada 2 do documento “IEEE3”: (a) original; (b) codificado e reconstruído com o AVC-I (PSNR = 26,5116 dB); (c) codificado e reconstruído com o JPEG2000 (PSNR = 23,9971 dB); (d) codificado e reconstruído com o codificador MRC proposto (PSNR = 24,2050 dB). A taxa de *bits* observada foi aproximadamente 0,19 *bit/pixel*.

Since the dawn of mankind, our ancestors have made objects to serve in everyday life. Inevitably, they will erode, fragment, and break apart due to various causes which include so-called “acts of God” (floods and earthquakes), war, neglect, carelessness, and many more. During field excavations, archaeologists and anthropologists rarely find intact artifacts. More often, they find fragments of complete relics, and much work is spent studying individual fragments and trying to find correlations between the fragments. In these pursuits, they typically examine the geometry of the fragment and match the geometry of a given fragment against a candidate set of fragments that are

(a)

Since the dawn of mankind, our ancestors have made objects to serve in everyday life. Inevitably, they will erode, fragment, and break apart due to various causes which include so-called “acts of God” (floods and earthquakes), war, neglect, carelessness, and many more. During field excavations, archaeologists and anthropologists rarely find intact artifacts. More often, they find fragments of complete relics, and much work is spent studying individual fragments and trying to find correlations between the fragments. In these pursuits, they typically examine the geometry of the fragment and match the geometry of a given fragment against a candidate set of fragments that are

(b)

Since the dawn of mankind, our ancestors have made objects to serve in everyday life. Inevitably, they will erode, fragment, and break apart due to various causes which include so-called “acts of God” (floods and earthquakes), war, neglect, carelessness, and many more. During field excavations, archaeologists and anthropologists rarely find intact artifacts. More often, they find fragments of complete relics, and much work is spent studying individual fragments and trying to find correlations between the fragments. In these pursuits, they typically examine the geometry of the fragment and match the geometry of a given fragment against a candidate set of fragments that are

(c)

Since the dawn of mankind, our ancestors have made objects to serve in everyday life. Inevitably, they will erode, fragment, and break apart due to various causes which include so-called “acts of God” (floods and earthquakes), war, neglect, carelessness, and many more. During field excavations, archaeologists and anthropologists rarely find intact artifacts. More often, they find fragments of complete relics, and much work is spent studying individual fragments and trying to find correlations between the fragments. In these pursuits, they typically examine the geometry of the fragment and match the geometry of a given fragment against a candidate set of fragments that are

(d)

Figura 7.22: Comparação entre codificadores. Parte ampliada 3 do documento “IEEE3”: (a) original; (b) codificado e reconstruído com o AVC-I (PSNR = 26,5116 dB); (c) codificado e reconstruído com o JPEG2000 (PSNR = 23,9971 dB); (d) codificado e reconstruído com o codificador MRC proposto (PSNR = 24,2050 dB). A taxa de *bits* observada foi aproximadamente 0,19 *bit/pixel*.



(a)



(b)



(c)



(d)

Figura 7.23: Comparação entre codificadores. Parte ampliada 4 do documento “IEEE3”: (a) original; (b) codificado e reconstruído com o AVC-I (PSNR = 26,5116 dB); (c) codificado e reconstruído com o JPEG2000 (PSNR = 23,9971 dB); (d) codificado e reconstruído com o codificador MRC proposto (PSNR = 24,2050 dB). A taxa de *bits* observada foi aproximadamente 0,19 *bit/pixel*.

Alternativamente, o usuário pode optar por não realizar o pós-processamento dos planos. Nesse caso, tem-se a versão não ressuavizada do documento, conforme ilustrado na Figura 7.24.

A rectangular box containing the text "Computational Reconstruction of Ancient Artifacts" in a bold, black, serif font. The text is centered and occupies most of the box's area.

**Computational
Reconstruction of
Ancient Artifacts**

(a)

A rectangular box containing the text "Computational Reconstruction of Ancient Artifacts" in a bold, black, serif font. The text is centered and occupies most of the box's area. The font appears slightly smoother than in (a).

**Computational
Reconstruction of
Ancient Artifacts**

(b)

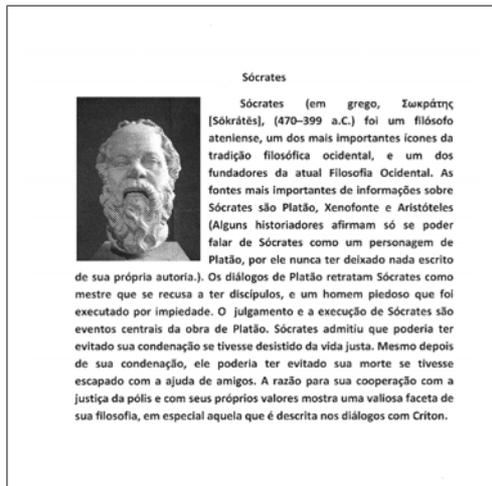
A rectangular box containing the text "Computational Reconstruction of Ancient Artifacts" in a bold, black, serif font. The text is centered and occupies most of the box's area. The font appears slightly less smooth than in (b).

**Computational
Reconstruction of
Ancient Artifacts**

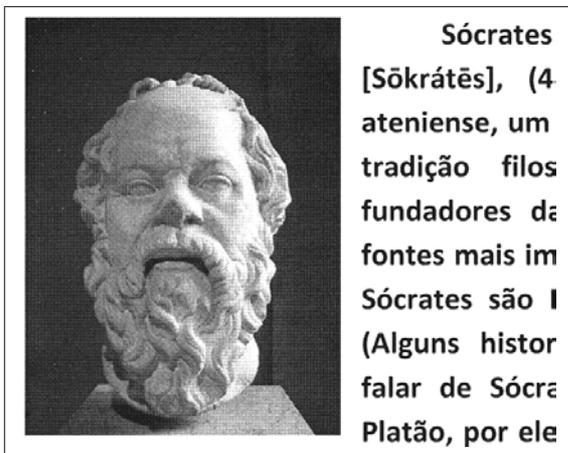
(c)

Figura 7.24: Efeito do pre/pós-processamento: (a) parte ampliada do documento “IEEE3” original; (b) parte ampliada do documento “IEEE3” ressuavizado (pré/pós-processado); e (c) parte ampliada do documento “IEEE3” não ressuavizado (pré-processado apenas).

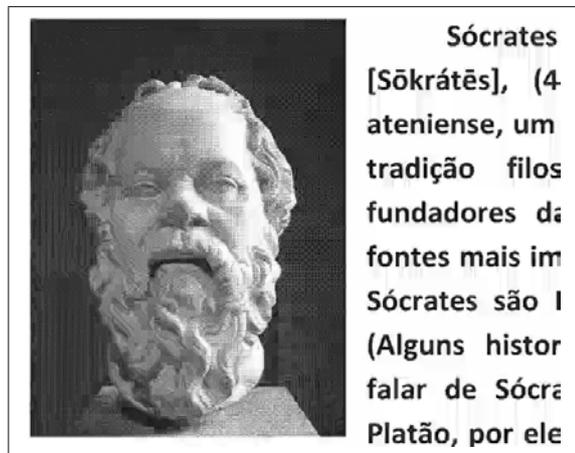
É possível, ainda, que o usuário deseje codificar o BG com melhor qualidade que o FG, ou vice-versa. Considere, por exemplo, o documento “Sócrates” mostrado na Figura 7.25 (a). Suponha que o usuário deseje dar mais qualidade à foto da escultura do filósofo e ao formato dos caracteres, sendo a textura do texto irrelevante. Nessas condições, o codificador MRC pode ser ajustado de forma a empregar um parâmetro de quantização QP_{BG} na codificação do BG (que contém a foto da escultura do filósofo) e outro parâmetro de quantização QP_{FG} na codificação do FG (que contém a textura dos caracteres), sendo $QP_{BG} < QP_{FG}$. Além disso, pode-se utilizar fatores de transformação de escala S_{BG} e S_{FG} diferentes para o BG e o FG, respectivamente. As Figuras 7.25 (b) a (e) mostram parte do documento “Sócrates original, além das versões codificadas e reconstruídas com o AVC-I (PSNR = 25,3025 e $QP = 45$), com o JPEG2000 (PSNR = 23,7246) e com codificador MRC proposto (PSNR = 28,3906, $QP_{BG} = 41$, $QP_{FG} = 51$, $S_{BG} = 1$ e $S_{FG} = 4$). A taxa de *bits* observada foi de aproximadamente 0,2 *bit/pixel*. Note como a solução proposta oferece um documento com qualidade subjetiva superior à alcançada pelo AVC-I e pelo JPEG2000.



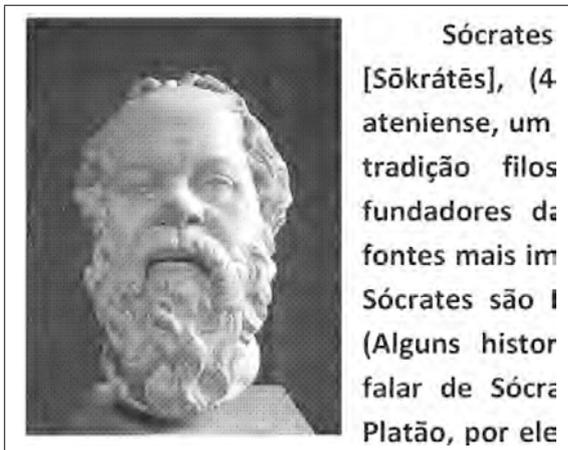
(a)



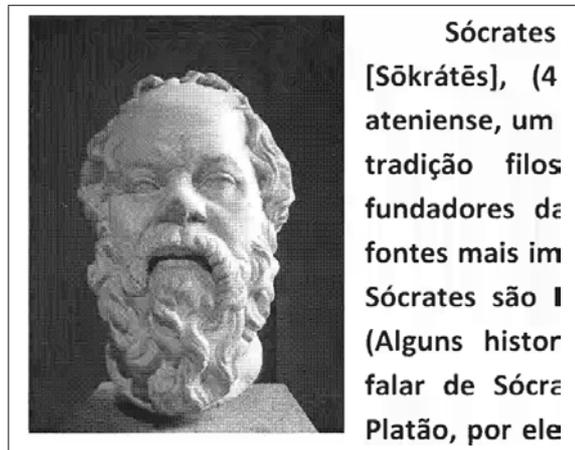
(b)



(c)



(d)



(e)

Figura 7.25: Comparação entre codificadores: (a) documento “Sócrates” original; (b) parte do documento “Sócrates” original; (c) parte do documento “Sócrates”codificado com o AVC-I (PSNR = 25,3025 e $QP = 45$); (d) parte do documento “Sócrates”codificado com o JPEG2000 (PSNR = 23,7246); e (e) parte do documento “Sócrates”codificado com o MRC proposto (PSNR = 28,3906, $QP_{BG} = 41$, $QP_{FG} = 51$, $S_{BG} = 1$ e $S_{FG} = 4$). A taxa de *bits* observada foi de aproximadamente 0,2 *bit/pixel*.

No presente capítulo foi apresentado um método para se contrabalancear os efeitos de transições suaves na codificação MRC de documentos digitalizados. Uma técnica para se aguçar apenas as transições de caracteres foi desenvolvida. Foi apresentado, também, um algoritmo que parametriza, no codificador, a suavidade das bordas de texto e elementos gráficos por meio de um filtro gaussiano passa-baixas, além de gerar um mapa de localização de picos indesejáveis a partir de uma análise de desempenho em taxa e distorção. Os parâmetros do filtro e o mapa gerado são utilizados no decodificador para reestabelecer a suavidade original das bordas. O método também contempla a mudança de escala dos planos FG e BG, e propõe um algoritmo para seleção de pontos de operação ótimos.

Em algumas situações, o modelo proposto foi capaz de superar objetivamente a abordagem que utiliza um codificador único, aqui representada pelo AVC-I. Isso ocorre principalmente para documentos menos complexos, codificados a baixas taxas. Para documentos mais complexos, o codificador MRC proposto normalmente é superado pelo AVC-I. A decisão de se utilizar a representação MRC, porém, pode não se basear exclusivamente em uma análise de taxa e distorção, mas também na exigência de qualidade subjetiva ou na necessidade de se integrar alguma outra funcionalidade, como o reconhecimento óptico de caracteres, por exemplo.

Apesar das técnicas aqui propostas terem sido desenvolvidas com o objetivo de se oferecer um documento reconstruído que seja o mais próximo possível de sua representação original, em algumas aplicações específicas a etapa de pós-processamento pode ser desligada. Subjetivamente, documentos aguçados (pré-processados apenas) podem apresentar melhor qualidade que documentos ressuavizados (pós-processados). Assim, o decodificador tem a opção de selecionar entre o documento ressuavizado e o aguçado. Além disso, decodificadores MRC clássicos podem ignorar o mapa C e decodificar a versão aguçada do documento.

Por fim, o modelo contempla o ajuste dos parâmetros de quantização e fatores de transformação de escala individualmente para os planos BG e FG. Isso faz com que o usuário possa optar por destinar mais qualidade ao plano que contém informações subjetivamente mais relevantes.

Capítulo 8

Método 4: Compressão de Livros Digitalizados utilizando o H.264/AVC

8.1 Considerações Iniciais

Recentemente, um novo algoritmo universal para codificação de imagens, denominado *Multidimensional Multiscale Parser* (ou MMP), foi proposto [8, 126, 127, 128, 129, 130]. Tal algoritmo não é fundamentado no tradicional esquema de codificação composto por transformação, quantização e codificação de entropia, como o AVC-I e o JPEG2000. Ao invés disso, a imagem é recursivamente segmentada em blocos de comprimento variável, sendo cada bloco codificado por meio de expansões e contrações de elementos em um dicionário, esse atualizado a partir também de expansões e contrações de concatenações de blocos previamente codificados. O efeito observado é o de grande adaptação em relação aos dados de entrada, o que justifica, em parte, a eficiência do codificador na compressão de um conjunto bastante heterogêneo de imagens.

Inspirado no princípio do casamento aproximado de padrões, proposto pelo MMP, o presente capítulo apresenta um método para codificação de livros digitalizados. Em documentos dessa natureza, há uma repetição muito grande de padrões dentro de uma mesma página e em páginas vizinhas. Essa característica sugere o uso da estimação de movimento do H.264/AVC como uma forma de segmentação da imagem original em blocos de tamanhos variados, com o objetivo de se realizar o casamento aproximado de padrões. Nesse cenário macroblocos anteriormente codificados podem ser utilizados como elementos de um dicionário na codificação dos macroblocos que ainda estão por codificar. Além disso, a estimação em metade e quarto de *pixel* pode ser vista como uma forma de se aumentar a variabilidade dos elementos do dicionário, proposta pelo MMP. A atualização desse dicionário, por sua vez, ocorre à medida em que novos macroblocos tornam-se disponíveis como referências para a predição inter-quadro.

De uma maneira geral, o método aqui apresentado consiste basicamente em pré-processar as páginas do livro digitalizado, para que a estimação de movimento do H.264/AVC consiga encontrar em uma mesma página ou em páginas passadas algum padrão semelhante àqueles contidos no macrobloco que está sendo codificado. A partir desse casamento, o codificador prossegue com a geração do resíduo de predição, a transformação, a quantização e a codificação de entropia. Isso

faz com que o método proposto seja um híbrido entre o esquema de codificação tradicional e aquele que utiliza o casamento aproximado de padrões, proposto pelo MMP.

Assume-se, ainda, que o usuário deseja comprimir o livro preservando as características que o aproximam ao máximo de sua versão impressa. Dessa forma, não serão considerados, para efeito de comparação, algoritmos de compressão de imagens binárias, mas sim algoritmos de compressão de imagens em tons contínuos.

A próxima seção descreve o pré-processamento sugerido.

8.2 Pré-processamento para Codificação de Livros Digitalizados utilizando o H.264/AVC

O pré-processamento aqui proposto visa explorar a recorrência de padrões dentro de uma mesma página e entre páginas vizinhas. Sabendo-se que o livro será codificado por meio do H.264/AVC, suas páginas são organizadas de forma que a estimação de movimento possa encontrar dentre os macroblocos previamente codificados algum cujo conteúdo seja semelhante ao conteúdo do macrobloco atual. Ou seja, o codificador irá operar a partir do paradigma do casamento aproximado de padrões. A Figura 8.1 ilustra a organização proposta. Cada página de dimensões $H \times W$ pixels é segmentada em 4 sub-páginas de $H/2 \times W/2$ pixels dispostas sequencialmente de forma a se gerar uma sequência de vídeo cujo número de quadros é igual à quatro vezes o número de páginas. A Figura 8.2 mostra um exemplo de três páginas de um livro e a Figura 8.3 ilustra a sequência de vídeo “samkhya” correspondente, criada a partir do esquema proposto.

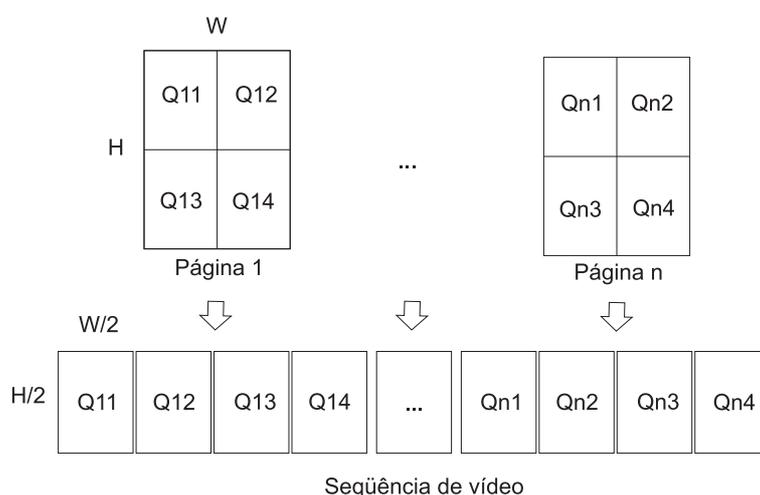


Figura 8.1: Esquema de pré-processamento das páginas do livro digitalizado.

A sequência de vídeo é, então, apresentada ao H.264/AVC. É importante ressaltar que dois parâmetros terão grande influência na eficiência do codificador. O primeiro é a quantidade de quadros de referência utilizados na predição inter e o segundo é a janela dentro da qual será realizada a busca por padrões, conforme ilustrado na Figura 8.4.

A seguir serão apresentados alguns resultados.

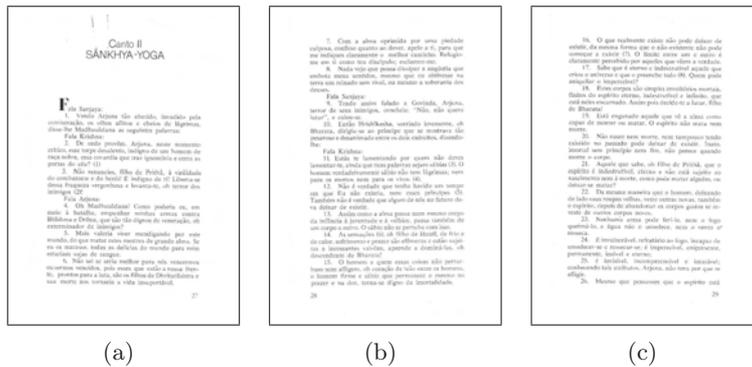


Figura 8.2: Exemplo de três páginas de um livro digitalizado: (a) página 1; (b) página 2; e (c) página 3.

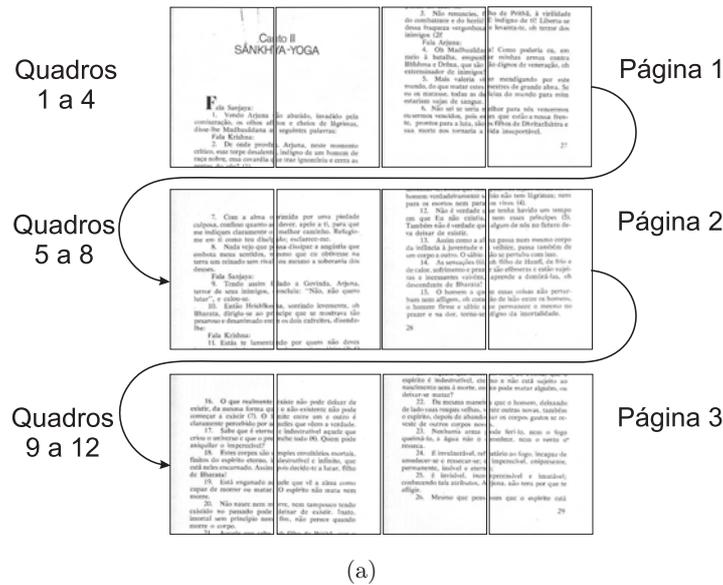


Figura 8.3: Sequência de vídeo “samkhya” criada a partir das três páginas mostradas na Figura 8.2: os quadros de 1 a 4, 5 a 8, e 9 a 12 são criados a partir das páginas 1, 2 e 3, respectivamente.

8.3 Resultados

Na geração dos resultados, as páginas do conjunto de teste foram comprimidas independentemente por meio do AVC-I e do JPEG2000, e de forma dependente utilizando o H.264/AVC, conforme proposto na seção anterior. Em todos os testes o primeiro quadro foi codificado como um quadro *I* (que utiliza somente os modos de predição intra). Os restantes foram codificados como quadros *P* (que também utilizam os modos de predição inter). Inicialmente avaliou-se o efeito da escolha do tamanho da janela de busca ($JB = 08, 16$ e 32 pixels) e da quantidade de quadros de referência ($QR = 1, 3$ e 5 quadros). Depois avaliou-se a eficiência do codificador para todo o conjunto de teste, fixando-se JB e QR .

As Figuras 8.5 (b) e (c) mostram as curvas PSNR¹ resultantes da codificação da sequência “guita”, cuja primeira página pode ser visualizada na Figura 8.5 (a). Na Figura 8.5 (b) o de-

¹A PSNR foi calculada à partir do MSE global.

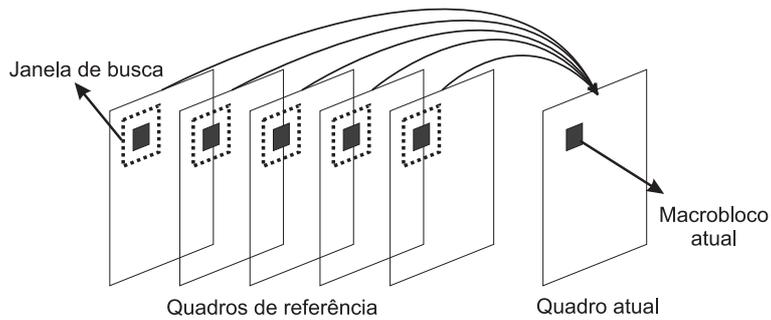


Figura 8.4: Ilustração do processo de previsão do macrobloco atual a partir da busca dentro de uma janela em quadros de referência anteriormente codificados.

sempenho do codificador foi avaliado para $JB = 08, 16$ e 32 pixels , mantendo-se $QR = 5$. Já na Figura 8.5 (c) a janela de busca JB foi mantida constante em 32 pixels e ajustou-se QR para 1, 3 e 5 quadros. Observe como o aumento de JB ou de QR determina um melhor desempenho do codificador em termos de taxa e distorção. Em qualquer caso, a eficiência do método proposto é igual ou superior à abordagem clássica, aqui representada pelos codificadores AVC-I e JPEG2000. Em especial, quando $JB = 32 \text{ pixels}$ e $QP = 5$, a proposta supera o AVC-I em mais de 2 dB e o JPEG2000 em mais de 5 dB, a uma taxa de $0,5 \text{ bit/pixel}$.

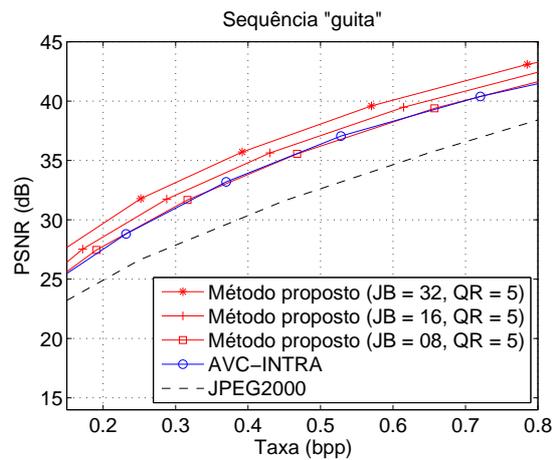
As Figuras 8.6 (a) a 8.10 (a) mostram a página inicial das sequências que compõem o conjunto de teste utilizado na avaliação do método proposto. Suas respectivas curvas PSNR são mostradas nas Figuras 8.6 (b) a 8.10 (b). Para essas sequências, JB foi mantido constante em 32 pixels e QR foi fixado em 5 quadros. Também para esses documentos, o método proposto supera a abordagem clássica.

Por fim, as Figuras 8.11 (a) a (d) mostram um trecho original da sétima página da sequência “principia”, bem como suas versões codificadas e reconstruídas por meio dos codificadores AVC-I, JPEG2000 e H.264/AVC (com o pré-processamento proposto) a aproximadamente $0,25 \text{ bits/pixel}$. Observe como o método proposto fornece, a uma mesma taxa, imagens com qualidade subjetiva superior.

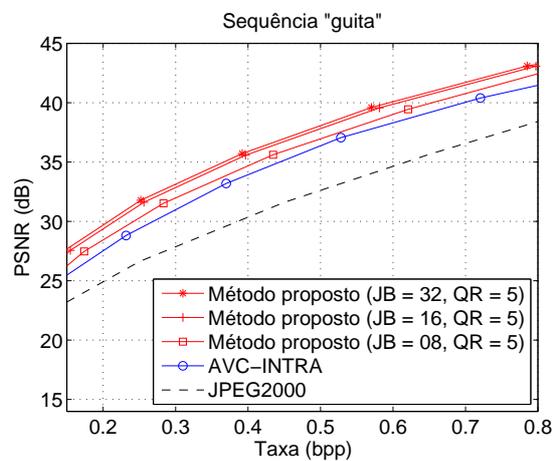
O presente capítulo apresentou um método para codificação de livros digitalizados que explora a recorrência de padrões dentro de uma mesma página e entre páginas vizinhas. Os resultados mostram que o método proposto supera objetiva e subjetivamente a abordagem clássica, aqui representada pelos codificadores JPEG2000 e AVC-I.



(a)



(b)

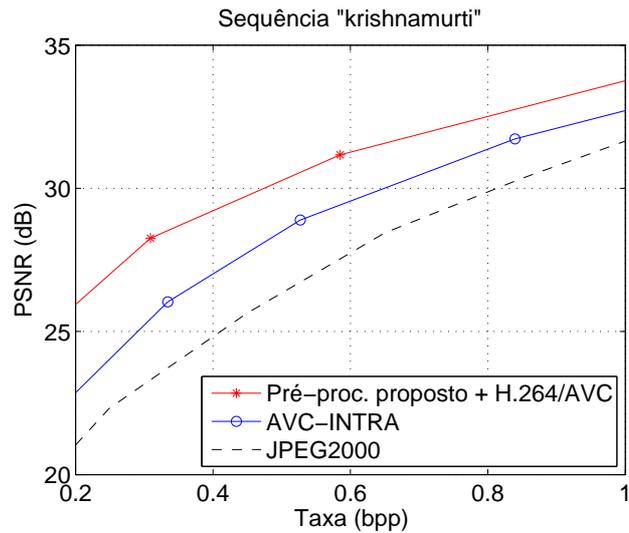


(c)

Figura 8.5: Sequência “guita”: 2 páginas, 8 quadros: (a) página inicial; (b) comparação entre o método proposto (pré-processamento das páginas e codificação utilizando o H.264/AVC), com $QR = 5$ quadros e $JB = 08, 16$ e 32 pixels, e a abordagem clássica, que comprime cada página independentemente (codificação utilizando o AVC-I ou o JPEG2000); e (c) comparação entre o método proposto, com $JB = 32$ pixels e $QR = 1, 3$ e 5 quadros, e os codificadores AVC-I e JPEG2000. Para $JB = 32$ e $QR = 5$, por exemplo, o método proposto supera o AVC-I em mais de 2 dB e o JPEG2000 em mais de 5 dB, a uma taxa de $0,5$ bit/pixel.



(a)

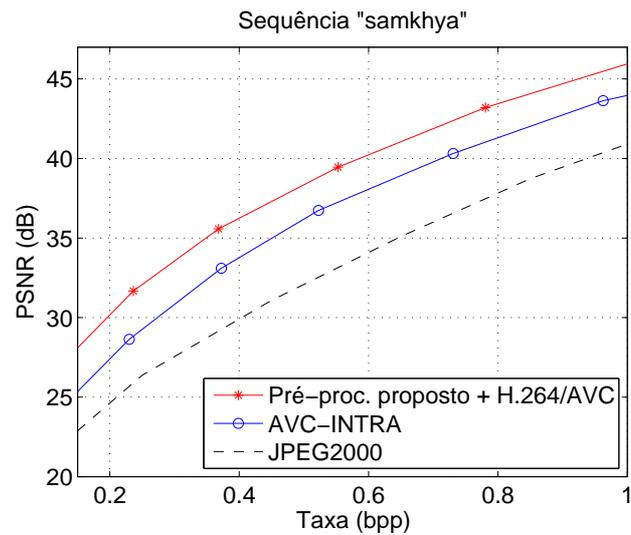


(b)

Figura 8.6: Sequência “krishnamurti”: 2 páginas, 8 quadros: (a) página inicial; e (b) comparação entre o método proposto (pré-processamento das páginas e codificação utilizando o H.264/AVC) e a abordagem clássica, que comprime cada página independentemente (codificação utilizando o AVC-I ou o JPEG2000).

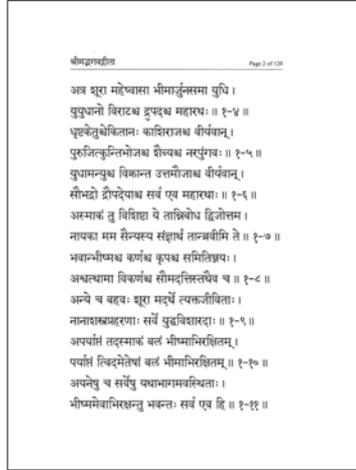


(a)

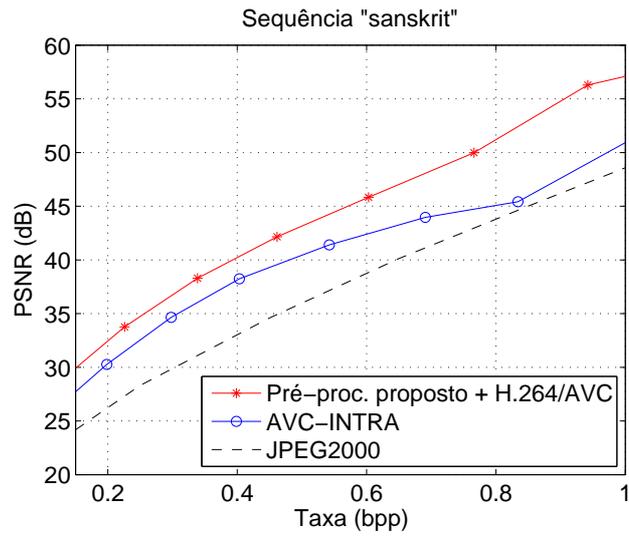


(b)

Figura 8.7: Sequência “samkhya”: 3 páginas, 12 quadros: (a) página inicial; e (b) comparação entre o método proposto (pré-processamento das páginas e codificação utilizando o H.264/AVC) e a abordagem clássica, que comprime cada página independentemente (codificação utilizando o AVC-I ou o JPEG2000).

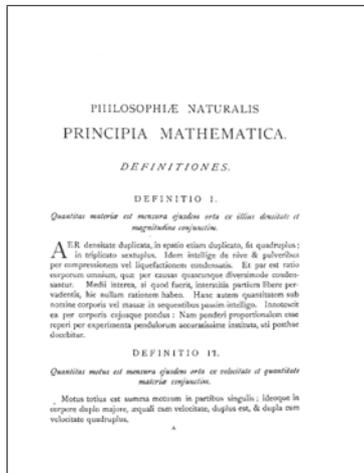


(a)

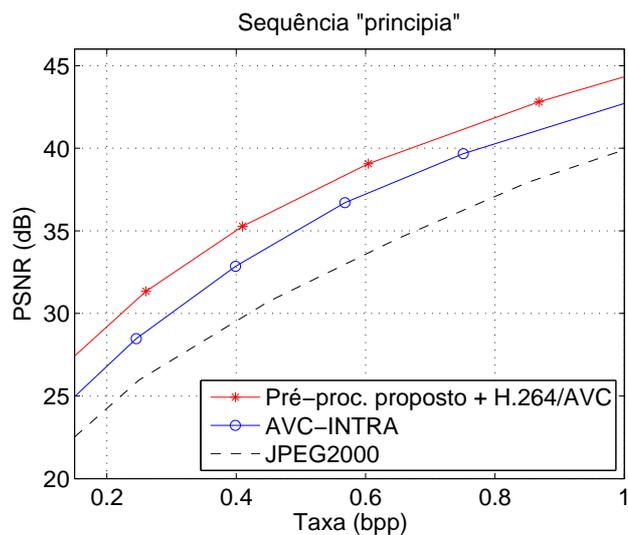


(b)

Figura 8.8: Sequência “sanskrit”: 4 páginas, 16 quadros: (a) página inicial; e (b) comparação entre o método proposto (pré-processamento das páginas e codificação utilizando o H.264/AVC) e a abordagem clássica, que comprime cada página independentemente (codificação utilizando o AVC-I ou o JPEG2000).



(a)



(b)

Figura 8.10: Sequência “principia”: 8 páginas, 32 quadros: (a) página inicial; e (b) comparação entre o método proposto (pré-processamento das páginas e codificação utilizando o H.264/AVC) e a abordagem clássica, que comprime cada página independentemente (codificação utilizando o AVC-I ou o JPEG2000).

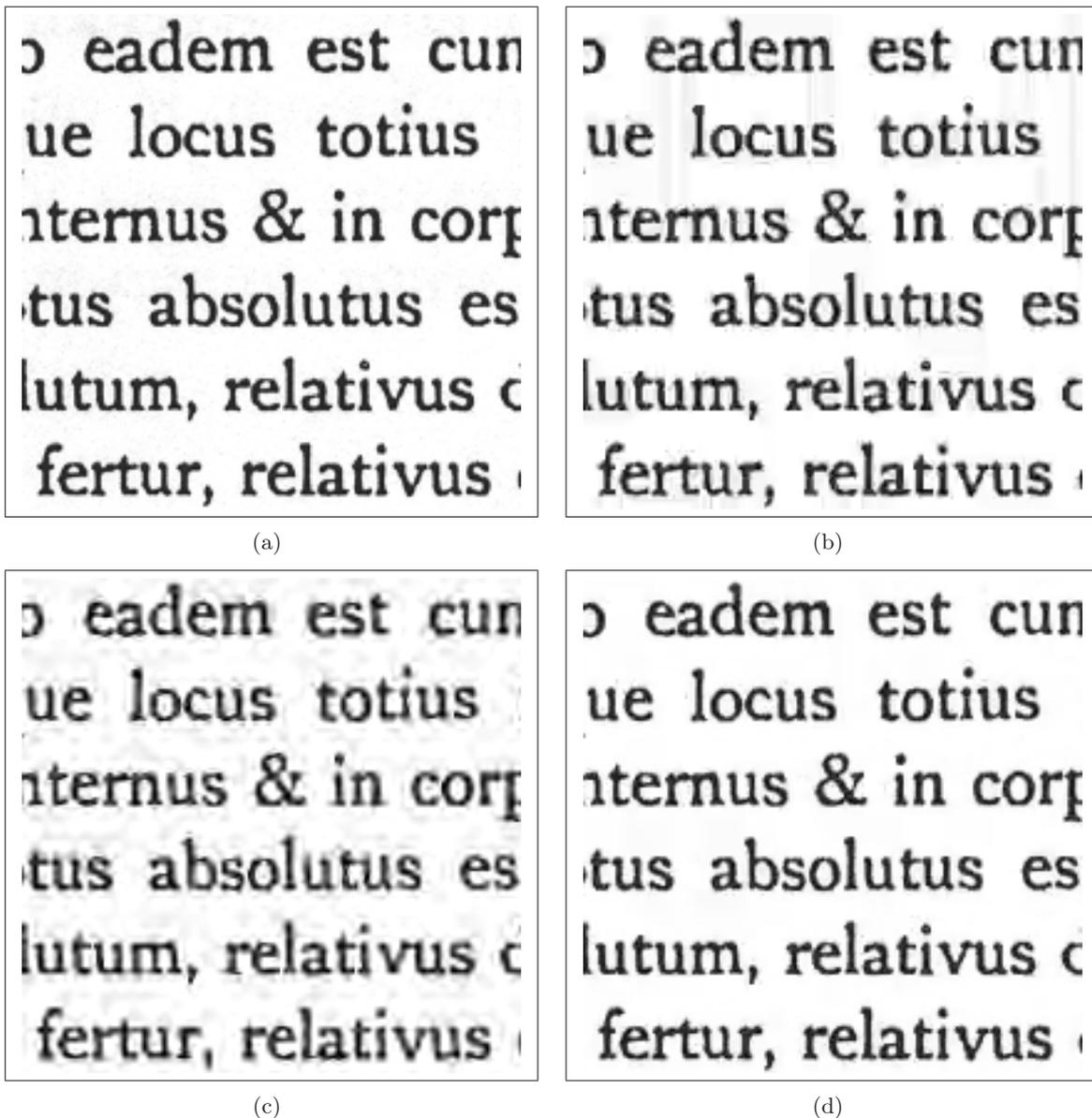


Figura 8.11: Avaliação subjetiva: (a) trecho original da sétima página da sequência “principia”, bem como suas versões codificadas e reconstruídas por meio dos codificadores (b) AVC-I, (c) JPEG2000 e (d) método proposto (pré-processamento das páginas e codificação utilizando o H.264/AVC) a aproximadamente 0,25 *bits*/pixels. Observe como o método proposto fornece uma qualidade subjetiva superior.

Capítulo 9

Conclusão

9.1 Considerações Iniciais

A presente tese teve por meta reunir o que há de mais avançado em compressão de imagens, com o objetivo de propor novas técnicas, distribuídas por quatro métodos, destinadas à compressão eficiente de documentos digitais. As conclusões serão apresentadas a seguir.

9.2 Método 1: Advanced Video Coding - Composto

A principal contribuição desse método foi agregar ao H.264/AVC a possibilidade de se codificar adaptativamente regiões de texto e imagem a partir da análise do conteúdo de macroblocos. Um algoritmo para seleção dos parâmetros de quantização aplicados a regiões pictóricas e textuais também foi apresentado. O método proposto, ao ser generalizado, pode ser utilizado na codificação de regiões de interesse, funcionalidade antes não especificada no padrão H.264/AVC.

Os experimentos mostram que para uma mesma taxa de *bits* é possível elevar a qualidade objetiva de regiões textuais, enquanto a qualidade global da imagem não é demasiadamente reduzida. As perdas em regiões pictóricas são tais que o AVC-C, em geral, ainda supera objetivamente o JPEG2000. Apesar do AVC-C não superar o AVC-I em termos globais, o método permite codificar regiões de texto com qualidade superior.

Verificou-se, também, que a eficiência em taxa e distorção do AVC-C depende do equilíbrio entre o número de macroblocos textuais e o número de macroblocos pictóricos. Quando o número de macroblocos pictóricos é bem maior que o número de macroblocos textuais, é possível aumentar significativamente a qualidade das regiões de texto. Quando a situação se inverte, ou seja, quando o número de macroblocos textuais é muito maior que o número de macroblocos pictóricos, a melhoria da qualidade das regiões de texto é bem reduzida.

Uma outra característica interessante do codificador proposto é o fato de ele ser compatível com o AVC-I. Em outras palavras, usuários que tenham um decodificador AVC-I podem, sem qualquer prejuízo, decodificar os documentos codificados pelo AVC-C.

Quanto à complexidade, o AVC-C apresenta um custo bem mais elevado que outros codificadores, como, por exemplo, o JPEG2000 ou o próprio AVC-I. O alto grau de complexidade do AVC-C está relacionado ao fato da codificação requerer múltiplas passagens. A aplicação do codificador proposto, porém, não é comprometida, uma vez que implementações rápidas podem ser utilizadas e que a compressão de documentos é frequentemente realizada *off-line*.

Para versões futuras, pode-se pensar em técnicas que reduzam o número de passagens atualmente empregado na codificação dos documentos.

9.3 Método 2: Compressão MRC baseada em H.264/AVC-I e JBIG2 para Documentos Gerados Eletronicamente

Na compressão MRC, espera-se um melhor desempenho em taxa e distorção quando cada camada é individualmente codificada por meio de algoritmos mais eficientes. Assim, uma das contribuições desse método foi integrar o H.264/AVC e o JBIG2 ao modelo MRC básico composto por duas camadas em tons contínuos (FG e BG) e uma camada binária (máscara). Também foram propostas a generalização do algoritmo de segmentação por limiarização de blocos e uma técnica para preenchimento de regiões irrelevantes baseada nas predições $I16$ do H.264/AVC.

Os resultados mostram que o codificador MRC proposto apresenta um desempenho objetivo e subjetivo superior ao dos codificadores AVC-I e JPEG2000. Sem dúvida o esquema MRC proposto, baseado no H.264/AVC e no JBIG2, estabelece um novo patamar de codificação no que se refere à compressão de documentos compostos.

A desvantagem do método, porém, é que o elevado ganho em relação à abordagem baseada em um único codificador limita-se a documentos gerados eletronicamente, pois em documentos dessa natureza as bordas de texto são suficientemente abruptas a ponto de poderem ser modeladas por meio de uma máscara binária. O mesmo não ocorre com documentos digitalizados.

Para trabalhos futuros, pode-se pensar em integrar também as predições $I8$ e $I4$ à técnica de *data-filling* proposta.

9.4 Método 3: Compressão MRC baseada em H.264/AVC-I e JBIG2 para Documentos Digitalizados

Em documentos digitalizados, as bordas de texto são caracterizadas por transições suaves, fazendo com que a segmentação em planos seja dificultada. A contribuição desse método está em técnicas desenvolvidas com o objetivo de se contrabalancear os efeitos de transições suaves na codificação MRC de documentos digitalizados. Em resumo, tais técnicas aguçam as bordas de texto antes da codificação MRC, ressuavizando-as no decodificador a partir de parâmetros previamente determinados no codificador.

Os resultados mostram que, para documentos menos complexos, o método proposto é capaz de superar, em baixas taxas, a abordagem baseada em um codificador único. No caso de documentos

mais complexos, o resultado mais frequentemente observado é a superioridade do AVC-I sobre a codificação proposta. Ainda assim, essa última supera o modelo MRC baseado no JPEG2000.

A decisão de se utilizar a representação em camadas, proporcionada pelo MRC, nem sempre é realizada a partir de uma imposição baseada exclusivamente em desempenho de taxa e distorção. Assim, o método proposto apresenta-se como uma alternativa competitiva, por ser capaz de gerar documentos com boa qualidade subjetiva, a muito baixas taxas.

Em algumas aplicações específicas, a etapa de pós-processamento pode ser desligada, pois documentos aguçados podem apresentar melhor qualidade subjetiva. Além disso, decodificadores MRC clássicos podem ignorar o pós-processamento (ressuavização das bordas) e decodificar a versão aguçada do documento.

Para trabalhos futuros, pode-se propor uma técnica de segmentação dentro do arcabouço do H.264/AVC, onde macroblocos seriam otimamente decompostos a partir de uma análise em taxa e distorção. A separação em planos poderia até não ocorrer, caso a codificação intra se mostrasse mais vantajosa. Além disso, pode-se investigar novas formas de se gerar a informação lateral necessária à reconstrução dos *pixels* que foram alterados na etapa de pré-processamento.

9.5 Método 4: Compressão de Livros Digitalizados utilizando o H.264/AVC

A contribuição desse método está na utilização dos modos de predição inter do H.264/AVC como uma forma de se realizar a compressão de livros a partir do casamento aproximado de padrões. Assim, foi proposto um método de codificação híbrido, que utiliza o paradigma fundamentado na recorrência de padrões, representado pelo MMP, e o esquema de codificação tradicional, baseado nas etapas de transformação, quantização e codificação de entropia. Os resultados mostram que com o método proposto é possível alcançar melhor qualidade objetiva e subjetiva, quando comparado ao AVC-I e ao JPEG2000.

Trabalhos futuros podem incluir a pré-determinação do tamanho do quadro a ser utilizado na construção da sequência de vídeo, bem como do tamanho da janela de busca e o número de quadros de referência a serem considerados na predição inter. Tal método poderia, ainda, ser empregado na codificação de livros compostos. O pré-processamento, nesse caso, incluiria a classificação das páginas em três tipos: *pictóricas*, *textuais* e *e compostas*. As páginas pictóricas e as páginas textuais seriam codificadas com o AVC-I e com o método 4 proposto, respectivamente. Já as páginas compostas seriam submetidas a um processo de segmentação em *regiões pictóricas* e *regiões textuais*. As regiões pictóricas seriam codificadas com o AVC-I e as regiões textuais passariam por uma reorganização antes de serem codificadas pelo H.264/AVC.

9.6 Considerações Finais

As contribuições aqui apresentadas, longe de serem definitivas, caracterizam-se por mais um passo em direção à compressão eficiente de documentos digitais. Constituem um avanço, em si, por

oferecem, a uma mesma taxa de *bits*, documentos com qualidade objetiva e/ou subjetva superior a abordagens consideradas o estado-da-arte em compressão de imagens estáticas. Por outro lado, representam também um avanço indireto no que se refere à publicação e ao acesso à produção intelectual humana, cumprindo com o objetivo inicialmente proposto pela tese.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Times Online. <http://archive.timesonline.co.uk/tol/archive/>, último acesso em 30/10/2008.
- [2] Domínio Público. <http://www.dominiopublico.gov.br/>, último acesso em 30/10/2008.
- [3] GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 2nd. ed. EUA: Prentice-Hall, 2002.
- [4] JBIG. Information Technology - Coded Representation of Picture and Audio Information - Progressive Bi-level Image Compression. ITU-T Recommendation T.82. Março 1993.
- [5] JBIG2. Information Technology - Coded Representation of Picture and Audio Information - Lossy/Lossless Coding of Bi-level Images. ITU-T Recommendation T.88. Março 2000.
- [6] JPEG. Information Technology - Digital Compression and Coding of Continuous-tone Still Images - Requirements and Guidelines. ITU-T Recommendation T.81. Setembro 1992.
- [7] JPEG. Information Technology - JPEG2000 Image Coding System - Part 1: Core Coding System. ISO/IEC 15444-1. 2000.
- [8] CARVALHO, M. de; SILVA, E. da; FINAMORE, W. Multidimensional Signal Compression using Multiscale Recurrent Patterns. *Elsevier Signal Processing*, n. 82, pp. 1559–1580, Novembro 2002.
- [9] JVT. Advanced Video Coding for Generic Audiovisual Services. ITU-T Recommendation H.264. Novembro 2007.
- [10] MRC. Mixed Raster Content (MRC). ITU-T Recommendation T.44. 1999.
- [11] GONZALEZ, R. C.; WOODS, R. E.; EDDINS, S. L. *Digital Image Processing using Matlab*. EUA: Prentice-Hall, 2004.
- [12] SHARMA, G. *Digital Color Imaging Handbook*. EUA: CRC Press, 2002.
- [13] PRATT, W. K. *Digital Image Processing*. 3rd. ed. EUA: Willey, 2001.
- [14] WU, H. R.; RAO, K. R. *Digital Video Image Quality and Perceptual Coding*. 3rd. ed. EUA: Willey, 2001.
- [15] BOVIK, A. *Handbook of Image and Video Processing*. EUA: Academic Press, 2000.
- [16] CASTLEMAN, K. R. *Digital Image Processing*. 2nd. ed. EUA: Prentice Hall, 1995.

- [17] IEEE Transactions on Image Processing. <http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=83>, último acesso em 01/07/2008.
- [18] LYONS, R. G. *Understanding Digital Signal Processing*. 2nd. ed. EUA: Prentice-Hall, 2004.
- [19] DINIZ, P. S. R.; SILVA, E. A. B. da; NETTO, S. L. *Digital Signal Processing: System Analysis and Design*. São Paulo, Brasil: Cambridge, 2002.
- [20] MITRA, S. K. *Digital Signal Processing: A Computer-Based Approach*. EUA: McGraw-Hill, 2001.
- [21] OPPENHEIM, A. V.; SCHAFER, R. W. *Discrete-Time Signal Processing*. EUA: Prentice-Hall, 1989.
- [22] IEEE Signal Processing Society. <http://www.signalprocessingsociety.org/>, último acesso em 01/07/2008.
- [23] IEEE Institute of Electrical and Electronics Engineers. <http://www.ieee.org/>, último acesso em 01/07/2008.
- [24] INMETRO. *SI - Sistema Internacional de Unidades*. 8a. ed. Brasil: Instituto Nacional de Metrologia, Normalização e Qualidade Industrial, 2007.
- [25] WYSZECKI, G.; STILES, W. S. *Color Science*. 2nd. ed. : Wiley, 2000.
- [26] HUNT, R. W. G. *The Reproduction of Colour*. 6th. ed. EUA: Wiley, 2004.
- [27] RICHARDSON, I. E. G. *Video Codec Design: Developing Image and Video Compression Systems*. EUA: Wiley, 2002.
- [28] SHANNON, C. E. A Mathematical Theory of Communication. *Bell System Technical Journal*, v. 27, pp. 379–423,623–656, Julho, Outubro 1948.
- [29] COVER, T. M.; THOMAS, J. A. *Elements of Information Theory*. 2nd. ed. EUA: Wiley, 2006.
- [30] GERSHO, A.; GRAY, R. M. *Vector Quantization and Signal Compression*. EUA: Kluwer Academic, 1992.
- [31] SAYOOD, K. *Introduction to Data Compression*. 3rd. ed. EUA: Morgan Kaufman, 2006.
- [32] LANGDON, G.; RISSANEN, J. Compression of Black-white Images with Arithmetic Coding. *IEEE Transactions on Communications*, v. 29, n. 6, pp. 858–867, Junho 1981.
- [33] RISSANEN, J. Generalized Kraft Inequality and Arithmetic Coding. *IBM J. Res. Develop.*, v. 20, n. 3, pp. 197–203, Maio 1976.
- [34] PASCO, R. C. *Source Coding Algorithms for Fast Data Compression*. EUA: Stanford University, 1976.
- [35] RISSANEN, J.; LANGDON, G. Arithmetic Coding. *IBM J. Res. Develop.*, v. 23, n. 2, pp. 149–162, Março 1979.

- [36] CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L. *Introduction to Algorithms*. EUA: McGraw-Hill, 1999.
- [37] HUFFMAN, D. A Method for the Construction of Minimum Redundancy Codes. *Proceedings of the IRE*, pp. 1098–1101, Setembro 1951.
- [38] GALLAGHER, R. G. Variations on a Theme by Huffman. *IEEE Transactions on Information Theory*, pp. 668–674, Novembro 1978.
- [39] JPEG/JBIG. <http://www.jpeg.org/>, último acesso em 01/07/2008.
- [40] ISO - International Organization for Standardization. <http://www.iso.org/>, último acesso em 01/07/2008.
- [41] IEC - International Electrotechnical Commission. <http://www.iec.ch/>, último acesso em 01/07/2008.
- [42] ITU - International Telecommunication Union. <http://www.itu.int>, último acesso em 01/07/2008.
- [43] MH/MR. Standardization of Group 3 Facsimile Terminals for Document Transmission. ITU-T Recommendation T.4. Março 1993.
- [44] MMR. Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus. ITU-T Recommendation T.6. Novembro 1988.
- [45] HAMPEL, H. et al. Technical Features of the JBIG Standard for Progressive Bi-level Image Compression. *SP:IC*, v. 4, n. 2, pp. 103–111, Abril 1992.
- [46] REAVY, M. D.; BONCELET, C. G. An Algorithm for Compression of Bilevel Images. *IEEE Transactions on Image Processing*, v. 10, n. 5, pp. 669–676, Maio 2001.
- [47] HOWARD, P. G. Lossless and Lossy Compression of Text Images by Soft Pattern Matching. *Proceedings of Data Compression Conference*, pp. 210–219, Março 1996.
- [48] WITTEN, I. H.; MOFFAT, A.; BELL, T. C. *Managing Gigabytes: Compression and Indexing Documents and Images*. EUA: Van Nostrand Reinhold, 1994.
- [49] PENNEBAKER, W. B. et al. An Overview of the Basic Principles of the Q-coder Adaptive Binary Arithmetic Coder. *IBM J. Res. Develop.*, v. 32, n. 2, pp. 717–726, Novembro 1988.
- [50] KUHN, M. K. JBIG-KIT Portable ANSI C library. <http://www.cl.cam.ac.uk/mgk25/jbigkit/>, último acesso em 01/07/2008.
- [51] HOWARD, P. G. et al. The Emerging JBIG2 Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 8, n. 7, pp. 838–848, Novembro 1998.
- [52] ONO, F. et al. JBIG2 - The Ultimate Bi-level Image Coding Standard. *Proceedings of IEEE International Conference on Image Processing*, pp. 140–143, Setembro. 2000.

- [53] MOHIUDDIN, K.; RISSANEN, J. J.; ARPS, R. Lossless Binary Image Compression Based on Pattern Matching. *Proceedings of International Conference on Computers, Systems, and Signal Processing*, pp. 447–451, Dezembro 1984.
- [54] HOWARD, P. G. Text Image Compression using Soft Pattern Matching. *The Computer Journal*, v. 40, n. 2 and 3, pp. 146–156, Junho 1997.
- [55] TAUBMAN, D. High Performance Scalable Image Compression with EBCOT. *IEEE Transactions on Image Processing*, v. 9, n. 7, pp. 1158–1170, Julho 2004.
- [56] KOSENTINI, F.; TOMPKINS, D. A Fast Segmentation Algorithm for Bi-level Image Compression using JBIG2. *Proceedings of IEEE International Conference on Image Processing*, v. 1, pp. 224–228, Outubro 1999.
- [57] JBIG. JBIG Maui Meeting Press Release: New Document Compression Standard Quadruples Compression of Today’s Fax Standards and Runs at Unprecedented Speeds. *ISO/IEC JTC1/SC29/WG1*, Dezembro 1999.
- [58] POULARIKAS, A. D. *The Transform and Applications Handbook*. 2nd. ed. EUA: IEEE Press, 2000.
- [59] PENNEBAKER, W. B.; MITCHELL, J. L. *JPEG Still Image Data Compression Standard*. : Chapman and Hall, 1993.
- [60] RAO, K. R.; YIP, P. *Discrete Cosine Transform - Algorithms, Advantages, Applications*. EUA: Academic Press, 1990.
- [61] YU, J. Advantages of Uniform Scalar Dead-zone Quantization in Image Coding System. *Proceedings of IEEE International Conference on Communications, Circuits and Systems*, v. 2, pp. 805–808, Junho 2004.
- [62] IJG - Independent JPEG Group. <http://www.ijg.org/>, último acesso em 01/07/2008.
- [63] MALLAT, S. G. A Theory for Multiresolution Signal Decomposition: the Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 11, n. 7, pp. 674–693, Julho 1989.
- [64] DAUBECHIES, I. The Wavelet Transform, Time-frequency Localization and Signal Analysis. *IEEE Transactions on Information Theory*, v. 36, n. 5, pp. 961–1005, Setembro 1990.
- [65] STRANG, G.; NGUYEN, T. *Wavelets and Filter Banks*. Revised. EUA: Wellesley-Cambridge Press, 1997.
- [66] DEVORE, R. A.; JAWERTH, B.; LUCIER, B. J. Image Compression Through Wavelet Transform Coding. *IEEE Transactions on Information Theory*, v. 38, n. 2, pp. 719–746, Março 1992.
- [67] ANTONINI, M. et al. Image Coding using Wavelet Transform. *IEEE Transactions on Image Processing*, v. 1, n. 2, pp. 205–220, Abril 1992.

- [68] LEWIS, A. S.; KNOWLES, G. Image Compression using the 2-D Wavelet Transform. *IEEE Transactions on Image Processing*, v. 1, n. 2, pp. 244–250, Abril 1992.
- [69] TAUBMAN, D. S.; MARCELLIN, M. W. *JPEG 2000: Imagem Compression Fundamentals, Standards and Practice*. EUA: Kluwer Academic, 2002.
- [70] DAUBECHIES, I. *Ten Lectures on Wavelets*. EUA: SIAM, 1992.
- [71] ADAM, M. The JasPer Project Home Page. <http://www.ece.uvic.ca/~mdadams/jasper/>, último acesso em 01/07/2008.
- [72] RICHARDSON, I. E. G. *H.264 and MPEG-4 video Compression*. EUA: Wiley, 2003.
- [73] JPEG. Information Technology - JPEG2000 Image Coding System - Part 3: Motion JPEG 2000. ISO/IEC 15444-3. 2001.
- [74] JPEG. Information Technology - JPEG 2000 Image Coding System: Core Coding System, AMENDMENT 1: Profiles for Digital Cinema Applications. ISO/IEC 15444-1/FDAM 1. 2004.
- [75] BILGIN, A.; MARCELLIN, M. W. JPEG2000 for Digital Cinema. *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 3878–3881, Maio 2006.
- [76] DCI. *Digital Cinema System Specification - Version 1.2*. EUA: Digital Cinema Initiatives, LLC, 2008.
- [77] JVT - Joint Video Team. <http://www.itu.int/ITU-T/studygroups/com16/jvt/index.html>, último acesso em 01/07/2008.
- [78] MPEG - Moving Picture Experts Group. <http://www.mpeg.org/>, último acesso em 01/07/2008.
- [79] VCEG - Video Coding Experts Group. <http://www.itu.int/ITU-T/studygroups/com16/sg16-q6.html>, último acesso em 01/07/2008.
- [80] JVT. Information Technology - Coding of Audio-visual Objects - Part 10: Advanced Video Coding. ISO/IEC 14496-10. 2005.
- [81] WIEGAND, T. et al. Overview of the H.264/AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 13, n. 7, pp. 560–576, Julho 2003.
- [82] WIEGAND, T. et al. Rate-constrained Coder Control and Comparison of Video Coding Standards. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 13, n. 7, pp. 688–703, Julho 2003.
- [83] STOCKHAMMER, T.; HANNUKSELA, M. M.; WIEGAND, T. H.264/AVC in Wireless Environments. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 13, n. 7, pp. 657–673, Julho 2003.
- [84] SULLIVAN, G. J.; TOPIWALA, P.; LUTHRA, A. The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions. *Proceedings of SPIE Conference on Applications of Digital Image Processing XXVII, Special Session on Advances in the New Emerging Standard: H.264/AVC*, v. 5558, pp. 53–74, Agosto 2004.

- [85] OSTERMANN, J. et al. Video Coding with H.264/AVC: Tools, Performance, and Complexity. *IEEE Circuits and Systems Magazine*, v. 4, n. 1, pp. 7–28, (Primeiro quarto) Março 2004.
- [86] HASKELL, B. G.; PURI, A.; NETRAVALLI, A. N. *Digital Video: An Introduction to MPEG-2*. EUA: Chapman and Hall, 1997.
- [87] ITU-T. Video Coding for Low Bit Rate Communication. ITU-T Recommendation H.263. Version 1: Nov. 1995, Version 2: Jan. 1998, Version 3: Nov. 2000.
- [88] QUEIROZ, R. L. de et al. Fringe Benefits of the H.264/AVC. *Proceedings of International Telecommunications Symposium*, pp. 208–212, Setembro 2006.
- [89] H.264/AVC JM Reference Software. <http://iphome.hhi.de/suehring/tml/>, último acesso em 01/07/2008.
- [90] HHI - Fraunhofer Heinrich-Hertz-Institut. <http://www.hhi.fraunhofer.de/>, último acesso em 01/07/2008.
- [91] PARK, J. S.; SONG, H. J. Selective Intra Prediction Mode Decision for H.264/AVC Encoders. *Transactions on Engineering, Computing and Technology*, v. 13, pp. 51–55, Maio 2006.
- [92] SULLIVAN, G. J.; WIEGAND, T. Rate-Distortion Optimization for Video Compression. *IEEE Signal Processing Magazine*, pp. 74–90, Novembro 1998.
- [93] PARK, J. S.; SONG, H. J. Selective Intra Prediction Mode Decision for H.264/AVC Encoders. *Transactions on Engineering, Computing and Technology*, v. 13, pp. 51–55, Maio 2006.
- [94] HUANG, Y.-W. et al. Analysis and Complexity Reduction of Multiple Reference Frames Motion Estimation in H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 16, n. 4, pp. 507–522, Abril 2006.
- [95] LI, H.; NGAN, K. N.; WEI, Z. Analysis and Complexity Reduction of Multiple Reference Frames Motion Estimation in H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 18, n. 6, pp. 756–768, Junho 2008.
- [96] MARPE, D.; SCHWARZ, H.; WIEGAND, T. Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 13, n. 7, pp. 620–636, Julho 2003.
- [97] MARPE, D.; GEORGE, V.; WIEGAND, T. Performance Comparison of Intra-only H.264/AVC and JPEG2000 for a Set of Monochrome ISO/IEC Test Images. *Contribution JVT ISO/IEC MPEG and ITU-T VCEG, Doc. JVT M-014*, Outubro 2004.
- [98] MARPE, D. et al. Performance Evaluation of Motion-JPEG2000 in Comparison with H.264/AVC Operated in Pure Intra Coding Mode. *Wavelet Applications in Industrial Processing*, v. 5266 of Proceedings of SPIE, pp. 129–137, Outubro 2004.
- [99] AL, A. et al. Quality and Complexity Comparison of H.264 Intra Mode with JPEG2000 and JPEG. *Proceedings of IEEE International Conference on Image Processing*, v. 1, pp. 24–27, Outubro 2004.

- [100] REID, G. *Postscript Language: Program Design*. EUA: Addison-Wesley, 1987.
- [101] ADOBE. *PDF Reference - Version 1.4*. 3rd. ed. EUA: Adobe Systems Inc., 2003.
- [102] QUEIROZ, R. L. de. *Compressing Compound Documents, in The Document and Image Compression Handbook*. EUA: by M. Barni, Marcel-Dekker, 2005.
- [103] MARR, D. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. EUA: W. H. Freeman, 1982.
- [104] KONSTANTINIDE, K.; TRETTER, D. A JPEG Variable Quantization Method for Compound Documents. *IEEE Transactions on Image Processing*, v. 9, n. 7, pp. 1282–1287, Julho 2000.
- [105] EXTENSIONS, J. Digital Compression and Coding of Continuous-tone Still Images: Extensions. ITU-T Recommendation T.84. Novembro 1994.
- [106] RAMOS, M.; QUEIROZ, R. de. Classified JPEG Coding of Mixed Documents. *IEEE Transactions on Image Processing*, v. 9, n. 4, pp. 716–720, Abril 2000.
- [107] QUEIROZ, R. L. de; BUCKLEY, R.; XU, M. Mixed Raster Content (MRC) Model for Compound Image Compression. *Proceedings of SPIE Visual Communications and Image Processing*, v. 3653, pp. 1106–1117, Janeiro 1999.
- [108] SEZGIN, M.; SANKUR, B. Survey over Image Thresholding Techniques. *Journal of Electronic Imaging*, v. 13, n. 1, pp. 146–165, Janeiro 2004.
- [109] TRIER, O. D.; TAXT, T. Evaluation of Binarization Methods for Document Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 17, n. 3, pp. 312–315, Março 1995.
- [110] HANEDA, E.; YI, J.; BOUMAN, C. A. Segmentation for MRC Compression. *Proceedings of the SPIE on Color Imaging XII: Processing, Hardcopy, and Applications. Edited by Eschbach, Reiner; Marcu, Gabriel G.*, v. 6493, pp. 252–262, 2007.
- [111] HAFFNER, P. et al. High Quality Document Image Compression with DjVu. *Journal of Electronic Imaging*, v. 7, pp. 410–425, 1998.
- [112] SIMARD, P. Y. et al. A Foreground/Background Separation Algorithm for Image Compression. *Proceedings of the Data Compression Conference*, pp. 498–507, 2004.
- [113] QUEIROZ, R. L. de; FAN, Z.; TRAN, T. D. Optimizing Block-thresholding Segmentation for Multilayer Compression of Compound Images. *IEEE Transactions on Image Processing*, v. 9, n. 9, pp. 1461–1471, Setembro 2000.
- [114] CHENG, H.; BOUMAN, C. A. Document Compression using Rate-distortion Optimized Segmentation. *Journal of Electronic Imaging*, v. 10, n. 2, pp. 460–474, 2001.
- [115] CHENG, H.; FENG, G.; BOUMAN, C. A. Device-Independent Color, Color Hardcopy, and Applications VII. *Proceedings of the SPIE on Color Imaging: De*, v. 4663, pp. 21–23, Janeiro 2002.

- [116] QUEIROZ, R. L. de. On Data-filing Algorithms for MRC Layers. *Proceedings of IEEE International Conference on Image Processing*, pp. 586–589, Setembro 2000.
- [117] LAKHANI, G.; SUBEDI, R. Optimal Filling of FG/BG Layers of Compound Document Images. *Proceedings of IEEE International Conference on Image Processing*, pp. 2273–2276, Outubro 2006.
- [118] FAN, J. Text Extraction via an Edge-bounded Averaging and a Parametric Character Model. *Proceedings of SPIE Document Recognition and Retrieval X*, v. 5010, pp. 8–19, Janeiro 2003.
- [119] RAMAMURTHI, B.; GERSHO, A. Classified Vector Quantization of Images. *IEEE Transactions on Communications*, v. 34, n. 11, pp. 1105–1115, Novembro 1986.
- [120] ZAGHETTO, A.; QUEIROZ, R. de. Segmentation-driven Compound Document Coding based on H.264/AVC-INTRA. *IEEE Transactions on Image Processing*, v. 16, n. 7, pp. 1755–1760, Julho 2007.
- [121] ZAGHETTO, A.; QUEIROZ, R. de. Using H.264/AVC-Intra for Segmentation-driven Compound Document Coding. *Proceedings of IEEE International Conference on Image Processing*, Setembro 2007.
- [122] FENG, G.; BOUMAN, C. A. High-quality MRC Document Coding, volume = 15, year = 2006. *IEEE Transactions on Image Processing*, n. 10, pp. 3152–3169, Outubro.
- [123] MUKHERJEE, D.; CHRYSAFIS, C.; SAID, A. JPEG2000-matched MRC Compression of Compound Documents. *Proceedings of IEEE International Conference on Image Processing*, v. 3, pp. 73–76, Setembro 2002.
- [124] MUKHERJEE, D.; MEMON, N.; SAID, A. JPEG-matched MRC Compression of Compound Documents. *Proceedings of IEEE International Conference on Image Processing*, v. 3, pp. 434–437, Outubro 2001.
- [125] ZAGHETTO, A.; QUEIROZ, R. L. de. Iterative Pre- and Post-processing for MRC Layers of Scanned Documents. *Proceedings of IEEE International Conference on Image Processing*, pp. 1009–1012, Outubro 2008.
- [126] RODRIGUES, N. M. M.; SILVA, E. A. B. da; CARVALHO, M. B. de. Universal Image Coding using Multiscale Recurrent Patterns and Prediction. *Proceedings of IEEE International Conference on Image Processing*, pp. 245–248, Setembro 2005.
- [127] RODRIGUES, N. M. M. et al. Improving Multiscale Recurrent Pattern Image Coding with Enhanced Dictionary Updating Strategies. *Proceedings of International Telecommunications Symposium*, pp. 257–262, Setembro 2006.
- [128] FILHO, E. B. de L. et al. Universal Image Compression Using Multiscale Recurrent Patterns With Adaptive Probability Model. *IEEE Transactions on Image Processing*, v. 17, n. 4, pp. 512–527, Abril 2008.

- [129] FRANCISCO, N. C. et al. Casamento Aproximado de Padrões Multiescala com Segmentação Flexível e Treino do Dicionário. *Anais do Simpósio Brasileiro de Telecomunicações*, Setembro 2008.
- [130] FRANCISCO, N. C. et al. Multiscale Recurrent Pattern Image Coding with a Flexible Partition Scheme. *Proceedings of IEEE International Conference on Image Processing*, pp. 141–144, Outubro 2008.