



TESE DE DOUTORADO EM ENGENHARIA
ELÉTRICA

**CODIFICADOR DISTRIBUÍDO DE VÍDEO COM
COMPLEXIDADE VARIÁVEL A PARTIR DE
CODIFICAÇÃO EM RESOLUÇÃO ESPACIAL MISTA**

Bruno Luigi Macchiavello Espinoza

Brasília, Abril de 2009

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

CODIFICADOR DISTRIBUÍDO DE VÍDEO COM
COMPLEXIDADE VARIÁVEL A PARTIR DE
CODIFICAÇÃO EM RESOLUÇÃO ESPACIAL MISTA

Bruno Luiggi Macchiavello Espinoza

ORIENTADOR: Ricardo Lopes de Queiroz

CO-ORIENTADOR: Debargha Mukherjee

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA

Publicação: PPGENE.TD 036/09

Brasília/DF: Abril- 2009

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TESE DE DOUTORADO EM ENGENHARIA
ELÉTRICA

**CODIFICADOR DISTRIBUÍDO DE VÍDEO COM
COMPLEXIDADE VARIÁVEL A PARTIR DE
CODIFICAÇÃO EM RESOLUÇÃO ESPACIAL MISTA**

Bruno Luigi Macchiavello Espinoza

Tese de doutorado submetida ao Departamento de Engenharia Elétrica da Faculdade de Tecnologia da Universidade de Brasília, como parte dos requisitos necessários para a obtenção do grau de Doutor.

Banca Examinadora

Ricardo Lopes de Queiroz, PhD.
UnB/ ENE (Orientador)

Francisco Assis de O. Nascimento, Dr.
UnB/ ENE (Examinador Interno)

Eduardo A. Barros da Silva, PhD.
UFRJ/ COPPE (Examinador Externo)

Joao Souza Neto, Dr.
UCB/ MGCTI (Examinador Externo)

Adolfo Bauchspiess, Dr-Ing
UnB/ ENE (Examinador Interno)

FICHA CATALOGRÁFICA

MACCHIAVELLO, BRUNO L. E.

Codificador Distribuído de Vídeo com complexidade variável a partir de codificação em resolução espacial mista. [Distrito Federal] 2009.

xxii, 131p., 297 mm (ENE/FT/UnB, Doutor, Telecomunicações Processamento de Sinais, 2008). Tese de Doutorado.

Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Elétrica.

1. Codificação Distribuída de Fonte

2. Wyner-Ziv

3. Codificação Distribuída de Vídeo

4. Codificação de Baixa Complexidade

5. H.264/AVC

6. Semi super resolução

I. ENE/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

MACCHIAVELLO, B. L. E. (2009). Codificador Distribuído de Vídeo com complexidade variável a partir de codificação em resolução espacial mista. Tese de Doutorado em Engenharia Elétrica com ênfase em Telecomunicações, Publicação MTARH.DM - 326 A/08, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 131p.

CESSÃO DE DIREITOS

NOME DO AUTOR: Bruno Luigi Macchiavello Espinoza.

TÍTULO DA TESE DE DOUTORADO: Codificador Distribuído de Vídeo com complexidade variável a partir de codificação em resolução espacial mista.

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 dissertação de mestrado pode ser reproduzida sem a autorização por escrito do autor.

Bruno Luigi Macchiavello Espinoza
SQN 216 Bloco B Apto 322
70875-020 Asa Norte, Brasília - DF - Brasil.

Dedicatória

*À minha esposa Nubia, a meus irmãos Julissa e Alvaro, a meus pais Walter e Emmy, e
à minha avó Rosa*

Bruno Luigi Macchiavello Espinoza

Agradecimentos

Agradeço a meus pais Walter e Emmy, e a meu irmão Alvaro, que mesmo longe sempre estão presentes na minha vida. Agradeço especialmente a minha irmã Julissa, que me ajudou muito durante o período do meu doutorado e mestrado. Agradeço a minha esposa Nubia por fazer parte da minha vida. Ao meu orientador, Professor Ricardo Lopes de Queiroz, por sua ajuda, críticas e sugestões ao trabalho. Sem ele o trabalho não teria alcançado o nível desejado. Ao meu co-orientador Doutor Debargha Mukherjee, que contribuiu com idéias, apoio e me ajudou muito durante minha estadia em Palo Alto, USA. Ao Professor Francisco Assis de Oliveira Nascimento que foi meu orientador no mestrado, e abriu as portas da Universidade de Brasília para mim. Ao pessoal da HP Brasil, Ricardo Pianta, Paulo Sá e Marcelo Thielo, que todo ano apóiam o nosso grupo de pesquisa. Aos companheiros do grupo da HP, Rafael “Capim”, Fernanda Brandi, Alexandre Zaghetto, Eduardo “Grilo” Peixoto, Edson Mintsu, Tiago Alvez, Karen Oliveira, Renan Utida, Diogo “Buraco”, que me ajudaram muito durante a minha pesquisa, e compartilhamos risadas, jogos de vídeo game e sessões de BANG. Agradecimentos especiais a Mintsu e Grilo pelas revisões. Aos companheiros do GPDS Rafael Ortis, Maria do Carmo, Frederico “Aspira” Nogueira, Chaffim “Rei do Mingau”, Patrick, Alberto, Marcelo e Camilo que formam um excelente ambiente de trabalho. Aos meus amigos Fabio Paião, João Baptista e Christian “Magrão” simplesmente por sua amizade. Aos meus amigos de Brasília Rafael “Metal”, Alecio, Rodolfo, Flavio Vidal, Barbosa, Marcelino e todos os que esqueci mencionar. A Eliane, Edilene, Carlos e Denis. Aos meus amigos peruanos Sergio, Marco Carrion, Joel, Miguel, Burga, Miguelón, Victor, Marco “Fracasado”, Pavo, Renzo, Moto, José Balbuena, Wong, Bruno Tarzona, Francisco Orbegozo, Juan Bertolotti e todos os que posso ter esquecido mencionar. A meus ex-colegas de trabalho do MTC, Julissa, Karem, Marisol e José. Por fim, ao pessoal da banda MARBRU, Marcelo Portela e Marcelo “Baqueta”, pelas músicas que fazemos juntos.

Bruno Luigi Macchiavello Espinoza

RESUMO

Recentemente, novas exigências têm surgido para a codificação de vídeo. Estas exigências incluem flutuação da largura de banda, qualidade de serviço e limitações de energia. Em dispositivos móveis é importante limitar o consumo de energia. Tipicamente, o codificador possui uma maior complexidade, requerendo maior consumo de energia que o decodificador. Um novo paradigma, chamado codificação de vídeo distribuída, possibilita a codificação com complexidade reversa, onde o codificador requer um menor esforço computacional do que o decodificador. A codificação de vídeo distribuída é baseada no teorema de Wyner-Ziv para a codificação separada, com decodificação conjunta de fontes correlatas. Assim, um codec distribuído pode ser utilizado em cenários de potência limitada.

É proposto um modo de codificação baseado em quadros-chave de resolução espacial completa e quadros intermediários codificados a resolução reduzida mediante um codificador Wyner-Ziv. Um bom desempenho de taxa-distorção é alcançado mediante uma melhor geração da informação lateral no decodificador e um mecanismo automático de alocação de taxa no codificador. Este modo possibilita a redução de complexidade de codificação dos quadros intermediários, seguido de codificação Wyner-Ziv do resíduo. Os coeficientes quantizados do quadro residual são mapeados em *cosets* sem o uso de um canal de retorno. Para isto, foi feito um estudo dos parâmetros ótimos de codificação na criação dos *cosets* sem memória. Também, foi elaborado um mecanismo de estimação da correlação estatística entre os sinais. Este mecanismo guia a escolha dos parâmetros de codificação e a alocação de taxa, durante o processo da criação dos *cosets*.

A geração da informação lateral explora a informação obtida da camada base de baixa resolução. No decodificador, a decodificação de canal dos *cosets* é realizada usando a informação lateral para obter uma versão de alta qualidade do quadro intermediário decodificado. Resultados da complexidade de codificação e do desempenho, em termos de taxa-distorção, são apresentados usando o padrão H.264/AVC. É mostrado que o modo de codificação Wyner-Ziv proposto é competitivo ao ser comparado com a codificação convencional. O modo Wyner-Ziv proposto também possui adaptabilidade na redução de complexidade e suporta um modo de decodificação de baixa complexidade.

ABSTRACT

Recently, new requirements in video coding have emerged. These requirements include bandwidth fluctuation, quality of service and energy constraints. In handheld mobile devices, it is important to limit the energy consumption. Typically, the encoder has a higher complexity, requiring more energy consumption than the decoder. A new paradigm in video coding, called distributed video coding, enables a reversed complexity coding mode, where the decoder requires more computational effort than the encoder. Distributed video coding is based on the Wyner-Ziv theorem for separately coding but jointly decoding correlated sources. A distributed video codec can be used in power constrained scenarios.

We propose a mixed resolution framework based on full resolution key frames and spatial-reduction-based Wyner-Ziv coding of intermediate non-reference frames. Improved rate-distortion performance is achieved by enabling better side-information generation at the decoder and better rate-allocation at the encoder. The framework enables reduced encoding complexity by low-resolution encoding of the non-reference frames, followed by Wyner-Ziv coding of the residue. The quantized transform coefficients of the residual frame are mapped to cosets without the use of a feedback channel. For that purpose, a study to select optimal coding parameters in the creation of the memoryless cosets is made. Furthermore, a correlation estimation mechanism that guides the parameter choice process is proposed. This estimation algorithm is one of the main contributions of this work.

The side information generation method exploits information for the low resolution coded frames. At the decoder, coset decoding is carried using the side-information to obtain a higher quality version of the decoded frame. Results for the coding complexity and rate-distortion performance are presented using the H.264/AVC codec. It is shown that the proposed Wyner-Ziv coding mode is competitive. It allows a scalable complexity reduction and supports a low-decoding-complexity mode.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CODIFICAÇÃO DE VÍDEO	1
1.2	APRESENTAÇÃO DO PROBLEMA	3
1.3	ORGANIZAÇÃO	5
2	CODIFICAÇÃO DE FONTE COM INFORMAÇÃO LATERAL	7
2.1	CONCEITOS BÁSICOS DE TEORIA DA INFORMAÇÃO	7
2.2	CODIFICAÇÃO DE FONTE	9
2.3	CODIFICAÇÃO DE CANAL	10
2.4	CODIFICAÇÃO DE FONTE COM INFORMAÇÃO LATERAL	11
2.4.1	CODIFICAÇÃO SLEPIAN-WOLF	11
2.4.2	CODIFICAÇÃO WYNER-ZIV	13
3	PARADIGMAS DE CODIFICAÇÃO DE VÍDEO DIGITAL	15
3.1	CODIFICAÇÃO HÍBRIDA DE VÍDEO	15
3.2	CODIFICAÇÃO DISTRIBUÍDA DE VÍDEO	20
3.2.1	ARQUITETURA STANFORD	21
3.2.2	ARQUITETURA PRISM	24
3.3	VANTAGENS E DESVANTAGENS DA CODIFICAÇÃO DE VÍDEO DISTRIBUÍDA	27
4	CODIFICADOR DISTRIBUÍDO COM RESOLUÇÃO ESPACIAL MISTA ...	31
4.1	RESOLUÇÃO MISTA	32
4.2	ARQUITETURA DO CODEC	34
4.2.1	CODIFICADOR	34
4.2.2	DECODIFICADOR	36
4.3	CODIFICADOR WYNER-ZIV	37
4.4	SEMI SUPER RESOLUÇÃO	42
5	ESTIMAÇÃO DA CORRELAÇÃO ESTATÍSTICA PARA CODIFICAÇÃO DISTRIBUÍDA	47
5.1	MODELO ESTATÍSTICO	47
5.2	ESTIMAÇÃO DA CORRELAÇÃO ESTATÍSTICA	48
5.2.1	ESTIMAÇÃO DA VARIÂNCIA DOS COEFICIENTES RESIDUAIS LAPLACIANOS	49
5.2.2	ESTIMAÇÃO DO FATOR DE ATENUAÇÃO	50
5.2.3	ESTIMAÇÃO DA VARIÂNCIA DO RUÍDO GAUSSIANO	52
5.3	ESCOLHA DOS PARÂMETROS DE CODIFICAÇÃO	53
5.3.1	CODIFICAÇÃO POR <i>cosets</i> SEM MEMÓRIA SEGUIDA DE RECONSTRUÇÃO MSE COM INFORMAÇÃO LATERAL	54
5.3.2	FONTE LAPLACIANA COM RUÍDO ADITIVO GAUSSIANO	56
5.4	SELEÇÃO ÓTIMA DOS PARÂMETROS DE CODIFICAÇÃO	57
5.4.1	CORRESPONDÊNCIA DE DISTRORÇÃO	60
6	CODIFICAÇÃO DA CAMADA DE REALCE	63
6.1	CODIFICAÇÃO DE ENTROPIA NO H.264	64

6.1.1	CODIFICAÇÃO EXP-GOLOMB	64
6.1.2	SEQÜÊNCIAS DE ZEROS	66
6.1.3	CAVLC	66
6.2	CODIFICAÇÃO DA CAMADA WYNER-ZIV	68
6.2.1	D-CAVLC	69
7	EXPERIMENTOS	73
7.1	SEQÜÊNCIAS DE VÍDEOS PARA TESTES	74
7.2	DESEMPENHO DO D-CAVLC	75
7.3	ANÁLISE DE COMPLEXIDADE DE CODIFICAÇÃO	76
7.4	ESTIMAÇÃO DA CORRELAÇÃO ESTATÍSTICA	79
7.5	DESEMPENHO DO MODO DE CODIFICAÇÃO WYNER-ZIV COM RESOLUÇÃO MISTA	80
7.5.1	COMPARAÇÃO COM CODIFICAÇÃO CONVENCIONAL HÍBRIDA.....	80
7.5.2	COMPARAÇÃO COM OUTRAS ARQUITETURAS DVC.....	87
8	CONCLUSÕES	95
8.1	APRESENTAÇÃO DAS CONTRIBUIÇÕES DO DOUTORADO	96
8.2	PERSPECTIVAS PARA TRABALHOS FUTUROS	98
	REFERÊNCIAS BIBLIOGRÁFICAS.....	100
	APÊNDICES.....	111
I	CARACTERIZAÇÃO DAS FUNÇÕES DE TAXA E DISTORÇÃO	113
I.1	CODIFICAÇÃO POR <i>cosets</i> SEM MEMORIA SEGUIDA DE RECONSTRUÇÃO MSE COM INFORMAÇÃO LATERAL.....	113
I.2	CODIFICAÇÃO A TAXA ZERO SEGUIDA DE RECONSTRUÇÃO MSE COM INFORMAÇÃO LATERAL	115
I.3	CODIFICAÇÃO SLEPIAN-WOLF IDEAL COM RECONSTRUÇÃO MSE E INFORMAÇÃO LATERAL	116
I.4	CODIFICAÇÃO CONVENCIONAL SEGUIDA DE RECONSTRUÇÃO MSE COM E SEM INFORMAÇÃO LATERAL	117
II	TABELAS DO D-CAVLC	119
II.1	TABELAS DE CODIFICAÇÃO PARA O NÚMERO DE COEFICIENTES E <i>trailing ones</i>	119
II.1.1	GRUPO I	119
II.1.2	GRUPO II	121
II.1.3	GRUPO III	123
II.2	TABELAS DE CODIFICAÇÃO PARA O NÚMERO TOTAL DE ZEROS.....	125
III	PRODUÇÃO BIBLIOGRÁFICA	129
III.1	PUBLICAÇÕES EM REVISTA	129
III.1.1	CODIFICADOR DISTRIBUÍDO COM RESOLUÇÃO MISTA	129
III.1.2	OUTRAS CONTRIBUIÇÕES	129
III.2	PUBLICAÇÕES EM CONGRESSO	129
III.2.1	CODIFICADOR DISTRIBUÍDO COM RESOLUÇÃO MISTA	129
III.2.2	OUTRAS CONTRIBUIÇÕES	130

LISTA DE FIGURAS

1.1	Elementos de uma sequência de vídeo digital: <i>pixels</i> e quadros.	2
2.1	Compressão de fonte sem perdas. A fonte X deve ser transmitida com não menos que $H(X)$ bits.	9
2.2	Codificação conjunta de duas fontes X e Y . Uma taxa não inferior a $H(X, Y)$ é suficiente para reconstrução sem perdas.	11
2.3	Codificação separada de duas fontes, com decodificação conjunta. $H(X, Y)$ continua sendo suficiente para reconstrução sem perdas.	12
2.4	Região de possível reconstrução sem perda. A região triangular indica a região de codificação Slepian-Wolf.	13
2.5	Codificação de fonte sem perdas com informação lateral.	13
2.6	Codificação de fonte com perdas com informação lateral.	14
3.1	Esquema básico de um codificador híbrido.	17
3.2	Esquema básico de um decodificador híbrido.	17
3.3	Exemplos de codificação preditiva e bi-preditiva. É mostrada a ordem de exibição e codificação, as setas indicam quais quadros são usados como referência da codificação do quadro fonte.	19
3.4	Diagrama de blocos de codec DVC de Stanford.	22
3.5	Diagrama de Blocos do PRISM. (a) codificador, (b) decodificador.	25
4.1	Usos diferentes dos quadros NRWZ. (a) um quadro intermediário tipo B , (b) dois quadros intermediários tipo B , (c) um quadro tipo P	33
4.2	Três cenários de codificação adaptável. (a) adaptabilidade temporal, (b) adaptabilidade temporal e espacial que o SVC permite, (c) resolução mista.	35
4.3	Aquitetura do codificador para os quadros NRWZ.	36
4.4	Aquitetura do decodificador para os quadros NRWZ.	38
4.5	Funções massa de probabilidade. A função $f_X(x)$ é a fdp dos coeficientes transformados residuais modelada como distribuição laplaciana. $p(q)$ é calculada para cada valor possível de q . $p_C(c) = \psi(QP_{wz}, M)$ depende de $f_X(x)$ e a quantidade de valores possíveis é determinada por M , neste caso $M = 5$	40
4.6	Exemplo de Decodificação. O índice recebido é $c = 2$, a informação lateral $y = -1$. Logo, o valor reconstruído \hat{x} dadas todas as possibilidades onde o valor q gerou o índice $c = 2$, para as funções de massa de probabilidade exemplificadas, é igual a -3	42
4.7	Procura da melhor predição. Depois de seleccionar o melhor predito nos quadros filtrados, a alta frequência do bloco é adicionada.	45
4.8	Iteração do processo de semi super resolução.	46
4.9	Comparação entre interpolação e semi super resolução. (a) Quadro interpolado com PSNR 34,12 dB. (b) Quadro em semi super resolução com PSNR 36,775 dB. A melhora é visualmente perceptível.	46
5.1	As primeiras sete bandas de frequência em um bloco 8×8 de coeficientes transformados.	48

5.2	Estimação da variância dos coeficientes residuais Laplacianos. Valores reais de σ_x^2/QP_t^2 vs. E e sua aproximação linear usando H.263+ para 4 bandas de frequência.	51
5.3	Estimação da variância dos coeficientes residuais Laplacianos. Valores reais de σ_x^2/QP_t^2 vs. E e sua aproximação linear usando H.264/AVC para 6 bandas de frequência.	52
5.4	Estimação da variância do ruído Gaussiano. Valores reais de σ_X^2/σ_Z^2 vs. R_n e sua aproximação linear usando H.263+ para 4 bandas de frequência.	54
5.5	Curvas de RD obtidas para calcular o casco convexo inferior. Mostrando as curvas de codificação ideal Slepian-Wolf (SW), codificação convencional sem <i>cosets</i> com reconstrução ótima, e: (a) curvas RD com M constante e variando QP_{wz} , (b) curvas de RD com QP_{wz} constante e variando M	59
6.1	Ordem zig-zag de um bloco 4×4	66
7.1	Quadros de exemplo das seqüências utilizadas. (a) Baixo Movimento, da esquerda para direita: <i>Akiyo</i> , <i>Container</i> e <i>Hall Monitor</i> . (b) Movimento Médio, da esquerda para direita: <i>Mother-Daughter</i> e <i>Silent</i> . (c) Alto Movimento, da esquerda para direita: <i>Coastguard</i> , <i>Foreman</i> , <i>Mobile</i> e <i>Soccer</i>	76
7.2	Resultados de PSNR média para a componente de luminância da seqüência <i>Foreman</i> no modo <i>IbPbP</i> . Comparando H.264 convencional no modo <i>IBPBP</i> e o modo de codificação Wyner-Ziv depois de 1,2 e 3 iterações.	80
7.3	Resultados de PSNR média para a componente de luminância para seqüências de baixo e médio movimento no modo <i>IbPbP</i> . Comparando H.264 convencional no modo <i>IBPBP</i> , o modo de codificação Wyner-Ziv <i>IbPbP</i> depois de 3 iterações, a camada base e os quadros chave junto com os quadros SSR. (a) <i>Akiyo</i> , (b) <i>Hall Monitor</i> e (c) <i>Silent</i>	81
7.4	Resultados de PSNR para a componente de luminância para seqüências de alto movimento no modo <i>IbPbP</i> . Comparando H.264 convencional no modo <i>IBPBP</i> , o modo de codificação Wyner-Ziv depois de 3 iterações, a camada base e os quadros chave junto com os quadros SSR. (a) <i>Foreman</i> , (b) <i>Coastguard</i> , (c) <i>Soccer</i> e (d) <i>Mobile</i>	82
7.5	Resultados de PSNR para a componente de luminância e crominância para seqüências de baixo e médio movimento no modo <i>IbPbP</i> . Comparando H.264 convencional, o modo de codificação Wyner-Ziv depois de 3 iterações e a camada base. (a) <i>Akiyo</i> , (b) <i>Hall Monitor</i> e (c) <i>Silent</i>	84
7.6	Resultados de PSNR para a componente de luminância e crominância para seqüências de alto movimento no modo <i>IBPBP</i> . Comparando H.264 convencional, o modo de codificação Wyner-Ziv depois de 3 iterações e a camada base. (a) <i>Foreman</i> , (b) <i>Coastguard</i> , (c) <i>Soccer</i> e (d) <i>Mobile</i>	85
7.7	Resultados de PSNR para a componente de luminância no modo <i>IpPpP</i> . Comparando H.264 convencional, o modo de codificação Wyner-Ziv depois de 3 iterações, a camada base e os quadros chave junto com os quadros SSR. (a) <i>Container</i> - baixo movimento, (b) <i>Mother-Daughter</i> - movimento médio, (c) <i>Foreman</i> - alto movimento	86
7.8	Resultados de PSNR para a componente de luminância e crominância no modo <i>IpPpP</i> . Comparando H.264 convencional, o modo de codificação Wyner-Ziv depois de 3 iterações e a camada base. (a) <i>Container</i> - baixo movimento, (b) <i>Mother-Daughter</i> - movimento médio, (c) <i>Foreman</i> - alto movimento	87

7.9	Resultados de PSNR para a componente de luminância no modo <i>IbIbI</i> . Comparando H.264 convencional no modo <i>IBIBI</i> , o modo de codificação Wyner-Ziv depois de 3 iterações e a camada base e os quadros chave junto com os quadros SSR. (a) <i>Hall Monitor</i> - baixo movimento, (b) <i>Coastguard</i> - alto movimento, (c) <i>Foreman</i> - alto movimento	88
7.10	Resultados de PSNR para a componente de luminância e crominância no modo <i>IbIbI</i> . Comparando H.264 convencional no modo <i>IBIBI</i> ; o modo de codificação Wyner-Ziv depois de 3 iterações; e a camada base. (a) <i>Hall Monitor</i> - baixo movimento, (b) <i>Coastguard</i> - alto movimento, (c) <i>Foreman</i> - alto movimento	89
7.11	Resultados de PSNR para a componente de luminância da seqüência <i>Foreman</i> no modo <i>IbbPbbP</i> . Comparando H.264 convencional e o modo de codificação Wyner-Ziv.	90
7.12	Curvas de PSNR para a componente <i>Y</i> . Comparando o modo de codificação com resolução mista com outra arquitetura de DVC: (a) <i>Foreman</i> CIF; (b) <i>Hall Monitor</i> CIF, (c) <i>Soccer</i> CIF	91
7.13	Resultados do processo de semi super resolução para quatro seqüências diferentes. Comparando a semi super resolução para quadros <i>b</i> usando quadros-chave tipo <i>I</i> e tipo <i>P</i> , e a interpolação SE-B para os mesmos quadros chave.	93
I.1	Codificação de cosets con reconstrução MSE.	115
I.2	Codificação a taxa zero e reconstrução MSE.	116
I.3	Codificador Slepian-Wolf ideal com reconstrução MSE.	117
I.4	Codificação convencional com reconstrução MSE.....	118

LISTA DE TABELAS

5.1	Valores do fator de atenuação ρ para um bloco 4×4	51
5.2	Tabela de mapeamento para $\sigma_Z=1.1$	62
6.1	Códigos Exp-Golomb	65
6.2	Tabela de Codificação (Primeiro Grupo, $1 \leq max_C \leq 3$).....	70
7.1	Seqüências de vídeos utilizadas nos experimentos	75
7.2	Ganho da taxa de bits do <i>bit-stream</i> da camada Wyner-Ziv	75
7.3	Tempo de codificação do H.264/AVC codec. (EM: Estimaco de Movimento)	77
7.4	Tempo de codificaco do modo com resoluo mista no H.264/AVC (EM: Estimaco de Movimento)	78
7.5	Tempo de codificaco do modo com resoluo mista no H.264/AVC (EM: Estimaco de Movimento, FPS: <i>frame per second</i>).....	79
II.1	Tabela de Codificaco, grupo I, $10 < max_C \leq 16$. T1s significa nmero de <i>Trailing ones</i>	119
II.2	Tabela de Codificaco, grupo I, $6 < max_C \leq 10$	120
II.3	Tabela de Codificaco, grupo I, $3 < max_C \leq 6$	120
II.4	Tabela de Codificaco, grupo I, $1 < max_C \leq 3$	120
II.5	Tabela de Codificaco, grupo I, $0 \leq max_C \leq 1$	120
II.6	Tabela de Codificaco, grupo II, $10 < max_C \leq 16$	121
II.7	Tabela de Codificaco, grupo II, $6 < max_C \leq 10$	122
II.8	Tabela de Codificaco, grupo II, $3 < max_C \leq 6$	122
II.9	Tabela de Codificaco, grupo II, $1 < max_C \leq 3$	122
II.10	Tabela de Codificaco, grupo II, $0 \leq max_C \leq 1$	122
II.11	Tabela de Codificaco, grupo III, $10 < max_C \leq 16$	123
II.12	Tabela de Codificaco, grupo III, $6 < max_C \leq 10$	124
II.13	Tabela de Codificaco, grupo III, $3 < max_C \leq 6$	124
II.14	Tabela de Codificaco, grupo III, $3 < max_C \leq 6$	124
II.15	Tabela de Codificaco, grupo III, $0 \leq max_C \leq 1$	124
II.16	Cdigos quando a quantidade Total de zeros mxima igual a 1.	125
II.17	Cdigos quando a quantidade Total de zeros mxima igual a 2.	125
II.18	Cdigos quando a quantidade Total de zeros mxima igual a 5.	125
II.19	Cdigos quando a quantidade Total de zeros mxima igual a 9.	126
II.20	Cdigos quando a quantidade Total de zeros mxima igual a 15.	127

LISTA DE SIGLAS, ABREVIACÕES E ACRÔNIMOS

Abreviações, Acrônimos e Siglas

BF	Banda de Frequência
CABAC	<i>Context-Based Adaptive Binary Arithmetic Coding</i>
CAVLC	<i>Context-Based Adaptive Variable Length Coding</i>
CBP	<i>Coded Block Pattern</i>
CODEC	Codificador e Decodificador
CRC	<i>cyclic redundancy check</i>
DCT	<i>Discrete Cosine Transform</i>
DPCM	<i>Differential Pulse Code Modulation</i>
DSC	<i>Distributed Source Coding</i>
DVC	<i>Distributed Video Coding</i>
EM	Estimação de Movimento
FDP	<i>Função Densidade de Probabilidade</i>
FGS	<i>Fine Granular Scalability</i>
H.26X	Padrões de compressão de vídeo da <i>International Telecommunication Union</i>
H.261	Padrão de Compressão de vídeo
H.263	Padrão de Compressão de vídeo
H.263+	Padrão de Compressão de vídeo (versão melhorada do H.263)
H.264/AVC	Padrão de Compressão de vídeo (Advanced Video Coding)
H.26L	Codificador implementado entre os padrões H.263+ e H.264/AVC
IDCT	<i>Inverse Discrete Cosine Transform</i>
ITU	<i>International Telecommunication Union</i>
LPDC	<i>Low-Density Parity-Check Codes</i>
MPEG	<i>Motion Picture Experts Group</i>
MPEG-1	Padrão de Compressão de vídeo
MPEG-2	Padrão de Compressão de vídeo
MPEG-4	Padrão de Compressão de vídeo
MR-DVC	<i>Mixed Resolution Distributed Video Codec</i>
MSE	<i>Mean Square Error</i>
MV	<i>Motion Vectors</i>
NRWZ	<i>non-reference Wyner-Ziv</i>
PDF	<i>Probability Density Function</i>
PRISM	<i>Power-efficient, robust, high-compression syndrome based multimedia coding</i>
PSNR	<i>Peak Signal to Noise Ratio</i>
QoS	<i>Quality of Service</i>
QWZ	Quadros Wyner-Ziv
RD	<i>Rate-Distortion</i>
RDO	<i>Rate-Distortion Optimization</i>
SAD	<i>sum of absolute differences</i>
SI	<i>Side Information</i>
SNR	<i>Signal to Noise Ratio</i>
SSR	Semi Super Resolução
SVC	<i>Scalable Video Coder</i>
SW	Slepian-Wolf
TV	Televisão
WZ	Wyner-Ziv

1 INTRODUÇÃO

1.1 CODIFICAÇÃO DE VÍDEO

A compressão de vídeo digital é necessária para se realizar uma codificação eficiente, seja para armazenamento ou transmissão do sinal de vídeo. O objetivo é manter uma qualidade de reconstrução que depende da aplicação, ao mesmo tempo reduzindo ao máximo a quantidade de dados (bits) armazenados ou transmitidos.

Uma seqüência de vídeo natural é um sinal contínuo no tempo e espaço. O vídeo digital é composto por sinais amostrados temporal e espacialmente. Uma vez digitalizado, o vídeo possui uma disposição tridimensional. Duas dimensões indicam a disposição espacial (horizontal e vertical), e a terceira dimensão representa o domínio do tempo. Os elementos que compõem uma imagem espacialmente são denominados de *pixels*, do inglês *picture elements*. Os *pixels* possuem as informações de brilho e cor (luminância e crominância, respectivamente). No sistema de transmissão de cores *YUV*, o valor de *Y* (luminância) representa a componente de brilho. Este sinal pode ser reproduzido sem os demais componentes, gerando uma versão monocromática ou em escalas de cinza da imagem. Os componentes *U* e *V* (crominância) possuem a informação de saturação e matiz das cores [1].

Ao amostrar temporalmente uma seqüência de vídeo cria-se o que é denominado quadro ou *frame*. Um quadro representa o conjunto de todos os *pixels* que correspondem a um único instante de tempo. Basicamente, um quadro é o mesmo que um retrato estático. A digitalização de um sinal de vídeo é mostrada na Figura 1.1. A quantidade de *pixels* em cada quadro define a resolução espacial do vídeo, enquanto a quantidade de quadros por segundo define a resolução temporal.

A compressão de vídeo pode ser definida como a redução da taxa de dados necessários para codificar uma seqüência de quadros. Os métodos de compressão podem ser classificados como: compressão com perdas, ou compressão sem perdas. A compressão sem perdas significa comprimir os dados sem rejeitar ou alterar nenhuma informação presente neles. A compressão com perdas tenta eliminar a informação irrelevante e redundante existente nos sinais de vídeo.

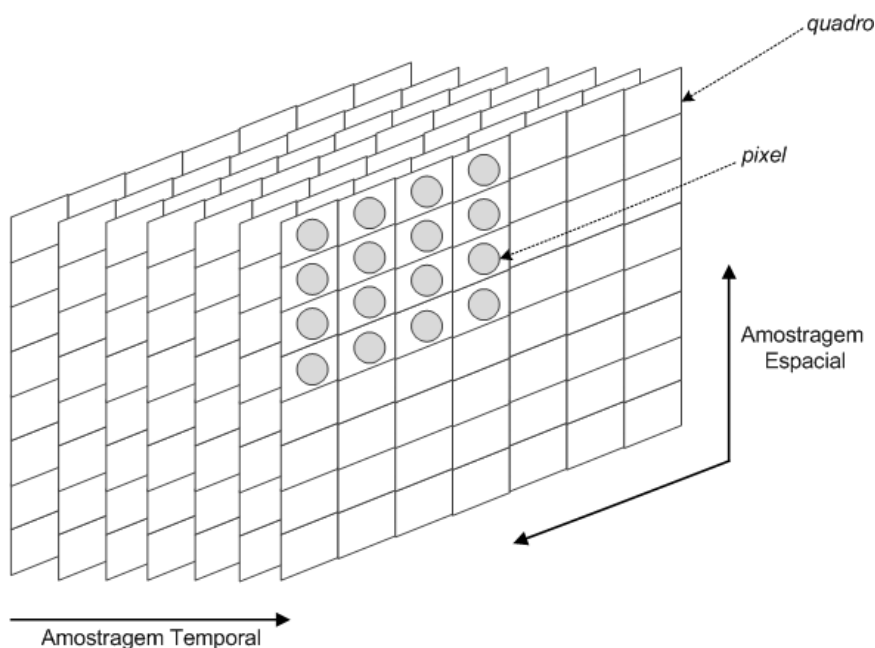


Figura 1.1: Elementos de uma sequência de vídeo digital: *pixels* e quadros.

Dependendo da taxa de compressão desejada, a quantidade de informação rejeitada aumenta ou diminui, alterando a qualidade do sinal. A redundância pode ser do tipo espacial ou temporal. Assim, a compressão de vídeo reduz tipicamente a redundância espacial utilizando técnicas de compressão da imagem em cada quadro. Isto é conhecido como a compressão *intra*-quadros. Por outro lado, a redução da redundância temporal é feita tipicamente mediante técnicas de compensação de movimento, conhecida como compressão *inter*-quadros.

Atualmente, a maioria dos padrões técnicos de compressão de vídeo são projetados para aplicações do tipo radiodifusão (*broadcasting*), onde existe um único codificador potente e vários decodificadores de baixa complexidade. Alguns padrões técnicos são concebidos para situações específicas como codificação a baixa taxa para videoconferência [2]. Os padrões de codificação de vídeo como MPEG [3, 4, 5] e H.26X [2, 6] atingem altas taxas de compressão a partir do uso da transformada discreta de cosseno em blocos e codificação preditiva *inter*-quadro. Tipicamente, o codificador requer um esforço computacional muito maior que o decodificador [6] devido principalmente à função de estimação de movimento. Portanto, eles são adequados para comunicações de vídeo onde a codificação é feita apenas uma vez e a decodificação é realizada várias vezes.

No entanto, novos requisitos na codificação de vídeo digital têm surgido. Requisitos como

flutuação de banda, restrição de energia e qualidade de serviço (QoS) podem chegar a ser tão importantes como a taxa de compressão. A necessidade de satisfazer diferentes exigências de decodificação faz com que a transcodificação seja necessária em muitas situações. Devido ao fato de o transcodificador requerer alta capacidade computacional, a necessidade de um codec (codificador e decodificador) de vídeo adaptável tem aumentado. Existe uma extensão com adaptabilidade na relação sinal-ruído (*signal to noise ratio* ou SNR) para o padrão H.264/AVC [7, 8, 9]. Esta extensão é conhecida como *scalable video coder* (SVC), e pode atingir algumas das novas exigências de codificação. Outros codificadores com SNR adaptável foram apresentados anteriormente [10, 11, 12], e há também trabalhos para obter e avaliar os limites teóricos do desempenho em taxa-distorção, em inglês *rate-distortion* (RD), para algoritmos de compressão de vídeo com taxa de bits adaptável [13].

A codificação distribuída de vídeo, em inglês *distributed video coding* (DVC), viabiliza a criação de codecs com complexidade reversa. Na codificação de vídeo, e especificamente neste trabalho, a complexidade se refere ao esforço computacional em termos de operações realizadas para comprimir uma seqüência de vídeo. Os codecs com complexidade reversa possuem baixa complexidade no codificador e alta complexidade de decodificação, ao contrário da codificação convencional, orientada a codecs de radiodifusão. Logo, na codificação distribuída, o decodificador requer maior esforço computacional que o codificador. Assim, o paradigma DVC é adequado para aparelhos com restrição de potência que capturam e codificam vídeo em tempo real, tanto para transmissão ou armazenamento, como redes de sensores, telefones móveis com vídeo chamada e aparelhos portáteis, por exemplo.

1.2 APRESENTAÇÃO DO PROBLEMA

A codificação distribuída de fonte, em inglês *distributed source coding* (DSC), é baseada na teoria de codificação de fontes correlatas desenvolvida por Slepian e Wolf [14] para o caso sem perdas e Wyner e Ziv [15] para o caso com perdas. Recentemente, a codificação distribuída tem sido o foco de diferentes tipos de codificadores de vídeo [16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26]. Na arquitetura DVC é necessário gerar uma informação lateral, *side information* (SI), para a

correta decodificação do sinal. O desempenho da codificação distribuída é altamente dependente da qualidade da informação lateral.

Em cenários realistas para comunicações de vídeo usando aparelhos com restrição de energia, é importante verificar os seguintes aspectos:

- Não é realmente necessário que o codificador de vídeo sempre opere no modo de complexidade reversa. Este modo pode ser ligado somente quando a bateria disponível diminui.
- Se a redução de complexidade é importante, manter a largura de banda também é um requisito básico. Logo, a redução de complexidade deve se adaptar levando em consideração um melhor desempenho em termos de RD.
- Como o vídeo transmitido por um aparelho móvel pode ser recebido e executado em tempo real por outro aparelho móvel, o decodificador deve pelo menos suportar um modo de decodificação de baixa qualidade, mas de baixo esforço computacional. Um processo de decodificação de alta qualidade pode ser realizado paralelamente por um servidor. A outra opção seria introduzir um transcodificador na rede que possa permitir baixa complexidade em ambos terminais de comunicação [27, 28].
- Enquanto a maioria dos trabalhos prévios em DVC [17, 24, 29] usam uma arquitetura que precisa de um canal de retorno, é mais prático considerar o caso onde a decodificação não necessariamente deve ser feita imediatamente após ser recebida. Logo, se a decodificação pode ser feita a qualquer momento o canal de retorno pode não estar disponível.

Além destas observações, como já mencionado, existem diferentes trabalhos recentes sobre DVC. Vários deles são baseados em codificadores DVC que usam quadros-chave periódicos, os quais são codificados utilizando somente técnicas de compressão *intra*-quadros ou simplesmente do tipo *intra* [17, 24, 30, 31, 32]. Logo, sem estimação de movimento, o codificador é significativamente menos complexo que o decodificador. Porém, isto limita o desempenho em termos de RD, devido à elevada taxa requerida para codificação apenas *intra*-quadro, e a baixa qualidade da SI. Uma forma óbvia de melhorar o desempenho seria permitir o uso de codificação *inter*-quadro nos quadros-chave. Porém, o desempenho ainda seria limitado, devido à qualidade

da SI gerada no decodificador. Uma variação deste tipo de métodos [18, 33] transmite uma informação auxiliar para melhorar a geração da SI. Porém, a taxa necessária para transmitir a informação auxiliar normalmente é proibitiva. Outro método [34] usa uma versão altamente comprimida dos quadros, limitando a estimação de movimento a vetores iguais a zero, para também melhorar a SI. Porém, no caso de haver um canal de retorno, este método não possui um mecanismo adequado de alocação de taxa.

Assim, neste trabalho apresentamos uma proposta para um codificador distribuído de vídeo que considera todos os aspectos mencionados anteriormente, além de oferecer uma nova arquitetura de DVC, novos métodos de geração de SI e mecanismos para omitir o canal de retorno.

1.3 ORGANIZAÇÃO

O trabalho está dividido em oito capítulos, incluindo o presente capítulo introdutório. O Capítulo 2 tem como objetivo apresentar conceitos básicos de codificação de fonte e canal, assim como introduzir os teoremas de Slepian-Wolf e Wyner-Ziv.

No Capítulo 3 serão abordados os princípios básicos da codificação de vídeo convencional, conhecida como codificação híbrida. Também, serão apresentadas as duas arquiteturas de DVC mais mencionadas na literatura; e serão listadas as vantagens e desvantagens da codificação distribuída em comparação à codificação convencional.

O Capítulo 4 representa a descrição do codificador e decodificador propostos. Neste capítulo, serão abordadas as formas de funcionamento do modo de codificação Wyner-Ziv desenvolvido, e a geração do código de canal utilizado.

Os mecanismos para a escolha dos parâmetros de codificação e estimação da correlação estatística dos sinais, para evitar o uso do canal de retorno, são descritos no Capítulo 5. Neste capítulo, uma caracterização RD para diferentes modelos de codificação distribuída é apresentada também.

No Capítulo 6 é detalhada a codificação de entropia dos *cosets*. A qual foi feita modificando um codificador existente no H.264/AVC.

Os resultados experimentais obtidos a partir da aplicação do trabalho desenvolvido ao padrão de codificação de vídeo H.264/AVC [6] são apresentados no Capítulo 7.

Finalmente, o Capítulo 8 é destinado às conclusões finais do trabalho, detalhando as contribuições feitas e as perspectivas para a continuidade da pesquisa.

2 CODIFICAÇÃO DE FONTE COM INFORMAÇÃO LATERAL

Neste capítulo definimos certos conceitos básicos de teoria da informação e apresentamos uma introdução à codificação de fonte e canal. Também trataremos brevemente dos resultados teóricos da codificação de fonte usando informação lateral no decodificador, tanto para codificar sem perdas como com perdas. O entendimento destes conceitos básicos é importante para a compreensão da aplicação da presente pesquisa.

2.1 CONCEITOS BÁSICOS DE TEORIA DA INFORMAÇÃO

A teoria da informação responde as duas questões fundamentais da compressão e codificação de dados: qual a máxima compressão possível, e qual a melhor taxa de transmissão nas comunicações [35]. Ela é elaborada sobre o tratamento de variáveis aleatórias, assim como o agrupamento delas em processos estocásticos. Uma variável aleatória é uma função mensurável, que atribui valores numéricos únicos a todos os possíveis resultados de um experimento aleatório sob determinadas condições. Os processos estocásticos permitem expressar matematicamente as relações entre suas variáveis aleatórias. Uma introdução a variáveis aleatórias e processos estocásticos pode ser encontrada em outros trabalhos [36].

Um conceito básico da teoria da informação é o conceito de fonte. O termo fonte é usado para indicar um processo que gera mensagens de informação sucessivas dentre um dado conjunto de mensagens possíveis. Uma fonte pode ser modelada como uma variável aleatória X que emite símbolos de um alfabeto χ e com função massa de probabilidade $p(x)$. Associada a cada fonte existe sua entropia H . A entropia é uma medida da incerteza de uma variável aleatória. Em termos de teoria de informação, a entropia indica a média da informação que uma fonte possui, em bits por símbolo.

Definição: Seja X uma variável aleatória discreta com alfabeto χ e função massa de probabilidade

$p(x)$, $x \in \chi$. Logo a entropia $H(X)$ é definida como:

$$H(X) = - \sum_{x \in \chi} p(x) \log_2 p(x). \quad (2.1)$$

A partir de (2.1) é possível definir a entropia conjunta e a entropia condicional de duas variáveis aleatórias. Seja Y uma variável aleatória discreta com alfabeto γ .

Definição: A entropia conjunta $H(X, Y)$ de duas variáveis aleatórias discretas (X, Y) com função massa de probabilidade conjunta $p(x, y)$ é definida como:

$$H(X, Y) = - \sum_{y \in \gamma} \sum_{x \in \chi} p(x, y) \log_2 p(x, y). \quad (2.2)$$

E a entropia condicional de Y dado X é expressada como:

$$H(Y|X) = - \sum_{x \in \chi} \sum_{y \in \gamma} p(x, y) \log_2 p(x|y). \quad (2.3)$$

Pode ser mostrado que a entropia conjunta de duas variáveis aleatórias é igual à entropia de uma delas mais a entropia condicional da outra:

$$H(X, Y) = H(X) + H(Y|X). \quad (2.4)$$

Outro conceito importante é a informação mútua. A informação mútua é uma medida da quantidade de informação que uma variável aleatória possui sobre outra variável aleatória. Em outras palavras, é a redução de incerteza de uma fonte X devido ao conhecimento de Y .

Definição: Considere duas variáveis aleatórias X e Y com função massa de probabilidade conjunta $p(x, y)$ e com funções massa de probabilidade marginais $p(x)$ e $p(y)$. A informação mútua $I(X; Y)$ é a entropia relativa entre a distribuição conjunta e o produto de $p(x)p(y)$:

$$I(X; Y) = \sum_{x \in \chi} \sum_{y \in \gamma} p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)}. \quad (2.5)$$

A equação (2.5) pode ser expressa em termos da entropia de uma das fontes e sua respectiva entropia condicional:

$$I(X; Y) = H(Y) - H(Y|X). \quad (2.6)$$

Como $H(X, Y) = H(X) + H(Y|X)$ temos que:

$$I(X; Y) = H(X) + H(Y) - H(X, Y). \quad (2.7)$$

2.2 CODIFICAÇÃO DE FONTE

No trabalho de Shannon [37] existem três resultados principais: o teorema de codificação de fontes, o teorema da taxa-distorção e o teorema da codificação de canal. O primeiro consiste na compressão sem perdas de uma fonte discreta; fontes contínuas não podem ser reproduzidas sem perdas. Neste teorema, foi provado que uma fonte discreta X pode ser reconstruída perfeitamente se, e somente se foi transmitida com uma taxa R_X não menor que a entropia $H(X)$. Logo, a condição necessária para compressão sem perdas é:

$$R_X \geq H(X). \quad (2.8)$$

A Figura 2.1 descreve um sistema básico de codificação e transmissão de uma fonte sem perda de informação.

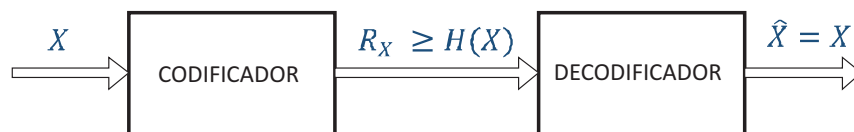


Figura 2.1: Compressão de fonte sem perdas. A fonte X deve ser transmitida com não menos que $H(X)$ bits.

A compressão com perdas é tratada no teorema de taxa-distorção. Se a reconstrução do sinal X é denotada por \hat{X} ou X' , então definimos a distorção D como:

$$D = d(X; \hat{X}), \quad (2.9)$$

onde d é uma medida de distorção. Uma medida comum é o erro quadrático medio, ou *mean square error* (MSE), definido como $d(X; \hat{X}) = E[(X - \hat{X})^2]$. Porém, existem outras medidas de distorção [38].

O teorema taxa-distorção prova que dada uma distorção D aceitável, existe uma taxa mínima R_X associada a esta distorção. A taxa é determinada pela função de taxa-distorção $R_X(D)$. A função $R_X(D)$ é convexa e retorna a mínima taxa para reconstruir X com uma distorção máxima de D . Em outros estudos pode-se encontrar mais detalhes sobre ambos os teoremas [35, 37, 38, 39]. O teorema da codificação de canal é abordado na subseção a seguir.

2.3 CODIFICAÇÃO DE CANAL

Um código de canal transforma uma entrada binária \vec{i} em um código \vec{c} . A taxa do código é definida como $R_C = \frac{k}{n} \leq 1$, o qual especifica que uma entrada de tamanho k gera um código de tamanho n . Os códigos corretores de erro ajudam a inferir a informação original \vec{i} mesmo se o código \vec{c} estiver corrompido [40, 41]. O teorema da codificação de canal [35] indica que para um valor real $\epsilon > 0$ e uma taxa de codificação $R_C < \mathbb{C}$, onde \mathbb{C} é a capacidade do canal, existe um código C tal que a probabilidade de erro depois da decodificação é menor que ϵ . A definição da capacidade do canal é descrita a seguir.

Definição: Se X é a entrada de um código discreto de canal sem memória que gera a saída C , a capacidade de canal é definida como:

$$\mathbb{C} = \max_{p(x)} I(X; C), \quad (2.10)$$

A capacidade do canal indica quanta informação pode ser transmitida por um canal com uma probabilidade de erro próxima a zero. Uma extensão dos teoremas de codificação de fonte e codificação de canal é o teorema de codificação de fonte-canal [35]. Ele estabelece que existe um codec fonte-canal que permite codificar uma fonte com entropia $H(X)$ de maneira confiável num determinado canal se e somente se $H(X) < \mathbb{C}$. No caso de codificação com perdas, sendo D a distorção permitida, é fácil verificar que se pode obter um código com taxa $R(D) < \mathbb{C}$, já que $H(X) = R(0) \geq R(D)$.

Um código de canal sistemático é formado pelo vetor original de entrada mais uma informação extra, chamada de paridade. A paridade ajuda a corrigir o vetor original no processo de decodificação caso um erro de transmissão tenha ocorrido. Logo, em um codificador sistemático $C = [X|P_d]$, onde P_d é a paridade e o operador “|” representa concatenação de vetores.

2.4 CODIFICAÇÃO DE FONTE COM INFORMAÇÃO LATERAL

A codificação de fonte com informação lateral é baseada nos resultados de Slepian-Wolf [14] para codificação sem perda de informação, e Wyner-Ziv [15] para a codificação com perdas.

2.4.1 Codificação Slepian-Wolf

A teoria de codificação de fonte de Shannon, expressa por (2.8), pode ser facilmente expandida para a codificação conjunta de duas fontes (X, Y) com entropia conjunta $H(X, Y)$, se for tratada como a codificação de uma única fonte Z com entropia $H(Z) = H(X, Y)$. Logo, para obter uma reconstrução sem perdas basta usar uma taxa $R_Z \geq H(Z)$. Porém, o problema também pode ser visto da seguinte forma: podemos transmitir a fonte X a uma taxa $R_X \geq H(X)$, e transmitir Y usando $R_Y \geq H(Y|X)$ bits, baseado no perfeito conhecimento prévio de X . Note que $R_X + R_Y \geq H(X, Y)$ por (2.4). O diagrama básico da codificação conjunta de duas fontes é mostrado na Figura 2.2.

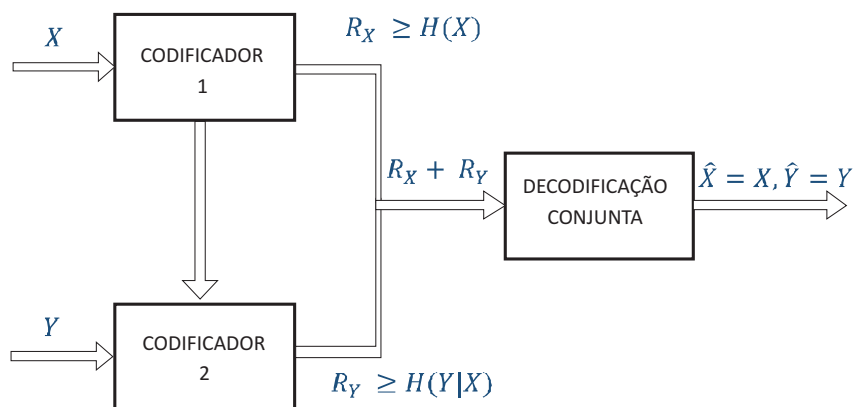


Figura 2.2: Codificação conjunta de duas fontes X e Y . Uma taxa não inferior a $H(X, Y)$ é suficiente para reconstrução sem perdas.

Em 1973, o resultado de Slepian e Wolf [14] conseguiu expandir a teoria de Shannon para a codificação separada de duas fontes correlatas. De acordo com o teorema de Slepian-Wolf, duas fontes podem ser codificadas separadamente e reconstruídas sem perdas se as estatísticas são conhecidas e $R_X \geq H(X|Y)$, $R_Y \geq H(Y|X)$ e $R_X + R_Y \geq H(X, Y)$. Este resultado é mostrado na Figura 2.3.

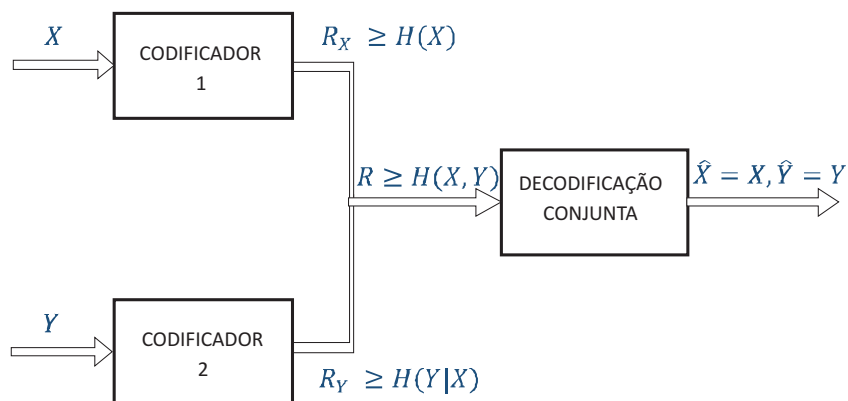


Figura 2.3: Codificação separada de duas fontes, com decodificação conjunta. $H(X, Y)$ continua sendo suficiente para reconstrução sem perdas.

Este teorema estende a região de codificação com possível reconstrução sem perdas para duas fontes correlatas. Na Figura 2.4 é mostrada a região de possível decodificação perfeita, ressaltando a região triangular superior (entre os pontos A e B) que representa a contribuição da codificação Slepian-Wolf.

A codificação de fonte com informação lateral sem perda de informação é um caso particular da codificação Slepian-Wolf. Pela teoria de Slepian-Wolf se a fonte Y existe somente no decodificador, ou foi transmitida a uma taxa não menor que $H(Y)$, é possível codificar a fonte correlata X a uma taxa não menor que $H(X|Y)$ para obter uma reconstrução perfeita $\hat{X} = X$. A fonte Y recebe o nome de informação lateral.

Se na codificação com informação lateral $R_X = H(X|Y)$, o sistema estaria operando exatamente no ponto A na Figura 2.4. O diagrama básico da codificação de fonte sem perdas com informação lateral é apresentada na Figura 2.5.

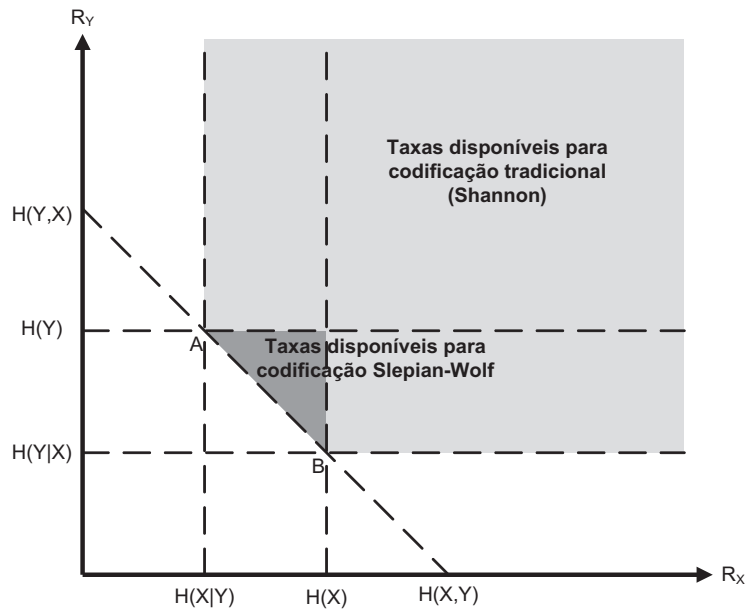


Figura 2.4: Região de possível reconstrução sem perda. A região triangular indica a região de codificação Slepian-Wolf.

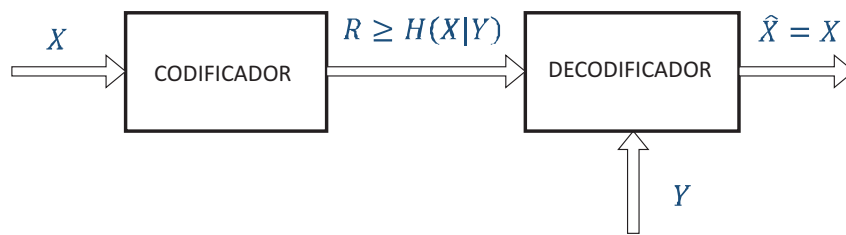


Figura 2.5: Codificação de fonte sem perdas com informação lateral.

2.4.2 Codificação Wyner-Ziv

Três anos depois da publicação de Slepian e Wolf, Wyner e Ziv [15] expandiram os resultados para a codificação com perdas com informação lateral. Assim como o teorema da codificação de fonte pode ser visto como um caso especial do teorema da taxa-distorção, se $D = 0$, a codificação de fonte com informação lateral de Slepian-Wolf pode ser vista como o caso de distorção zero para a codificação de Wyner-Ziv.

Seja $R_{WZ}(D)$ a função taxa-distorção de Wyner-Ziv. Se $R_X(D)$ é a função taxa-distorção para codificar e decodificar a fonte X com um valor esperado de distorção D , e $R_{X|Y}(D)$ é a função associada à codificação de X dada a informação perfeita de Y com distorção aceitável D . Wyner e Ziv provaram a veracidade das relações: $R_{WZ}(D) \geq R_{X|Y}(D)$ e $R_{WZ}(D) \leq R_X(D)$.

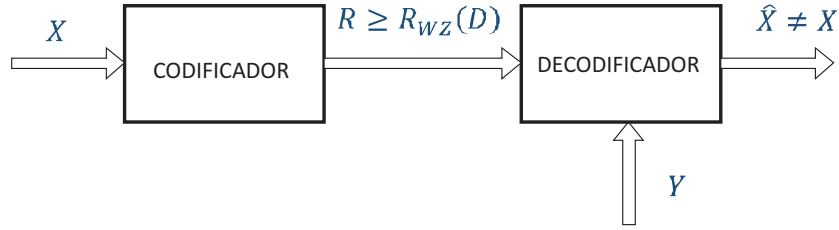


Figura 2.6: Codificação de fonte com perdas com informação lateral.

O cenário interessante é quando a informação lateral Y está presente somente no decodificador (Figura 2.6). No trabalho de Wyner-Ziv foi provado que, na codificação com informação lateral, existe uma situação onde $R_{WZ}(D) = R_{X|Y}(D)$, com o MSE como medida de distorção. O que significa que a codificação com informação lateral iguala o desempenho do codificador com conhecimento perfeito de Y . Para isso, X precisa ser uma fonte discreta ou contínua com distribuição de massa ou densidade, respectivamente, gaussiana e a informação lateral deve ser igual a:

$$Y = X + Z, \quad (2.11)$$

onde Z é gaussiana e independente de X . Mais tarde, foi provado que é necessário somente que Z seja gaussiana [42]. Z é conhecido como ruído de correlação Wyner-Ziv.

Quando estas condições não são cumpridas, existe uma diferença de taxa $\Delta R = R_{WZ}(D) - R_{X|Y}(D)$. Neste caso, a codificação com informação lateral sempre terá um desempenho pior que a codificação conjunta de ambas as fontes. ΔR é chamada de perda de codificação Wyner-Ziv. Foi provado que o limite superior desta perda é: $\Delta R < R_X(D) - R_{X|Y}(D)$ [43].

Neste capítulo foram apresentados conceitos básicos de teoria da informação e codificação de fonte com informação lateral. Estes conceitos são importantes para o entendimento do codificador proposto. No capítulo seguinte abordaremos os paradigmas de codificação híbrida e distribuída de vídeo.

3 PARADIGMAS DE CODIFICAÇÃO DE VÍDEO

DIGITAL

Este capítulo tem como foco a compressão de vídeo digital. Aqui será revisto o estado-da-arte em codificação convencional, conhecida como codificação híbrida, seguido de uma revisão de contribuições importantes na codificação distribuída de vídeo.

Uma seção discutindo as vantagens e desvantagens da codificação distribuída comparada a codificação híbrida está também presente neste capítulo. Assim, é possível entender os cenários onde a codificação distribuída é preferível à convencional, e vice-versa; como bem compreender melhor as contribuições do presente trabalho.

3.1 CODIFICAÇÃO HÍBRIDA DE VÍDEO

Um sinal digital de vídeo, como definido da Seção 1.1, é um sinal tridimensional. Este sinal é composto de uma seqüência discreta de imagens (quadros), que são representados por uma matriz discreta de valores de luminância e crominância. A maioria dos padrões de codificação de vídeo é baseada no mesmo modelo genérico de codec de vídeo [44]. Este codec incorpora funções de estimação e compensação de movimento, um estágio de transformação e quantização, e um codificador de entropia. Este modelo é normalmente conhecido como codificação híbrida, pois combina a codificação por transformada, normalmente a transformada discreta de cossenos (ou DCT), e a codificação por compensação de movimento, referida como modelo DPCM (do inglês *differential pulse code modulation*).

A redundância temporal de um sinal de vídeo digital é reduzida por codificação preditiva *inter*-quadro. Na codificação DPCM cada amostra é predita a partir de uma ou mais amostras previamente transmitidas. Usando o mesmo princípio, um codec de vídeo DPCM, gera um modelo do quadro a ser codificado baseado em quadros transmitidos (ou armazenados) anteriormente. O quadro a ser codificado é chamado de quadro fonte, enquanto que os quadros

usados para realizar a predição são conhecidos como quadros de referência. Inicialmente, o quadro fonte é dividido em blocos. A predição temporal, ou estimação de movimento, é feita a partir destes blocos de forma independente. O objetivo é encontrar, nos quadros de referência, a região mais similar ao bloco que está sendo codificado. Os quadros de referência normalmente são quadros adjacentes, por exemplo, o quadro exatamente anterior ao quadro fonte. O processo de gerar o quadro resultante, ao substituir cada bloco do quadro fonte pelo seu predito, é chamado de compensação de movimento. Este quadro resultante é subtraído do quadro fonte original para gerar o quadro residual. A entropia do quadro residual é menor que a do quadro fonte [45], podendo ser codificado mais eficientemente. Assim, somente são codificadas as informações necessárias para gerar o mesmo quadro compensado no decodificador (vetores de movimento) e o quadro residual. Os vetores de movimento indicam a posição da região escolhida no quadro de referência com relação ao bloco no quadro fonte.

No processo de transformação, o quadro residual é levado para outro domínio e representado por coeficientes. No domínio espacial, as amostras são altamente correlatas. O objetivo da transformação é reduzir esta correlação, gerando idealmente um pequeno número de coeficientes significativos para obter uma representação visual com uma distorção aceitável, e um número alto de coeficientes que podem ser eliminados [45]. Finalmente, os coeficientes restantes são codificados mediante um codificador de entropia (ou codificador sem perdas).

Nas Figuras 3.1 e 3.2 são mostrados um codificador e um decodificador genéricos DPCM/DCT, respectivamente. Se $X(m, n, t)$ denota um sinal de vídeo tridimensional, onde (m, n) são as dimensões espaciais e t representa a dimensão temporal, podemos definir um quadro como simplesmente $X(t)$. Assim, no codificador (Figura 3.1) o quadro fonte, $X(t)$, é processado para gerar um *bit-stream* (seqüência de bits codificados). No decodificador (Figura 3.2) o *bit-stream* recebido é usado para obter um quadro reconstruído $X'(t)$ (ou $\hat{X}(t)$). Note que normalmente o processo possui perdas devidas principalmente à quantização dos coeficientes transformados, logo $X'\{t\} \neq X(t)$.

No codificador existem dois fluxos principais de dados: o fluxo de codificação e o fluxo de reconstrução, conhecido como decodificador local. O fluxo de codificação DPCM/DCT pode ser descrito da seguinte forma [44]:

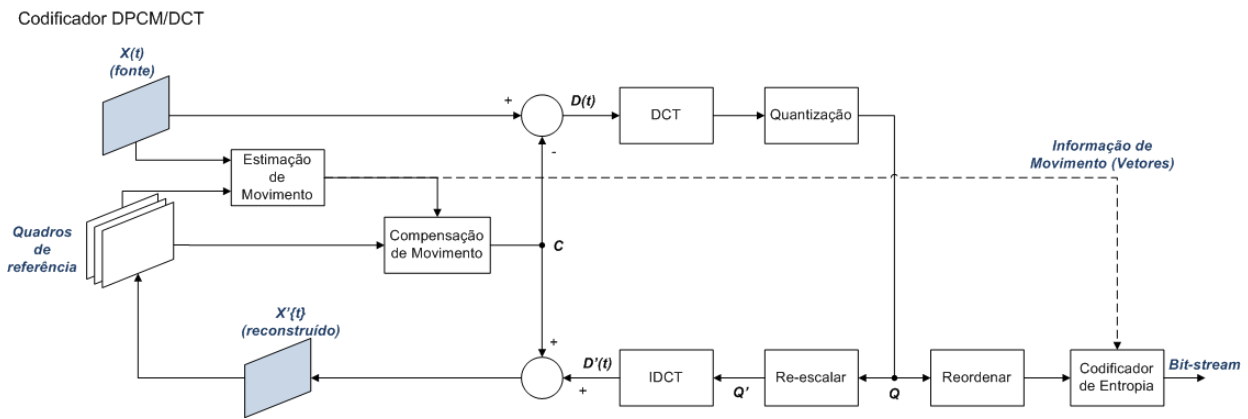


Figura 3.1: Esquema básico de um codificador híbrido

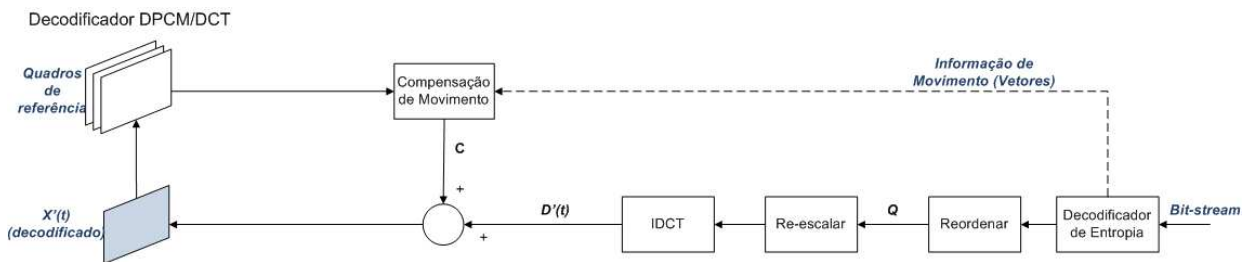


Figura 3.2: Esquema básico de um decodificador híbrido

- O quadro fonte, $X(t)$, a ser codificado, é inicialmente dividido em blocos de $N \times N$ pixels. Normalmente $N = 16$. Os blocos são denominados macroblocos.
- Depois, os macroblocos de $X(t)$ são comparados com os quadros de referência, por exemplo com $X'(t - 1)$. Assim, a função de estimação de movimento encontra a região de $N \times N$ pixels em $X'(t - 1)$ mais similar ao macrobloco que esta sendo codificado. A região escolhida em $X'(t - 1)$ é chamada de predita.
- A posição no quadro $X'(t - 1)$ do predito de cada macrobloco é guardada em forma de vetores de movimento (em inglês *motion vectors*, ou MV).
- Usando os MVs de cada macrobloco é gerado um quadro compensado, que é formado ao substituir cada macrobloco em $X(t)$ pelo seu predito. Este processo é chamado de compensação de movimento.
- O quadro residual $D(t)$ é formado subtraindo o quadro compensado do quadro fonte original $X(t)$.

- $D(t)$ é transformado usando a DCT em blocos $M \times M$. Usualmente, $M = 8$ ou $M = 4$.
- Cada bloco transformado é quantizado, usando um passo de quantização QP , e reordenado. O reordenamento é feito para melhorar a eficiência da codificação de entropia.
- Finalmente, o sinal reordenado, junto com os MVs e informações adicionais de cabeçalho, são comprimidos sem perdas, mediante um codificador de entropia.

O fluxo do decodificador local é o seguinte [44]:

- Cada bloco quantizado Q passa por um processo de re-escala. A re-escala tenta inverter a quantização. Porém, não é possível recuperar a perda de informação gerada pela quantização, logo $Q' \neq Q$.
- Os coeficientes passam pelo processo de transformação inversa (IDCT).
- O quadro compensado é adicionado ao resíduo $D'(t)$ para produzir o frame reconstruído $X'(t)$. Este quadro será armazenado para ser usado como quadro de referência.

Note que a estimação de movimento é feita utilizando dados reconstruídos, pois o decodificador não possui os dados originais. Assim, evitam-se erros de escorregamento (no inglês *drifting errors*). Existem trabalhos recentes que usam dados originais para a estimação de movimento e dados reconstruídos para a compensação de movimento para permitir a paralelização do processo [46, 47].

O fluxo de dados no decodificador remoto é similar ao fluxo do decodificador local, com a diferença que no decodificador remoto é necessário utilizar um decodificador de entropia para obter o resíduo quantizado e os vetores de movimento a partir do *bit-stream*.

Quando é utilizado somente o quadro anterior para codificar um quadro fonte, a codificação é chamada de preditiva e o quadro codificado é chamado de tipo P . Se o quadro anterior e posterior são utilizados como referência, então a codificação é bi-preditiva e o quadro é chamado de tipo B . Os quadros B normalmente não são usados como referência para outros quadros, porém no padrão H,264/AVC é possível ter quadros B de referência. Isto é exemplificado na Figura 3.3,

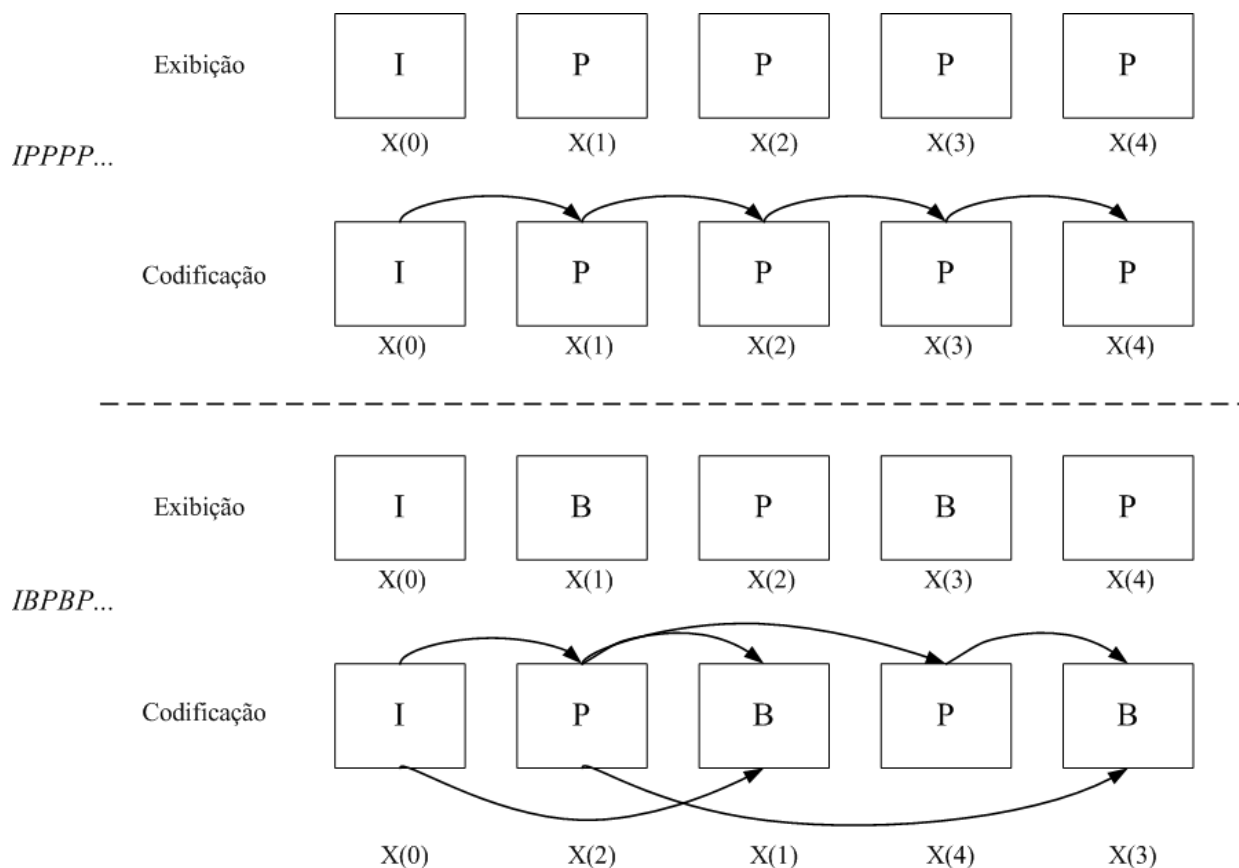


Figura 3.3: Exemplos de codificação preditiva e bi-preditiva. É mostrada a ordem de exibição e codificação, as setas indicam quais quadros são usados como referência da codificação do quadro fonte.

onde os quadros *I* representam os quadros codificados sem informação temporal (codificação *intra*-quadro), que basicamente seria transformação e quantização.

O modelo híbrido descrito é compatível com os padrões H.261 [48], H.263 [49], MPEG-1 [50], MPEG-2 [51], MPEG-4 *Visual* [52] e H.264/AVC [53]. O padrão MPEG-2 é um dos mais usados em produtos comerciais na atualidade, como por exemplo, em vídeos no formato de *digital versatile disc* (DVD) e alguns padrões de TV digital. O padrão MPEG-4 *Visual* além de comprimir seqüências de vídeo natural com melhor desempenho que o seu antecessor MPEG-2, possui uma série de ferramentas, como codificação de objetos com forma arbitrária e codificação de vídeo sintético, como por exemplo, animações 3-D. Porém, o padrão H.264/AVC, também conhecido como MPEG-4 Part 10, representa o estado-da-arte para codificação de seqüências naturais de quadros retangulares. O H.264/AVC consegue, para uma qualidade determinada, menor taxa que os outros padrões mencionados [44]. Uma boa introdução aos algoritmos de codificação do

H.264/AVC pode ser encontrada em trabalhos anteriores [6, 44].

O H.264/AVC obtém um melhor desempenho que seus antecessores principalmente devido à: possibilidade de dividir o macrobloco em blocos retangulares de diferentes tamanhos; possibilidade de utilizar até 15 quadros de referência não necessariamente adjacentes (tanto anteriores como posteriores); aplicação de codificação preditiva nos próprios vetores de movimento e informações adicionais de cabeçalho; e ao uso de codificadores de entropia com contexto. O padrão também possui uma série de modos de predição *intra*-quadro, que fazem uso da correlação espacial no domínio dos *pixels* para comprimir o quadro sem utilizar informação temporal, obtendo um melhor desempenho, em termos de taxa-distorção, do que somente a transformação e quantização. O codificador H.264 operando em modo *intra* é capaz de superar inclusive padrões considerados o estado-da-arte em codificação de imagens estáticas, como o JPEG-2000 [54].

A qualidade de um sinal de vídeo reconstruído é na verdade subjetiva. Porém, o desempenho na codificação de vídeo é medida de forma objetiva na literatura pela taxa de bits versus o MSE (*mean square error*), ou mais comumente versus a PSNR (*peak signal to noise ratio*). A formulação da PSNR em decibéis é dada por:

$$PSNR = 10 \log_{10} \frac{(2^n - 1)^2}{MSE}, MSE = \frac{1}{N} \sum_{t=0}^N (X(t) - \hat{X}(t))^2, \quad (3.1)$$

onde n representa o número de bits utilizados para definir um *pixel*. Se $n = 8$, por exemplo, o valor de pico do numerador é 255.

Logo, o padrão H.264/AVC consegue para um mesmo valor de PSNR comprimir a menor taxa, que os outros padrões mencionados. O modo de codificação com complexidade reversa apresentado neste trabalho foi implementado no padrão H.264/AVC.

3.2 CODIFICAÇÃO DISTRIBUÍDA DE VÍDEO

A codificação distribuída de vídeo, DVC, é baseada na codificação de fonte com informação lateral presente somente no decodificador. No paradigma de codificação híbrida, o codificador requer um maior esforço computacional que o decodificador principalmente devido à função de

estimação de movimento [6]. Se estágios de otimização de RD são usados, então a complexidade do codificador aumenta ainda mais [6]. Logo, a idéia do paradigma DVC é transferir parte da complexidade para o decodificador, gerando um codificador menos complexo. Basicamente, tenta-se evitar ou reduzir ao máximo as operações ligadas à função de estimação de movimento, sendo esta operação transferida ao decodificador.

Embora a codificação distribuída tenha seus fundamentos em estudos de teoria da informação realizados nos anos setenta [14, 15], as pesquisas para a implementação de codecs de vídeo práticos são recentes. A primeira proposta de codificação Wyner-Ziv foi implementada em 1999 [16], onde foi considerado o caso assimétrico de codificação de fonte com informação lateral para fontes binárias e gaussianas. Trabalhos posteriores consideraram a codificação simétrica onde as fontes são codificadas com a mesma taxa [55, 56, 57, 58]. Anos depois as primeiras arquiteturas de DVC foram propostas: (i) modelo de Stanford [17, 24, 59], e (ii) o PRISM [18, 60, 61]. A arquitetura de Stanford tem sido explorada extensivamente na literatura, gerando diferentes trabalhos derivados da proposta inicial que apresentam contínuas melhorias de desempenho. A arquitetura PRISM, feita em Berkeley, não foi explorada tanto quanto a de Stanford por requerer uma maior dificuldade de implementação. Porém, possui uma proposta diferente, sendo referência para qualquer trabalho em DVC.

3.2.1 Arquitetura Stanford

O modelo de Stanford foi proposto inicialmente para trabalhar no domínio espacial (pixels) [24, 29, 59, 62], depois expandido para o domínio da transformada [17]. O codificador Slepian-Wolf foi inicialmente implementado usando *turbo codes* como código de canal [17, 24, 59, 29, 62]. Posteriormente, este modelo foi atualizado para códigos LDPC (*low-density parity-check codes*) [63, 64].

O diagrama de blocos básico do codec DVC é mostrado na Figura 3.4. Inicialmente o sinal $X(m, n, t)$ é dividido em dois grupos de quadros. Um grupo de quadros é denominado quadros-chave. Os quadros chave são normalmente os quadros ímpares $C(t) = X(2i + 1), i = 0, 1, 2, \dots$, e são codificados de forma independente sem considerar a informação temporal. Ou seja, são codificados somente usando técnicas de codificação *intra*. Os quadros restantes, $W(t) =$

$X(2i), i = 1, 2, \dots$, são codificados mediante técnicas de codificação Wyner-Ziv.

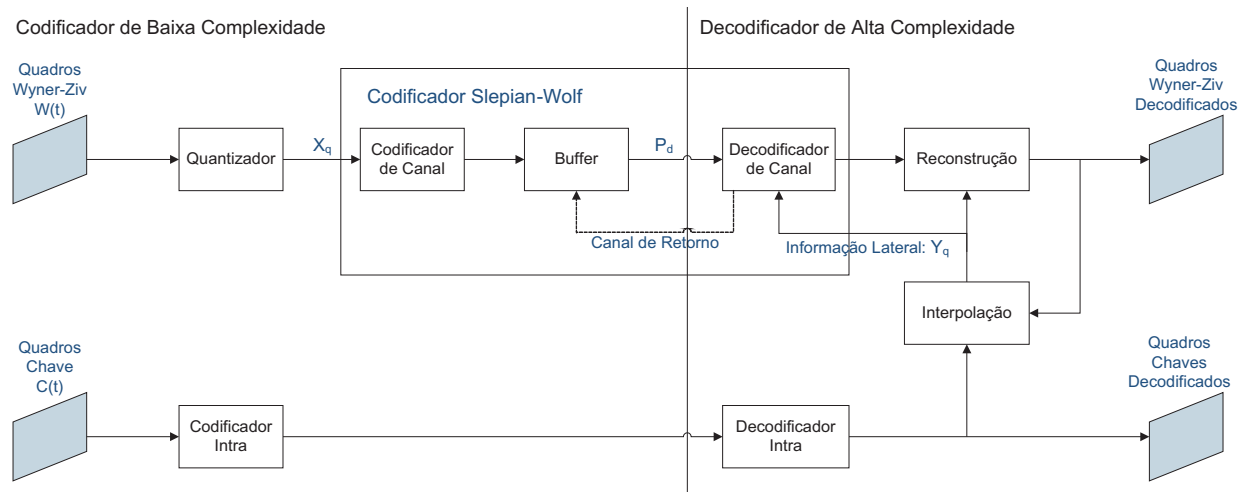


Figura 3.4: Diagrama de blocos de codec DVC de Stanford.

A codificação Wyner-Ziv é implementada por um processo de quantização e um codificador Slepian-Wolf. Como mencionado na Seção 2.4, o codificador Slepian-Wolf é um codificador que permite a reconstrução perfeita, enquanto a função de quantização é a responsável pela distorção introduzida da codificação Wyner-Ziv. O processo de codificação funciona da seguinte forma:

- $W(t)$ é quantizado, usando um passo de quantização $Q_{p,wz}$.
- O quadro quantizado é reordenado em planos de bits. O sinal quantizado e reordenado será denominado por X_q .
- X_q é alimentado a um codificador de canal sistemático, do tipo corretor de erro (por exemplo *turbo codes*) [40], gerando o sinal $[X_q|P_d]$, onde P_d representa a informação de paridade adicionada pelo codificador de canal.
- Somente os bits de paridade P_d são enviados, a medida que sejam requeridos pelo decodificador mediante o canal de retorno. (No caso da implementação usando LDPC a informação enviada é a síndrome).

O canal de retorno é responsável por permitir a implementação do codificador Slepian-Wolf. É por meio desse canal que o decodificador requer os bits de paridade até conseguir a reconstrução perfeita do sinal quantizado $\hat{X}_q = X_q$. A decodificação segue os seguintes passos:

- Os quadros-chave previamente decodificados são utilizados para gerar a informação lateral Y_q . Varias técnicas na literatura têm sido apresentadas para a geração de informação lateral em um codificador DVC com esta arquitetura [30, 31, 32, 65, 66]. Basicamente, a idéia é utilizar o quadro chave anterior $C(t) = X(i-1)$, ou o anterior $C(t) = X(i-1)$ e posterior $C(t+1) = X(i+1)$, ao quadro Wyner-Ziv $W(t) = X(i)$, para realizar uma interpolação temporal, gerando o quadro interpolado $Y(t)$.
- Y é quantizado e reordenado para gerar Y_q , que representa a informação lateral presente somente no decodificador.
- Y_q é concatenado aos bits de paridade recebidos, obtendo o código $[Y_q|P_d]$. Note que para o decodificador de canal Y_q é uma versão ruidosa de X_q .
- O decodificador de canal vai requerer a quantidade de bits de paridade necessários para reconstruir X_q .
- Finalmente, X_q e o quadro interpolado Y são usados para reconstruir o quadro $\hat{W}(t) \neq W(t)$. A reconstrução é feita por máxima verossimilhança $\hat{W}(t) = E[W(t)|X_q, Y]$.

A complexidade do codificador é reduzida, pois nenhuma técnica de codificação *inter-quadros*, como a estimação de movimento, é utilizada. Já o decodificador é mais complexo, pois o desempenho do codec está associado diretamente à qualidade da informação lateral gerada. A geração do quadro interpolado Y requer algoritmos tão ou mais complexos que a estimação de movimento usada na codificação convencional.

Esta arquitetura consegue reduzir drasticamente o esforço computacional no codificador. Uma versão de menor qualidade temporal pode ser obtida no decodificador com baixa complexidade, se são decodificados somente os quadros-chave. Por outro lado, as melhorias se restringem a melhorar a informação lateral ou a codificação de canal. Existe um trabalho onde uma informação auxiliar (*hash*) é enviada para o decodificador, para gerar um melhor quadro interpolado Y [33]. Porém, a taxa para enviar o *hash* é normalmente proibitiva. A necessidade de se ter um canal de retorno restringe o uso deste codificador a situações onde a decodificação deve ser feita imediatamente após o recebimento do sinal. Além disso, não é possível recuperar uma versão de baixa complexidade de decodificação dos quadros Wyner-Ziv.

O desempenho desta arquitetura, usando a codificação *intra* do padrão H.263+, é superior a codificar todos os quadros de forma *intra* (codificação *III...*), mas inferior a usar codificação *IPIP...* [25]. No caso de usar H.264/AVC, para codificar os quadros-chave, a arquitetura de Stanford apresenta um desempenho superior a codificação *IIII...* do H.264, porém inferior a codificação *IPIP...* com vetores de movimento zero [67]. Note que ter $MVs = 0$ é um subconjunto da codificação *IPIP* regular. Em sequências com muito movimento, o desempenho do codificador DVC em H.264/AVC pode ser pior que quadros *IIII...* codificados com H.264.

3.2.2 Arquitetura PRISM

A outra arquitetura DVC que foi proposta no início das pesquisas é a arquitetura PRISM (*Power-efficient, robust, high-compression syndrome-based multimedia coding*) [16, 18, 60, 61]. Esta proposta é significativamente diferente da arquitetura de Standford. O diagrama de blocos é mostrada na Figura 3.5.

Na arquitetura PRISM, a codificação Wyner-Ziv não é feita em nível de quadros, mas sim feita em nível de macroblocos. O quadro $X(t)$ é dividido em macroblocos e transformados ao domínio da DCT. Com base na estimação do ruído de correlação Wyner-Ziv, os macroblocos são classificados. Esta classificação determina a expectativa da qualidade do bloco decodificado. Dependendo da classificação, o codificador pode escolher por codificar o macrobloco de forma *intra*. Os passos na codificação são os seguintes:

- Os coeficientes transformados dos blocos são divididos em alta e baixa frequência. A classe do macrobloco determina como a divisão é feita. Logo, o número de coeficientes de alta e baixa frequência é variável.
- Os coeficientes de alta frequência são quantizados e codificados utilizando métodos tradicionais de codificação de entropia. O passo de quantização depende da classificação do macrobloco.
- Os coeficientes de baixa frequência passam por dois processos de quantização paralelos. O primeiro é idêntico a quantização dos coeficientes de alta frequência. O segundo é um processo de refinamento da quantização, a qual é dependente da distorção permitida (ou

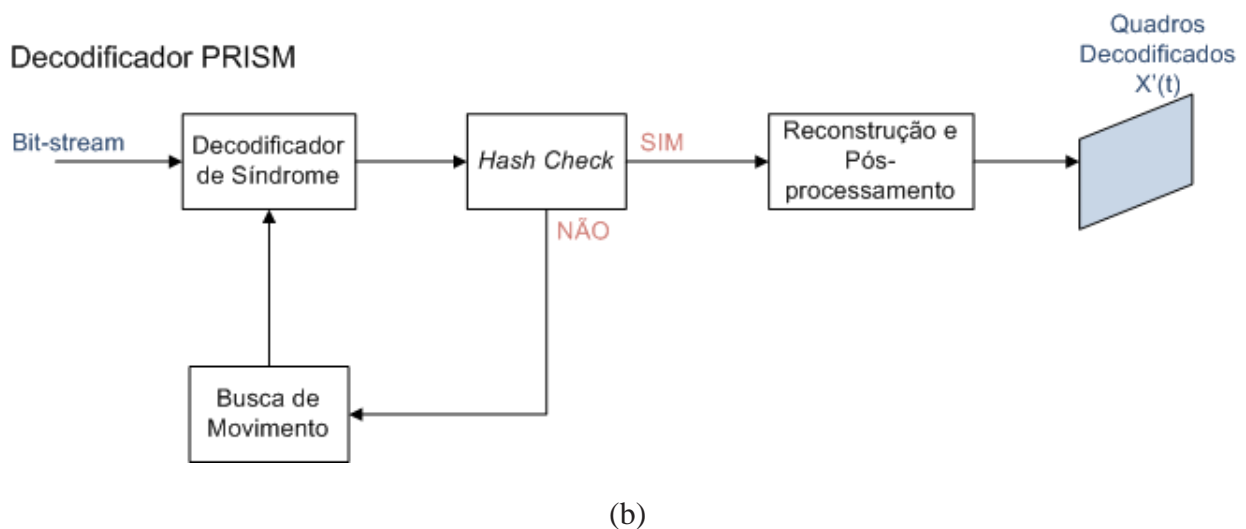
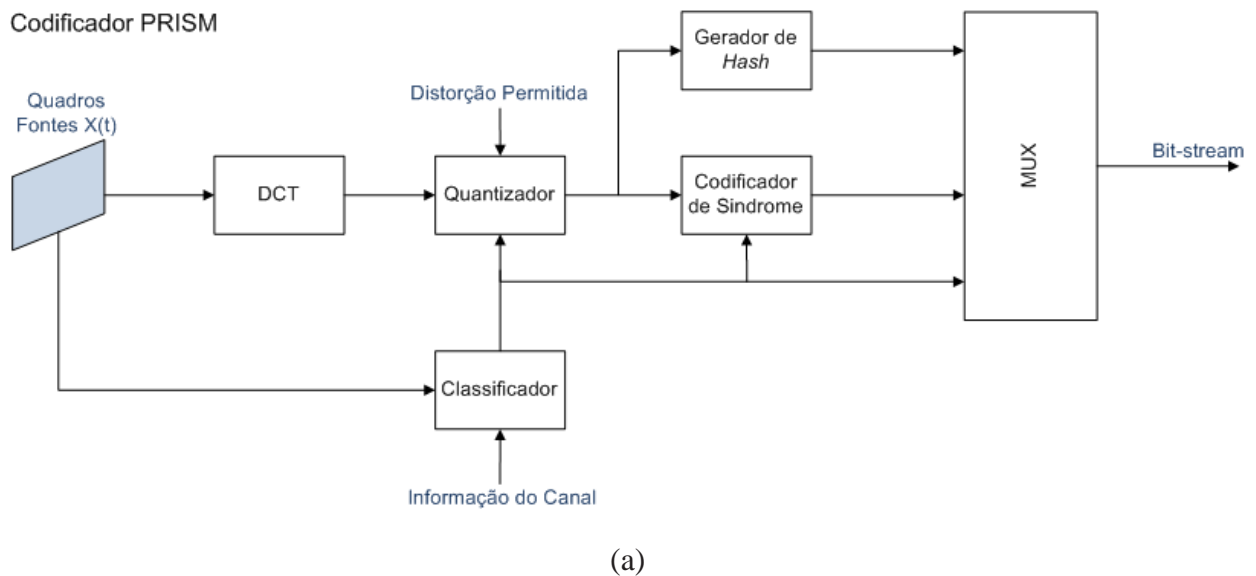


Figura 3.5: Diagrama de Blocos do PRISM. (a) codificador, (b) decodificador

seja, do passo de quantização QP que seria usado na codificação convencional) e não da classe do macrobloco.

- A diferença entre ambos os sinais, dos coeficientes quantizados de baixa frequência, é enviada ao codificador de entropia. O resultado do primeiro processo de quantização é usado para gerar um código síndrome.

A codificação por síndrome é um tipo de codificação de canal. Em lugar de enviar a paridade, é enviada a síndrome do código. Se $S = XG$, onde G é a matriz de paridade de um código de blocos linear, então S é a síndrome do código. Maiores detalhes podem ser encontrados em referências tradicionais [40]. A funcionalidade da síndrome é similar a da paridade, ambos são

usados para corrigir erros de transmissão. A síndrome pode ser considerada um *coset*. Se a fonte X possui um alfabeto χ , ao dividir o alfabeto em regiões, cada região seria um *coset*. Logo a síndrome indica a qual *coset* a informação pertence.

Além da informação de *coset*, o codificador PRISM gera um *hash* para cada macrobloco usando códigos CRC (*cyclic redundancy check*). Logo, o *bit-stream* é formado pela síndrome, coeficientes quantizados de alta frequência, a diferença entre a quantização escalar e o refinamento dos coeficientes de baixa frequência, e o código CRC de cada macrobloco.

No decodificador o fluxo de dados é feito da seguinte maneira:

- Uma série de blocos candidatos a informação lateral no quadro decodificado anteriormente $X'(t - 1)$ são selecionados mediante uma procura similar à estimação de movimento, usando os bits residuais do refinamento da quantização para realizar a procura dos blocos candidatos.
- No primeiro candidato são substituídos os coeficientes de alta frequência pelos existentes no *bit-stream*, gerando assim a informação lateral, ou versão ruidosa da informação original
- A informação lateral é corrigida com o uso da síndrome, em seguida uma verificação é feita usando os bits CRC. Se a verificação é incorreta é utilizado um novo candidato e o processo é repetido até obter uma verificação válida entre todos os candidatos. Se todas as verificações forem incorretas, então o candidato que gerou a verificação mais próxima à correta é selecionado.

A redução de complexidade é alcançada porque no codificador não é realizada nenhuma estimação de movimento. Já no decodificador, uma procura de movimento é feita quando são gerados os blocos candidatos a informação lateral. Porém, não há como obter uma seqüência decodificada, mesmo de baixa qualidade, com baixa complexidade de decodificação, pois a codificação Wyner-Ziv é feita em cada macrobloco de todos os quadros.

Note que nesta arquitetura a codificação com informação lateral no decodificador já é implementada com perdas. No caso da arquitetura de Stanford, a perda é devido ao processo de quantização. Aqui, além da quantização, a síndrome é gerada de acordo a classificação dos

blocos. Não existe canal de retorno, logo a reconstrução perfeita está associada a uma estimação perfeita do ruído para classificar corretamente cada macrobloco.

Sem o canal de retorno este codec pode ser usado em diferentes cenários onde a codificação em baixa complexidade é necessária, ao contrário do restrito uso da arquitetura de Stanford. Contudo, nos trabalhos publicados até a presente data, os autores não definem um mecanismo eficiente de estimação de ruído nem um modelo prático de classificação de macroblocos. Os autores assumem que possuem tanto no codificador como no decodificador a informação perfeita do ruído inserido na geração da informação lateral. Com esta estimação perfeita foram feitos testes que mostram resultados de desempenho similares à codificação *IPPP*.. no padrão H.263+ para uma quantidade limitada de quadros [61]. Estes testes mostram o potencial desta arquitetura, mas não é realista assumir que o codec possui a informação perfeita do ruído. Isto requereria que o codificador realize estimação de movimento para saber antecipadamente qual seria o candidato a informação lateral escolhido no decodificador. Realizando estimação de movimento, o codificador desta arquitetura seria tão complexo quanto um codificador híbrido convencional. Logo, tal como proposta, a arquitetura PRISM não pode ser implementada na indústria. A grande contribuição deste trabalho foi mostrar a possibilidade de criar um codificador DVC sem canal de retorno e verificar que esta arquitetura é mais robusta quando o canal de transmissão é ruidoso. Inclusive, o foco das simulações feitas com a arquitetura PRISM é verificar a resistência do codificador a erros de transmissão. Os autores mostraram que o PRISM é muito mais robusto contra erros que a codificação tradicional híbrida [61]. Existem outros trabalhos onde a codificação Wyner-Ziv de vídeo é usada especificamente para gerar um *bit-stream* resistente a erros, no lugar de procurar baixa complexidade de codificação [68, 69, 70, 71].

3.3 VANTAGENS E DESVANTAGENS DA CODIFICAÇÃO DE VÍDEO DISTRIBUÍDA

Ainda não existe um padrão técnico de codificação de vídeo que use o paradigma de codificação distribuída, porém existe um alto interesse deste novo método mostrado por diferentes grupos de pesquisa. A codificação distribuída possui certas funcionalidades que não são possíveis

de alcançar na codificação híbrida:

- Possibilidade de alocar a complexidade de forma adaptável entre o codificador e decodificador. Como a codificação DPCM/DCT realiza a estimação de movimento totalmente no codificador, este vai ser sempre mais complexo que o decodificador. Um codificador que consiga misturar ambos os paradigmas pode potencializar o balanceamento do esforço computacional de acordo com a necessidade.
- Os sinais codificados (*bit-streams*) dos codificadores Wyner-Ziv possuem uma proteção intrínseca contra erros de transmissão. Como mostrado em [61], os códigos corretores de erro que são usados para explorar a correlação dos sinais do decodificador também podem ajudar a corrigir erros gerados na transmissão.

Estas vantagens estão associadas a importantes desvantagens:

- O desempenho em termos de taxa-distorção dos codificadores DVC é pior que os dos codificadores híbridos. Como a codificação Wyner-Ziv pode obter uma taxa igual a codificação convencional somente quando trata-se de fontes gaussianas, como foi afirmado em outros trabalhos [25], o DVC não deve atingir o desempenho de codificadores convencionais. No entanto, arquiteturas como PRISM tem mostrado que existe uma possibilidade de realmente atingir um desempenho muito similar.
- A complexidade transferida ao decodificador não é proporcional à complexidade diminuída no codificador. O esforço computacional do codificador e decodificador juntos em uma arquitetura DVC é normalmente superior ao de uma arquitetura convencional.
- Devido ao uso de códigos corretores de erro, mesmo usando um canal de retorno, é muito mais difícil controlar a taxa e/ou distorção de um codificador DVC, já que isso depende de uma boa estimação das estatísticas do ruído de correlação Wyner-Ziv.
- O canal de retorno é inviável em termos práticos.

Logo, é importante notar que a codificação DVC dificilmente deve substituir a codificação convencional. O paradigma DVC deve ser usado como uma ferramenta adicional, que pode ser

ativada ou desativada num codificador, dependendo da situação, ou em aplicações específicas. No próximo capítulo será proposta uma nova arquitetura de codificação distribuída, a qual pode trabalhar como um modo opcional de codificação dentro de um codec convencional.

4 CODIFICADOR DISTRIBUÍDO COM RESOLUÇÃO ESPACIAL MISTA

Um dos meios utilizados pelos codecs de vídeo distribuídos para reduzir o esforço computacional na codificação é eliminar a etapa de estimação de movimento, transferindo esta função para o decodificador. Uma outra alternativa é usar codificação baseada em redução da resolução espacial. Na arquitetura que é apresentada neste capítulo, os quadros-chave são os quadros usados como referência no processo de estimação de movimento. Estes quadros podem ser codificados como quadros *I*, *P*, ou *B* (ver Seção 3.1) com resolução espacial completa. Os quadros intermediários são codificados usando duas camadas: a camada base codificada com resolução espacial reduzida e a camada de realce codificada mediante técnicas Wyner-Ziv. A codificação distribuída de vídeo em camadas já foi usada em outros trabalhos [20, 21], baseados em resultados teóricos de refinamento sucessivo de fontes gaussianas para codificação Wyner-Ziv [72], obtendo um codec com adaptabilidade de SNR com desempenho similar ao MPEG-4/H.26L FGS [44]. Aqui, o objetivo é obter uma adaptabilidade de complexidade de codificação, mediante resolução espacial mista.

O modo de codificação Wyner-Ziv mediante resolução espacial mista pode ser implementado em qualquer padrão de codificação de vídeo [2, 6]. A redução da complexidade de codificação é atingida pois os quadros não usados como referência são codificados a baixa resolução espacial. Como a redução da complexidade é alcançada independentemente da implementação do codec convencional, qualquer técnica de estimação de movimento rápida [73, 74] pode ser utilizada. No decodificador, se somente os quadros-chave são decodificados, podemos obter uma seqüência de baixa complexidade de decodificação com perda de quadros, de forma similar à arquitetura de Stanford (ver Seção 3.2.1). Porém, podemos também obter uma versão de baixa resolução dos quadros intermediários em tempo real simplesmente decodificando a camada base e usando filtros clássicos de interpolação.

Para decodificar a camada de realce é proposta uma decodificação iterativa. Outros estudos usam decodificação iterativa em arquiteturas similares à de Stanford [30, 31, 32]. Alguns geram

a informação lateral a medida que os planos de bits são decodificados [30], enquanto outros usam um processo similar à compensação de movimento [31]. O método aqui proposto é significativamente diferente.

Usar codificação em baixa resolução para diminuir a complexidade de codificação já foi o foco de outros estudos em DVC [30] e inclusive para codificadores tradicionais [75]. Porém, usar resolução mista na codificação distribuída evita degradar os frames chave possibilitando um melhor desempenho RD.

Este capítulo apresenta a arquitetura proposta de codificação e decodificação do modo Wyner-Ziv com resolução espacial mista ou MR-DVC (do inglês *mixed resolution-DVC*), assim como o código de canal utilizado para realizar a codificação distribuída. Note que a arquitetura não utiliza canal de retorno. No Capítulo 5 é detalhado como foi possível eliminar o canal de retorno.

4.1 RESOLUÇÃO MISTA

O *framework* proposto é baseado nos quadros que não são usados como referência, que chamaremos de quadros NRWZ (*non-reference Wyner-Ziv*). A redução de complexidade é aplicada somente a estes quadros. Já que os quadros de referência serão codificados como quadros *I*, *P* ou *B*, de um codificador híbrido convencional, não ocorrerão erros de *drifting*. Logo, o decodificador pode reproduzir uma seqüência de vídeo de menor qualidade com baixo esforço computacional. A decodificação da camada de realce aumenta a qualidade do sinal e a complexidade de decodificação. O nosso *framework* não limita a quantidade de quadros NRWZ. Idealmente, a quantidade de quadros NRWZ pode variar de acordo com os requisitos de esforço computacional e distorção permitidos. Na Figura 4.1 três cenários de codificação são mostrados. No primeiro, Figura 4.1 (a), os quadros do tipo *B* de um codificador convencional são convertidos em quadros NRWZ. A Figura 4.1 (b) mostra um caso similar com maior redução de complexidade, já que mais quadros NRWZ foram inseridos. Na Figura 4.1 (c) mostra-se o caso de baixo atraso onde os quadros NRWZ são utilizados como quadros *P*.

Note que o *framework* proposto é diferente da adaptabilidade espacial do SVC, ou H.264/SVC, [76]. O codec SVC possui um *bit-stream* adaptável composto de vários *sub-streams*, cada

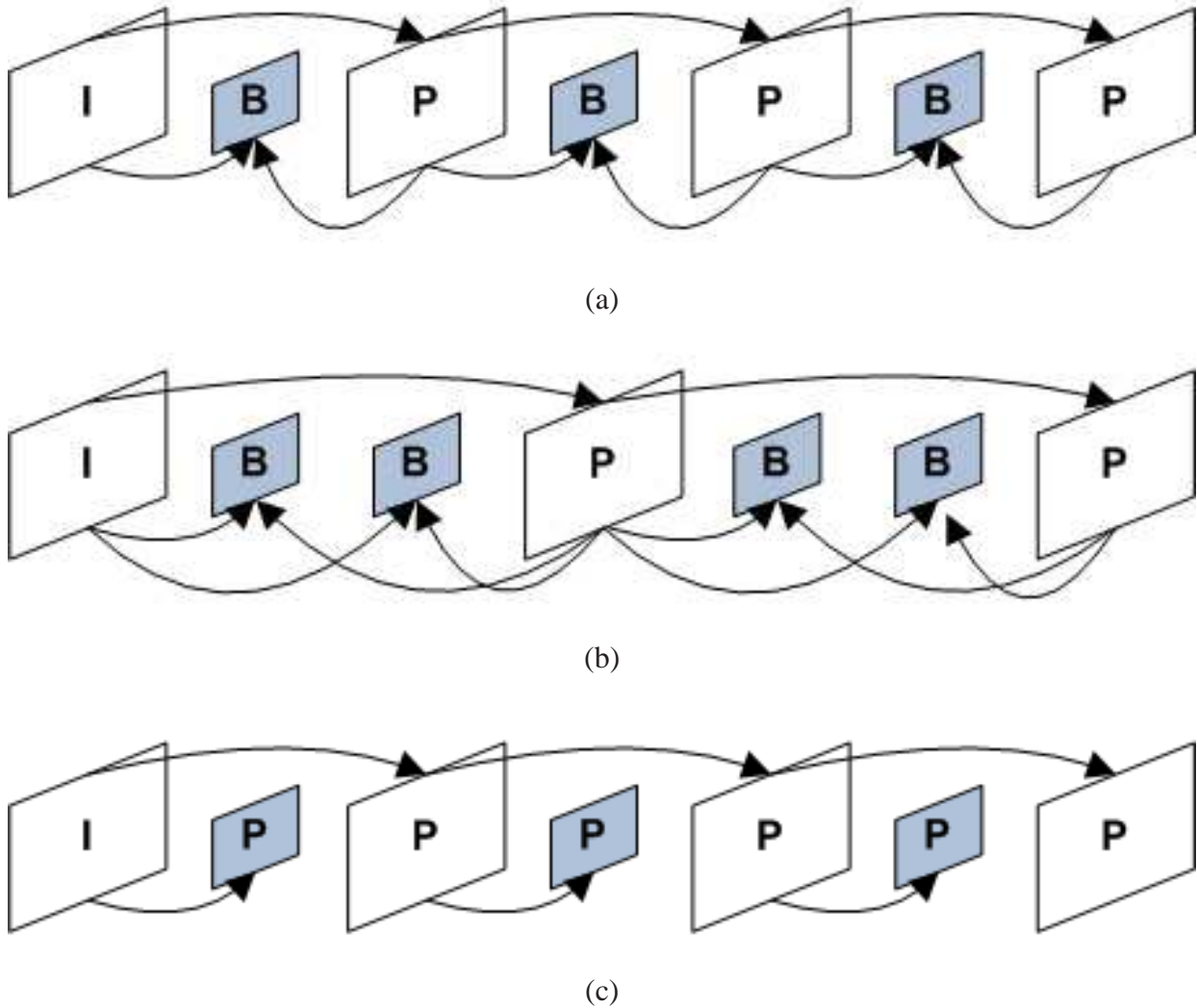


Figura 4.1: Usos diferentes dos quadros NRWZ. (a) um quadro intermediário tipo *B*, (b) dois quadros intermediários tipo *B*, (c) um quadro tipo *P*.

um deles é um *bit-stream* válido para um determinado decodificador. O SVC possibilita adaptabilidade temporal, espacial e de SNR. Porém, o foco do SVC não é a diminuição da complexidade. A Figura 4.2 (a) mostra um codec com adaptabilidade temporal. A primeira camada é formada pelos quadros H_0 , a segunda pelos quadros H_0 e H_1 , e a terceira por todos os quadros. Para codificadores híbridos, como o H.264/AVC, a adaptabilidade temporal pode ser alcançada simplesmente restringindo os quadros de referência a quadros da mesma camada. A Figura 4.2 (b) representa a combinação da adaptabilidade temporal e espacial que pode ser implementada no SVC. Note que ambas camadas são codificadas tradicionalmente, logo o processo de estimação de movimento é realizado em ambas camadas. Desta forma, o

esforço computacional aumenta proporcionalmente com o número de camadas. Cada camada é totalmente decodificável, exceto se a predição entre camadas (*inter-layer*) for utilizada na codificação. A codificação entre camadas permite usar quadros de camadas inferiores como referência para prever quadros de camadas superiores [76]. Na Figura 4.2 (c) mostra-se a adaptabilidade espacial do *framework* proposto. O codificador somente realiza estimação de movimento para os quadros-chave e quadros NRWZ na camada base, reduzindo a complexidade devido à codificação em baixa resolução. A informação de alta frequência dos quadros NRWZ é codificada mediante um codificador Wyner-Ziv. A camada de alta resolução, ou camada Wyner-Ziv, não representa um sinal de vídeo codificado. Esta camada é na verdade uma camada de realce da camada base, já que é necessário primeiro decodificar a camada base para poder decodificar a camada Wyner-Ziv.

4.2 ARQUITETURA DO CODEC

Nesta seção serão descritos o codificador e decodificador do modo de codificação com resolução mista. Este modo pode ser implementado como um modo opcional de codificação. Quando não é necessária a codificação a baixa complexidade, a sequência pode ser codificada mediante codificação híbrida tradicional.

4.2.1 Codificador

No geral, um quadro pode ser codificado como *inter*-quadro, usando predição baseada em múltiplos quadros de referência previamente codificados. Alguns codecs, como o H.264/AVC, usam informação dos elementos de sintaxe (*syntax elements*). Estes elementos incluem dados de movimento usados na predição do modo *Direct-B* [6] nos quadros tipo *B*, e para a geração dos MVs nas funções de estimação de movimento rápidas. Mais detalhes sobre o modo *Direct-B* e elementos de sintaxe podem ser encontrados em outros trabalhos [6].

Na arquitetura proposta, os quadros-chave são codificados mediante um codificador convencional DPCM/DCT com resolução espacial completa, podendo ser do tipo *I*, *P* ou *B*. No modo de codificação NRWZ, mostrado na Figura 4.3, a codificação é feita da seguinte maneira:

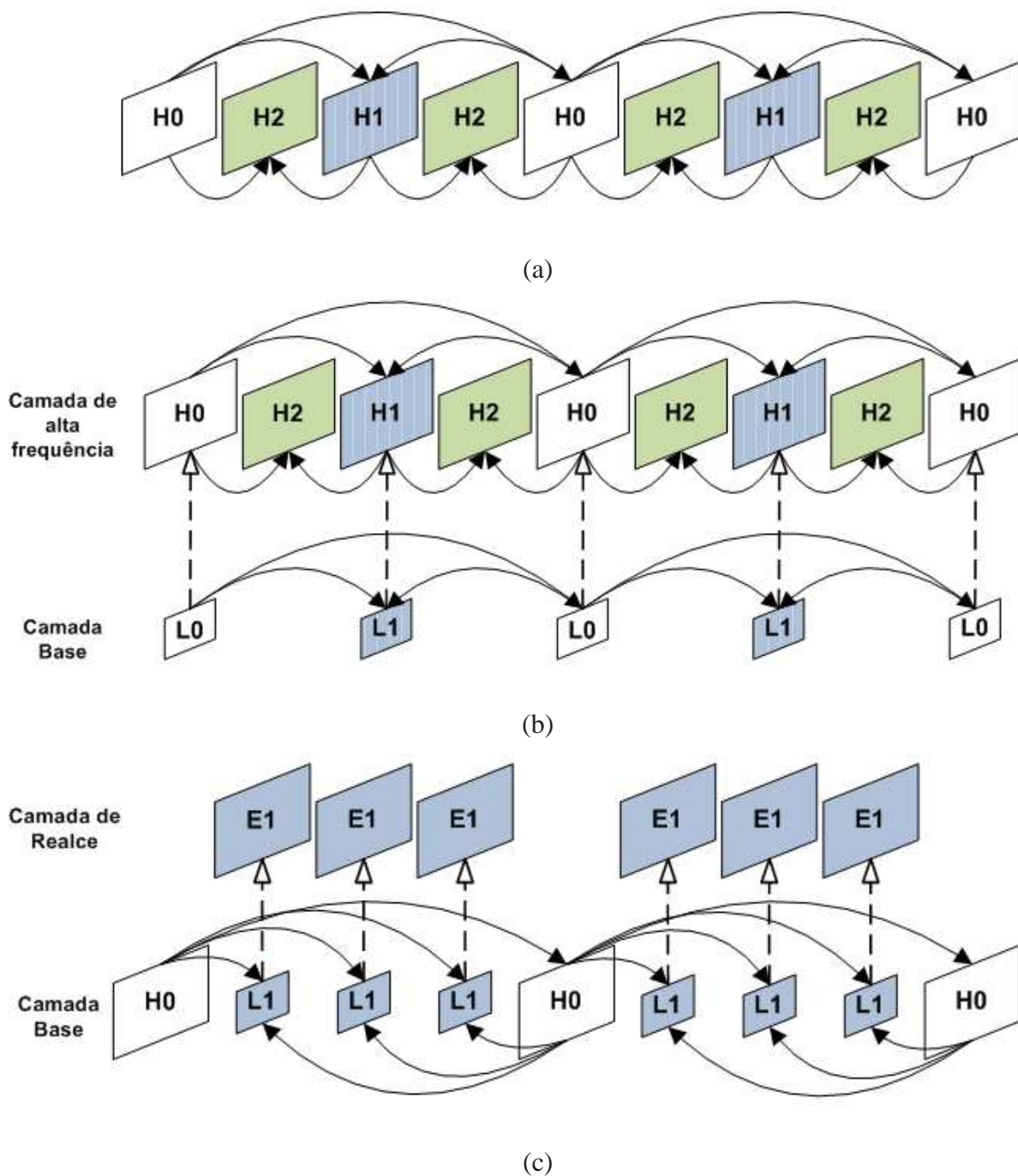


Figura 4.2: Três cenários de codificação adaptável. (a) adaptabilidade temporal, (b) adaptabilidade temporal e espacial que o SVC permite, (c) resolução mista.

- Todos os quadros da lista de referência e o quadro fonte são dizimados por um fator $2^n \times 2^n$, onde n pode ser selecionado dependendo da redução de complexidade requerida.
- Depois, o quadro fonte em baixa resolução é codificado usando um codificador híbrido convencional, o que requer um menor esforço computacional do que codificar o quadro em

resolução completa. Esta codificação em baixa resolução forma a camada base. O passo de quantização, QP , usado na codificação do quadro em baixa resolução é o mesmo que seria usado na codificação em resolução normal para obter uma determinada qualidade. Note que os elementos de sintaxe são também transformados para poderem ser utilizados na codificação em baixa resolução.

- Para criar a camada de realce, ou camada Wyner-Ziv, o codificador calcula a diferença entre o quadro fonte original com resolução normal e o quadro reconstruído NRWZ, codificado a baixa resolução, após ser interpolado. A interpolação do quadro em baixa resolução pode ser feita utilizando qualquer filtro clássico de interpolação [77].
- O quadro residual é mandado ao decodificador mediante codificação Wyner-Ziv. É fácil notar que a redução da complexidade é proporcional ao fator de dizimação e a quantidade de quadros NRWZ, uma vez que a estimação de movimento será feita agora num quadro de menor resolução.

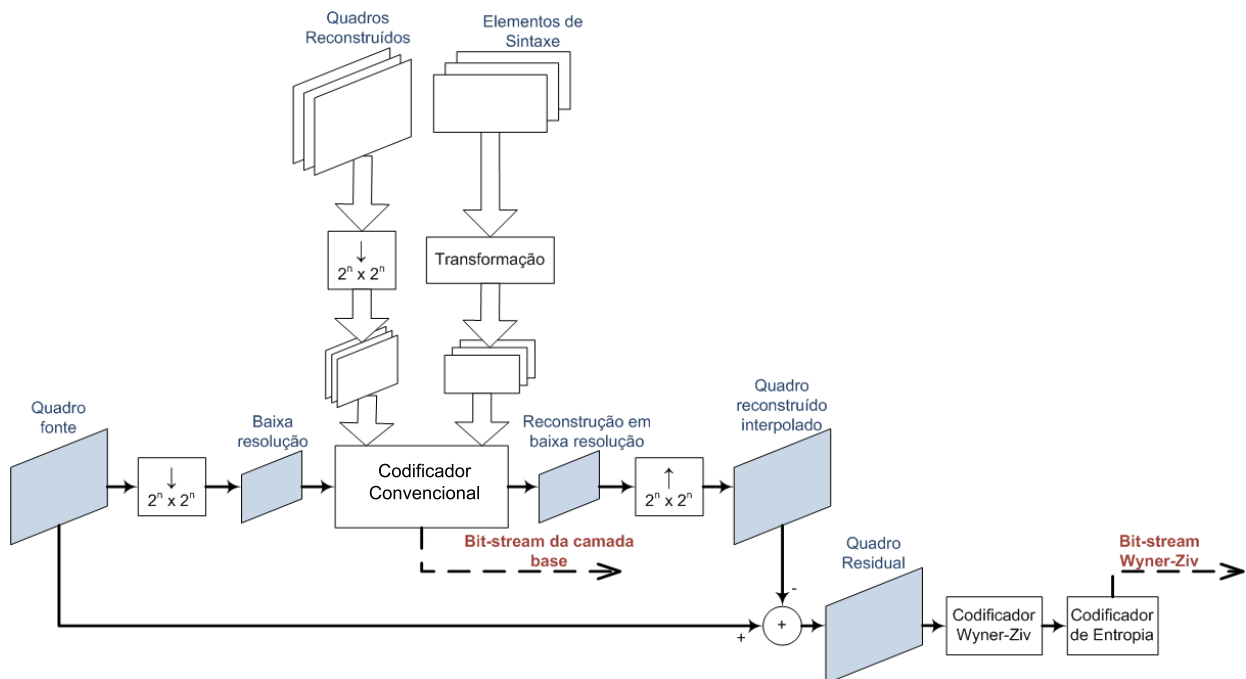


Figura 4.3: Arquitetura do codificador para os quadros NRWZ.

4.2.2 Decodificador

O decodificador é mostrado na Figura 4.4. O fluxo de dados é descrito a seguir:

- O quadro em baixa resolução é decodificado usando um decodificador convencional híbrido.
- O quadro NRWZ decodificado é interpolado, mediante o mesmo interpolador utilizado no codificador. O quadro NRWZ decodificado e interpolado é uma versão de baixa qualidade, porém de baixo esforço computacional de decodificação. Este quadro, junto aos quadros-chave, pode ser reproduzido com pouco custo de complexidade, por exemplo, em tempo real por um aparelho móvel com restrição de energia.
- O processo de decodificação da camada realce tem como primeiro passo a geração da informação lateral. O quadro interpolado e os quadros-chave previamente decodificados são usados para criar o denominado quadro em semi super resolução (SSR). O processo é chamado de semi super resolução porque os quadros chave já estão em resolução alta. No processo tradicional de super resolução, todos os quadros encontram-se em baixa resolução.
- Uma vez que o quadro em SSR é gerado, ele é subtraído do quadro decodificado interpolado, gerando assim a informação lateral. Note que esta informação lateral é uma versão ruidosa do quadro residual entre o quadro fonte original e o quadro reconstruído interpolado no codificador.
- O quadro residual é apresentado ao decodificador de canal, para corrigir os erros mediante a camada Wyner-Ziv. Este processo pode ser iterativo como detalhado na Seção 4.4.
- O quadro residual corrigido é adicionado ao quadro interpolado, gerando a versão de alta qualidade do quadro NRWZ.

4.3 CODIFICADOR WYNER-ZIV

O codificador Wyner-Ziv implementado trabalha no domínio da transformada. Em codecs onde diferentes tamanhos de transformada podem ser utilizados. Por exemplo, no H.264/AVC *Fidelity Range Extensions* [78], o maior tamanho de bloco para transformação é preferível, mas qualquer tamanho pode ser utilizado.

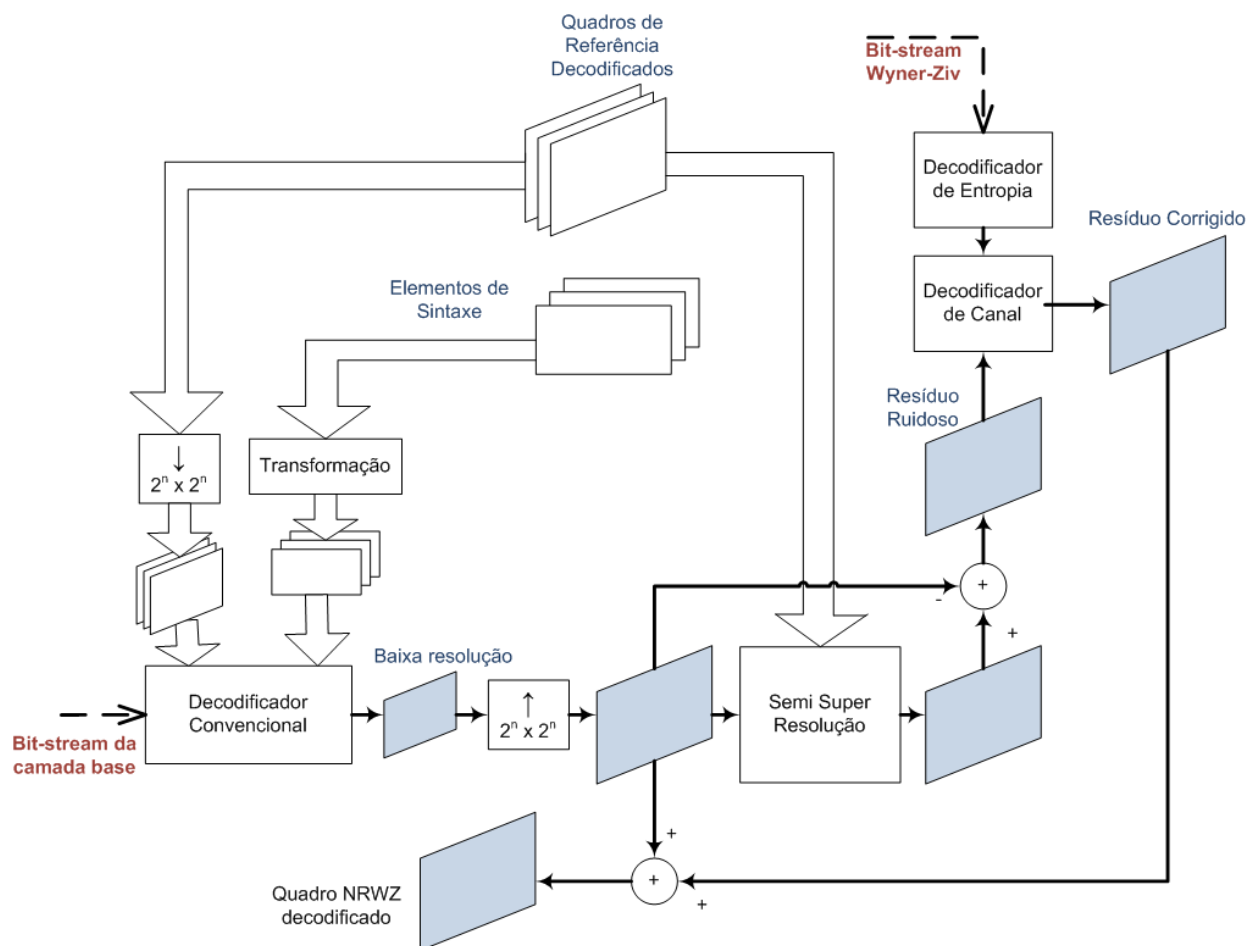


Figura 4.4: Arquitetura do decodificador para os quadros NRWZ.

O modelo assume que os coeficientes transformados do quadro residual seguem uma distribuição Laplaciana com desvio padrão σ_X . Foi provado em diferentes trabalhos que este é um modelo válido [17, 33]. Assim, de uma forma simplificada, a codificação Wyner-Ziv é feita da seguinte forma:

- O quadro residual, formado pela diferença entre a reconstrução interpolada do quadro NRWZ e o quadro original, é levado ao domínio da transformada DCT.
- Um processo de quantização é aplicado com passo de quantização QP_{wz} , não necessariamente igual ao QP utilizado na codificação convencional do quadro em baixa resolução da camada base.
- A camada de realce é formada por índices de *cosets* sem memória, em inglês *memoryless cosets*, gerados a partir dos coeficientes transformados quantizados.

Se a variável aleatória X denota os coeficientes transformados do quadro residual, X poderá tomar o valor x cujo valor quantizado será denotado por q . Logo, a variável aleatória que representa os coeficientes quantizados será $Q = \phi(X, QP_{wz})$. O alfabeto de Q é determinado por $\Omega_Q = \{-q_{max}, -q_{max-1}, \dots, -1, 0, 1, \dots, q_{max-1}, q_{max}\}$. Os limites $-q_{max}$ e q_{max} são determinados pelo passo de quantização QP_{wz} .

Como mencionado anteriormente, o codificador Wyner-Ziv não transmite os coeficientes residuais quantizados, mas índices de *cosets*. Os *cosets* são calculados gerando a variável aleatória $C = \psi(Q, M) = \psi(\phi(X, QP_{wz}), M)$, onde M é o módulo dos *cosets* [79]:

$$C = \psi(Q, M) = \begin{cases} (q) - M \lfloor q/M \rfloor, & (q) - M \lfloor q/M \rfloor < M/2 \\ (q) - M \lfloor q/M \rfloor - M, & (q) - M \lfloor q/M \rfloor \geq M/2 \end{cases} \quad (4.1)$$

Logo, o alfabeto de C será $\Omega_C = \{\lfloor -(M-1)/2 \rfloor, \dots, -1, 0, 1, \dots, \lfloor (M-1)/2 \rfloor\}$. A equação (4.1) garante que os *cosets* estão centrados em 0. Na implementação do modo de codificação Wyner-Ziv nesta tese os parâmetros $\{QP_{wz}, M\}$ variam para cada frequência (m, n) do coeficientes $x = X(m, n, t)$, e inclusive podem variar de macrobloco a macrobloco. Os parâmetros de codificação $\{QP_{wz}, M\}$ são escolhidos dependendo da estimação do ruído de correlação Wyner-Ziv (ver Capítulo 5). O número de coeficientes transmitidos em forma de *cosets* dentro de um macrobloco também varia. Os coeficientes que não são transmitidos são iguais a zero.

Se o *bin* quantizado q corresponde ao intervalo $[x_l(q), x_h(q)]$, logo a probabilidade de que o *bin* $q \in \Omega_Q$, e a probabilidade que o índice de *coset* $c \in \Omega_C$ são dadas pelas funções de massa de probabilidade:

$$p(q) = \int_{x_l(q)}^{x_h(q)} f_X(x) dx \quad (4.2)$$

$$p_C(c) = \sum_{q \in \Omega_Q, \psi(q, M) = c} p(q) = \sum_{q \in \Omega_Q, \psi(q, M) = c} \int_{x_l(q)}^{x_h(q)} f_X(x) dx, \quad (4.3)$$

onde $f_X(x)$ é a fdp (função densidade de probabilidade) de X . Como a distribuição $p_C(c)$ é simétrica para valores de M ímpares e possui moda zero, o codificador de entropia para q que já existe no codec convencional pode ser re-utilizado para c . Porém, esta arquitetura permite que

um codificador de entropia específico para os índices de *cosets* seja utilizado. Um codificador de entropia externo pode ser mais eficiente, mas usar o mesmo codificador de entropia existente no codec convencional minimiza as modificações necessárias para inserir o modo de codificação Wyner-Ziv. Neste trabalho modificamos um codificador existente do H.264/AVC para codificar os *cosets* (ver Capítulo 6). Assim o codificador terá um melhor desempenho e não será necessário introduzir um codificador de entropia externo. Exemplos das funções massa de probabilidade são mostradas nas Figura 4.5.

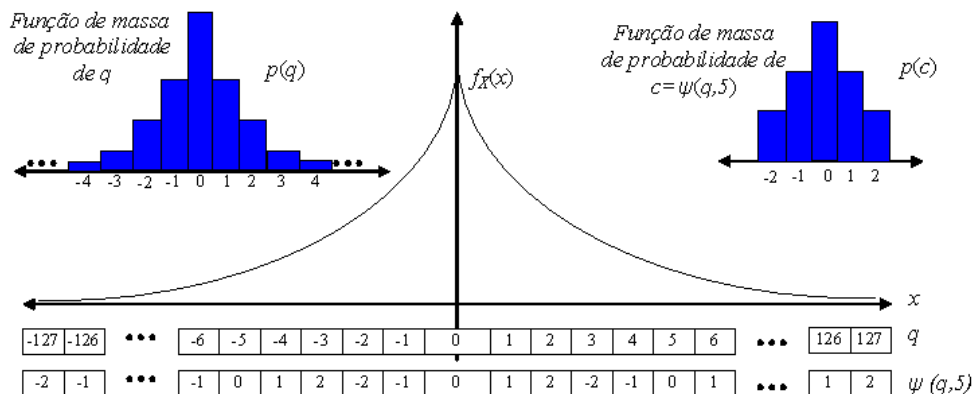


Figura 4.5: Funções massa de probabilidade. A função $f_X(x)$ é a fdp dos coeficientes transformados residuais modelada como distribuição laplaciana. $p(q)$ é calculada para cada valor possível de q . $p_C(c) = \psi(QP_{wz}, M)$ depende de $f_X(x)$ e a quantidade de valores possíveis é determinada por M , neste caso $M = 5$.

Note que não há canal de retorno. Idealmente, os parâmetros $\{QP_{wz}, M\}$ são escolhidos utilizando as estatísticas correlatas entre o quadro residual e a informação lateral. Se Y representa a versão ruidosa do quadro residual no decodificador, ou seja, a informação lateral, logo se pode assumir que:

$$Y = X + Z, \quad (4.4)$$

onde Z é o ruído de correlação Wyner-Ziv. Este modelo é utilizado em diversos trabalhos [33, 17, 67, 80] onde o ruído de correlação Z é simplificado assumindo que não possui correlação com X . No modelo probabilístico proposto, também assumimos que Z e X são decorrelacionados. Esta hipótese simplifica as deduções matemáticas e, após realizar simulações, verificamos que Z

pode ser modelado como uma variável aleatória com distribuição gaussiana e desvio padrão σ_Z . Logo, como não existe canal de retorno, uma adequada escolha dos parâmetros de codificação depende da correta estimação de $\{\sigma_X, \sigma_Z\}$. A estimação de $\{\sigma_X, \sigma_Z\}$ deve levar em consideração a qualidade desejada, ou distorção permitida. Note que Y não é quantizada na arquitetura proposta do decodificador, logo tem uma maior precisão que coeficientes quantizados. Mesmo que o valor exato de σ_X possa ser obtido no codificador, deve-se trabalhar com o valor estimado para garantir que o decodificador e o codificador estão trabalhando no mesmo ponto de operação (usando os mesmos parâmetros $\{QP_{wz}, M\}$).

Para decodificar, a função de reconstrução ótima (MSE), $\hat{X}_{YC}(y, c)$, baseada na informação lateral y e no índice de *coset* recebido c , é dada por:

$$\begin{aligned}\hat{X}_{YC}(y, c) &= E(X|Y = y, C = c), \\ \hat{X}_{YC}(y, c) &= \frac{\sum_{q \in \Omega_Q, \psi(q, M) = c} \int_{x_l(q)}^{x_h(q)} x f_{X|Y}(x, y) dx}{\sum_{q \in \Omega_Q, \psi(q, M) = c} \int_{x_l(q)}^{x_h(q)} f_{X|Y}(x, y) dx},\end{aligned}\tag{4.5}$$

onde $[x_l(q), x_h(q)]$ é o intervalo que corresponde ao *bin* quantizado q .

As estatísticas $\{\sigma_X, \sigma_Z\}$ estimadas para selecionar os parâmetros de codificação são também utilizadas no cálculo da equação (4.5). Porém, o cálculo exato de (4.5) é difícil e lento, sendo uma melhor solução usar aproximações obtidas a partir de tabelas pré-calculadas armazenadas no decodificador. As tabelas podem ser normalizadas para $\sigma_X = 1$ e diferentes valores de σ_Z . No processo de decodificação, estes valores podem ser adaptados usando a estimação de $\{\sigma_X, \sigma_Z\}$. Um exemplo do princípio de decodificação é ilustrado na Figura 4.6. Neste exemplo o valor recebido do índice de *coset* é $c = 2$ e a informação lateral existente no decodificador é $y = -1$. Para as funções de massa de probabilidade exemplificadas, o processo de decodificação calcula o valor esperado de x dentro de todas as possibilidades onde o valor quantizado q pode criar um *coset* $c = 2$. Assim, no exemplo, $\hat{x} = -3$.

Os coeficientes que foram igualados a zero na geração dos *cosets* na codificação, são igualados ao valor da informação lateral na decodificação.

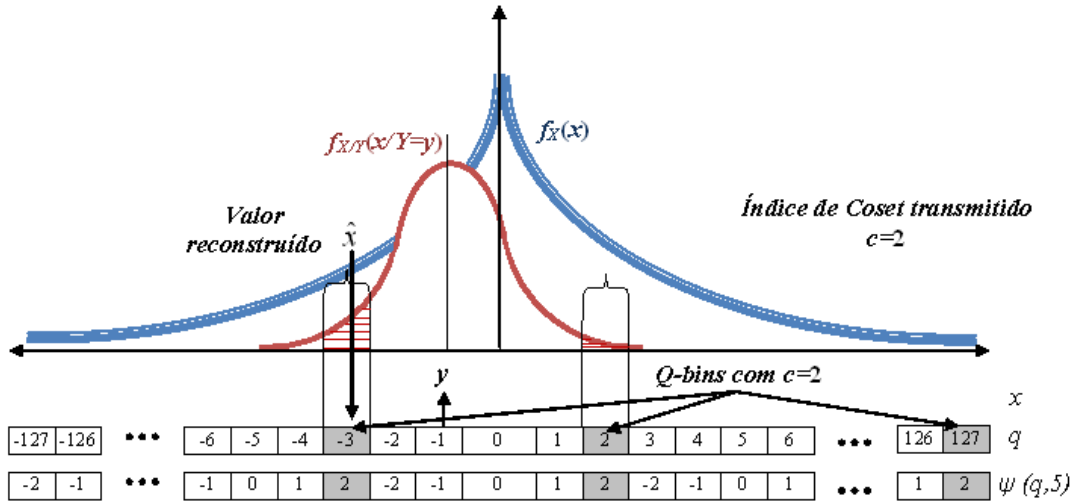


Figura 4.6: Exemplo de Decodificação. O índice recebido é $c = 2$, a informação lateral $y = -1$. Logo, o valor reconstruído \hat{x} dadas todas as possibilidades onde o valor q gerou o índice $c = 2$, para as funções de massa de probabilidade exemplificadas, é igual a -3 .

4.4 SEMI SUPER RESOLUÇÃO

A geração da informação tem como base o quadro obtido no processo de SSR, o qual pode ser feito de forma iterativa. Seja o quadro NRWZ, inicialmente decodificado a baixa resolução e interpolado, representado por $F_0(t)$. Seja a operação de SSR denotada pela função $SSR(F(t), FS)$, a qual obtém uma versão melhorada $F^{HR}(t)$ de $F_0(t)$ baseada nos quadros de referência FS . Seja a decodificação de canal representada por $CD(RF, b_{WZ})$, a qual corrige o quadro residual ruidoso RF usando o *bit-stream* da camada Wyner-Ziv b_{WZ} . A iteratividade do processo de decodificação para as iterações $i = 0, 1, 2, \dots, N$ pode ser descrita como:

$$F_i^{HR}(t) = SSR(F_i(t), FS) \quad (4.6)$$

$$F_{i+1}(t) = CD(F_i^{HR}(t) - F_0(t), b_{WZ}) + F_0(t). \quad (4.7)$$

Similarmente ao algoritmo baseado em exemplos (em inglês *example-based algorithm*) [81, 82, 83], procuramos recuperar a informação de alta frequência de um bloco interpolado mediante a procura em quadros previamente decodificados de blocos similares. O princípio é adicionar somente os componentes de alta frequência do bloco selecionado ao bloco interpolado.

A seqüência original em alta resolução possui quadros-chave e quadros NRWZ. O *framework* codifica os quadros NRWZ em baixa resolução e os quadros chave em alta resolução. O decodificador recebe uma seqüência de vídeo com resolução mista; o decodificador interpola os quadros NRWZ para obter uma seqüência na mesma resolução. Porém, os quadros NRWZ perderam informação de alta freqüência devido à dizimação e interpolação. Nosso algoritmo tenta recuperar os componentes de alta freqüência utilizando informação temporal presente nos quadros chave.

Primeiro, os quadros-chave imediatamente anterior e posterior ao quadro fonte NRWZ são filtrados utilizando um filtro passa-baixas. O filtro passa-baixas é implementado mediante dizimação e interpolação (usando os mesmos filtros do codificador). Assim, temos tanto os quadros NRWZ como os quadros chave na mesma resolução e sem detalhes de alta freqüência. Depois, um algoritmo de procura de blocos é aplicado usando o quadro NRWZ como fonte e os quadros chave filtrados como referência.

O algoritmo de procura funciona da seguinte forma: seja o quadro de referência $F(t) = B(t) + H(t)$, onde $B(t)$ é a versão dizimada e interpolada (filtrada) de $F(t)$, enquanto $H(t)$ é a alta freqüência, calculado como o resíduo entre $B(t)$ e $F(t)$. Para cada bloco 8×8 no quadro interpolado NRWZ, o melhor vetor de movimento, usando estimação *sub-pixel* [6], em cada quadro chave anterior e posterior, é calculado. Se os correspondentes blocos selecionados nos quadros-chave passado e futuro são denotados por $B_p(t)$ e $B_f(t)$ respectivamente, vários candidatos as predições são calculados como:

$$B(t) = \alpha B_p(t) + (1 - \alpha) B_f(t), \quad (4.8)$$

onde α assume valores entre 0 e 1. Conforme implementado, $\alpha \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$. Os valores possíveis de α foram obtidos após varias simulações. Logo, se a soma absoluta das diferenças (em inglês *sum of absolute differences*, SAD) do melhor candidato de um determinado bloco é menor que um limiar T , a alta freqüência correspondente do quadro de referência ($H_p(t)$ ou $H_f(t)$) é adicionada ao bloco do quadro NRWZ. Logo, a alta freqüência adicionada é dada por:

$$H(t) = \alpha H_p(t) + (1 - \alpha) H_f(t), \quad (4.9)$$

A Figura 4.7 ilustra este processo. Como é importante evitar adicionar ruído nos casos onde não existe um bloco suficientemente similar, usamos um fator de confiabilidade para escalar a alta frequência a ser adicionada. Assumimos que uma menor SAD implica em uma maior confiabilidade, logo maior conteúdo de alta frequência deve ser adicionado. Um outro parâmetro a utilizar para indicar o nível de confiabilidade é a taxa (R_n) para codificar um determinado macrobloco. Se R_n possui um valor alto, provavelmente se deve a um de dois fatores: (i) ou o símbolo codificado não aconteceu com muita frequência, o que significa que durante a codificação de entropia sua probabilidade é baixa; (ii) ou a informação residual do macrobloco for elevada, o que significa que existe uma diferença significativa entre o macrobloco atual e a sua melhor predição. Logo, é intuitivo pensar que devemos ter menos confiabilidade quando a taxa é elevada, já que pode indicar que não será possível recuperar as informações de alta frequência do macrobloco a partir de predições nos quadros adjacentes. O fator de confiabilidade multiplica cada *pixel* do bloco de alta frequência, antes de ser adicionado ao bloco NRWZ. A métrica do fator de confiabilidade usada foi:

$$f_c = 1 - \frac{\min(SAD + \lambda R_n, T)}{T}, \quad (4.10)$$

onde λ é um multiplicador de Lagrange. Note que se $SAD = R_n = 0$ então $f_c = 1$, o que significa que todo o conteúdo de alta frequência será adicionado. Por outro lado se $(SAD + \lambda R_n) = T$ então $f_c = 0$, então nenhuma informação será adicionada. Os valores de T e λ foram encontrados empiricamente usando seqüência de testes.

Se a decodificação iterativa é empregada, a primeira iteração é feita como descrito anteriormente. Nas demais iterações, certos parâmetros podem mudar. De iteração em iteração, a filtragem passa-baixas deve ser reduzida. Na nossa implementação a filtragem é eliminada após a primeira iteração. O *grid* para realizar o algoritmo de procura de blocos pode ser variado, para evitar o efeito de bloco. É importante notar que depois da primeira iteração, o quadro já possui informação de alta frequência. Então, depois da primeira iteração, o processo é similar, porém no lugar de adicionar a alta frequência, o conteúdo por inteiro do bloco é substituído pela versão não filtrada

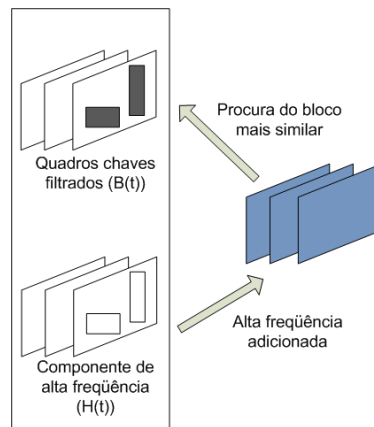


Figura 4.7: Procura da melhor predição. Depois de seleccionar o melhor predito nos quadros filtrados, a alta frequência do bloco é adicionada.

do predito. Em outras palavras, depois da primeira iteração, substitui-se $B(t) + H(t)$ em lugar de adicionar $H(t)$. Sendo que, depois da primeira iteração, o limiar T deve ser drasticamente reduzido, e gradualmente reduzido nas próximas iterações, para que a quantidade de blocos afetados diminua.

A Figura 4.8 exemplifica o processo de iteração da SSR, e na Figura 4.9 se mostra um quadro interpolado (depois de ser dizimado a um quarto da sua resolução), e o quadro gerado pela função de semi super resolução após a primeira iteração. Como medida de qualidade objetiva é utilizada a PSNR.

Neste capítulo foi apresentada a arquitetura do codec proposto, o qual opera com codificação em resolução mista. Também, foi detalhada a criação da camada Wyner-Ziv mediante *cosets* sem memória e a geração da informação lateral. No Capítulo 5 detalharemos como foi possível eliminar o canal de retorno e estimar o ponto de operação do codificador proposto. A codificação de entropia da camada Wyner-Ziv será abordada no Capítulo 6.

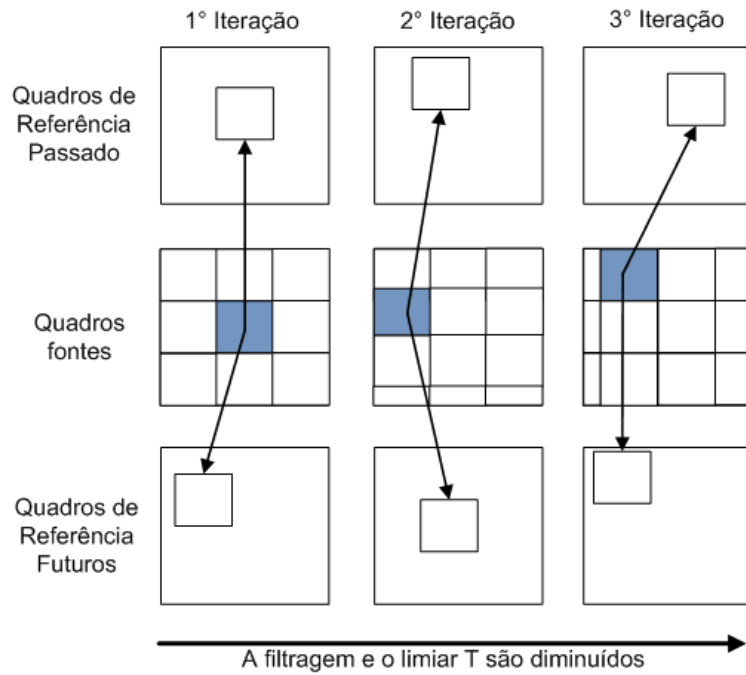


Figura 4.8: Iteração do processo de semi super resolução.



(a)

(b)

Figura 4.9: Comparação entre interpolação e semi super resolução. (a) Quadro interpolado com PSNR 34,12 dB. (b) Quadro em semi super resolução com PSNR 36,775 dB. A melhora é visualmente perceptível.

5 ESTIMAÇÃO DA CORRELAÇÃO ESTATÍSTICA PARA CODIFICAÇÃO DISTRIBUÍDA

O modo de codificação Wyner-Ziv com resolução espacial mista apresentado no Capítulo 4 foi implementado sem o uso de um canal de retorno. Isto faz possível a decodificação tanto *offline* como imediatamente após ter recebido o sinal codificado. Porém, o fato de não termos o canal de retorno dificulta a seleção correta dos parâmetros para que o codec trabalhe o mais próximo possível da região de codificação Slepian-Wolf. Logo, um mecanismo para selecionar corretamente os parâmetros de codificação, assim como uma correta estimação da correlação estatística dos sinais, é necessário.

Neste capítulo apresentamos um método para estimar a correlação estatística dos sinais assim como o ruído de correlação Wyner-Ziv, e um mecanismo para seleção ótima dos parâmetros de codificação usando *cosets* sem memória para os quadros NRWZ. Outros trabalhos usam códigos corretores de erro mais complexos como *turbo codes* ou LDPC, os quais são mais eficientes para determinadas arquiteturas [84]. Porém, foi mostrado que para a decodificação apresentada no Capítulo 4 o uso de código corretores de erro mais complexos não gera necessariamente um ganho significativo, comparado com *cosets* sem memória [85].

Note que estes mecanismos que serão descritos a seguir podem ser incorporados, com pequenas modificações, a outros *frameworks*, como na arquitetura PRISM [60, 61]. Estes modelos foram implementados usando os codecs H.263+ e H.264/AVC.

5.1 MODELO ESTATÍSTICO

Para realizar a seleção dos parâmetros de codificação $\{QP_{wz}, M\}$, assim como para estimar a correlação estatística, assumimos o modelo:

$$Y = \rho X + Z, \quad (5.1)$$

onde X é uma variável aleatória com distribuição Laplaciana, que representa o quadro residual Wyner-Ziv na codificação, Z é um ruído gaussiano aditivo não correlacionado com X , Y é a informação lateral presente no decodificador, ou seja, uma versão ruidosa do quadro residual obtida a partir do processo de SSR, e $0 < \rho < 1$ é um fator de atenuação, que deve decrescer em altas frequências. Note que este modelo é uma generalização do modelo clássico $Y = X + Z$, apresentado no Capítulo 4. Porém, podemos re-escrever (5.1) como $Y/\rho = X + Z/\rho$. Logo, a mesma codificação e decodificação dos *cosets* sem memória, da Seção 4.3, bem como o mesmo procedimento de estimação, podem ser aplicados simplesmente substituindo σ_Z^2 por $(\sigma_Z/\rho)^2$ e Y por Y/ρ durante a decodificação.

Assim, para simplificar certas deduções matemáticas, o modelo usado, na Seção 5.3, será $Y = X + Z$ enfatizando que na nossa implementação foi usado o novo modelo $Y = \rho X + Z$ substituindo σ_Z por (σ_Z/ρ) .

5.2 ESTIMAÇÃO DA CORRELAÇÃO ESTATÍSTICA

O modelo é específico para cada banda de frequência (BF) no macrobloco. Cada BF é definida como uma diagonal num bloco transformado como ilustrado na Figura 5.1.

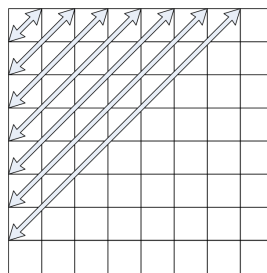


Figura 5.1: As primeiras sete bandas de frequência em um bloco 8×8 de coeficientes transformados

A correlação estatística entre os sinais X e Y é dependente do passo de quantização QP_t utilizado para codificar tanto os quadros-chave, quanto os quadros de baixa resolução. Além do

passo de quantização, outras informações podem ser obtidas da camada base para serem usadas no processo de estimação. Como toda informação da camada base está presente tanto na codificação como na decodificação, nenhuma informação adicional ou *hash* é necessária para realizar a estimação. Uma forma alternativa seria transmitir explicitamente informação estatística dos sinais, mas no nosso trabalho assumimos que a única comunicação existente entre o codificador e decodificador é o sinal de vídeo codificado em duas camadas. Para gerar o modelo de estimação usamos um sistema baseado em treinamento. Para um grupo de quadros de treinamento são coletados dados correspondentes a X e Y para diferentes valores de QP_t , juntamente com a informação adicional obtida da camada base. A idéia é estimar σ_X, σ_Z e ρ que serão usados para selecionar $\{QP_{wz}, M\}$ (ver Seção 5.4) para a criação dos *cosets* e para a reconstrução MSE.

5.2.1 Estimação da variância dos coeficientes residuais Laplacianos

A variância dos coeficientes residuais Laplacianos (σ_X^2) não é a mesma para cada bloco do quadro codificado, já que não depende somente de QP_t e BF , mas também do conteúdo de alta frequência do bloco. Se o quadro original possui muita informação de alta frequência, é provável que a diferença entre o quadro reconstruído interpolado e o quadro original seja grande. Note que o conteúdo original de alta frequência do quadro não está presente no decodificador, mas certos parâmetros podem ser usados para estimar σ_X^2 . A quantidade de bordas presentes no bloco em baixa resolução pode ser utilizado. É intuitivo pensar que uma maior quantidade de bordas e contornos no bloco da camada base indica que o quadro original possui maior informação de alta frequência, enquanto a energia dos coeficientes de baixa frequência está mais relacionada ao passo de quantização QP_t . Como medida objetiva da quantidade de bordas do bloco em baixa resolução, usaremos a soma acumulada da diferença absoluta entre os *pixels* adjacentes dentro de uma linha e coluna, denotado por E . Então, para um bloco bidimensional denotado por D de tamanho $M \times N$, E é definido como:

$$E = \sum_{j=1}^N \sum_{i=1}^{M-1} |D(i, j) - D(i+1, j)| + \sum_{i=1}^M \sum_{j=1}^{N-1} |D(i, j) - D(i, j+1)|. \quad (5.2)$$

Logo, a estimação de σ_X^2 é modelada como uma função de QP_t, BF e E . Isto é:

$$\sigma_X^2 = f_1(QP_t, BF, E). \quad (5.3)$$

Para simplificar, assumimos que σ_X^2 é proporcional a QP_t^2 . Após, processar os dados de treinamento, foi visto que era suficiente modelar a dependência com relação a BF de forma linear, logo:

$$\sigma_X^2 = (k_{1,BF}E + k_{2,BF})QP_t^2 \quad (5.4)$$

onde $k_{i,BF}$ são constantes que variam para cada BF . Na Figuras 5.2 e 5.3, mostramos as aproximações lineares de σ_X^2/QP_t^2 versus E comparadas com os valores reais de um grupo de quadros de treinamento, usando H.263+ e H.264/AVC, respectivamente. Pode-se observar que a aproximação é particularmente boa para valores baixos de E .

5.2.2 Estimação do fator de atenuação

Para estimar o fator de atenuação ρ , usamos um modelo simplificado que depende somente de QP_t e BF :

$$\rho = f_2(QP_t, BF). \quad (5.5)$$

Se Γ_{BF, QP_t} representa os dados de treinamento para determinados valores de QP_t e BF , a estimação é feita como:

$$\rho_{BF, QP_t} = \underset{\rho}{\operatorname{argmin}} \sum_{(X,Y) \in \Gamma_{BF, QP_t}} (\|Y - \rho X\|^2) \quad (5.6)$$

Note que é esperado que para maiores valores de QP_t , e para coeficientes de alta frequência, as variáveis X e Y devem ser menos correlacionadas. Isso foi confirmado obtendo os valores de ρ com o conjunto de treinamento utilizado.

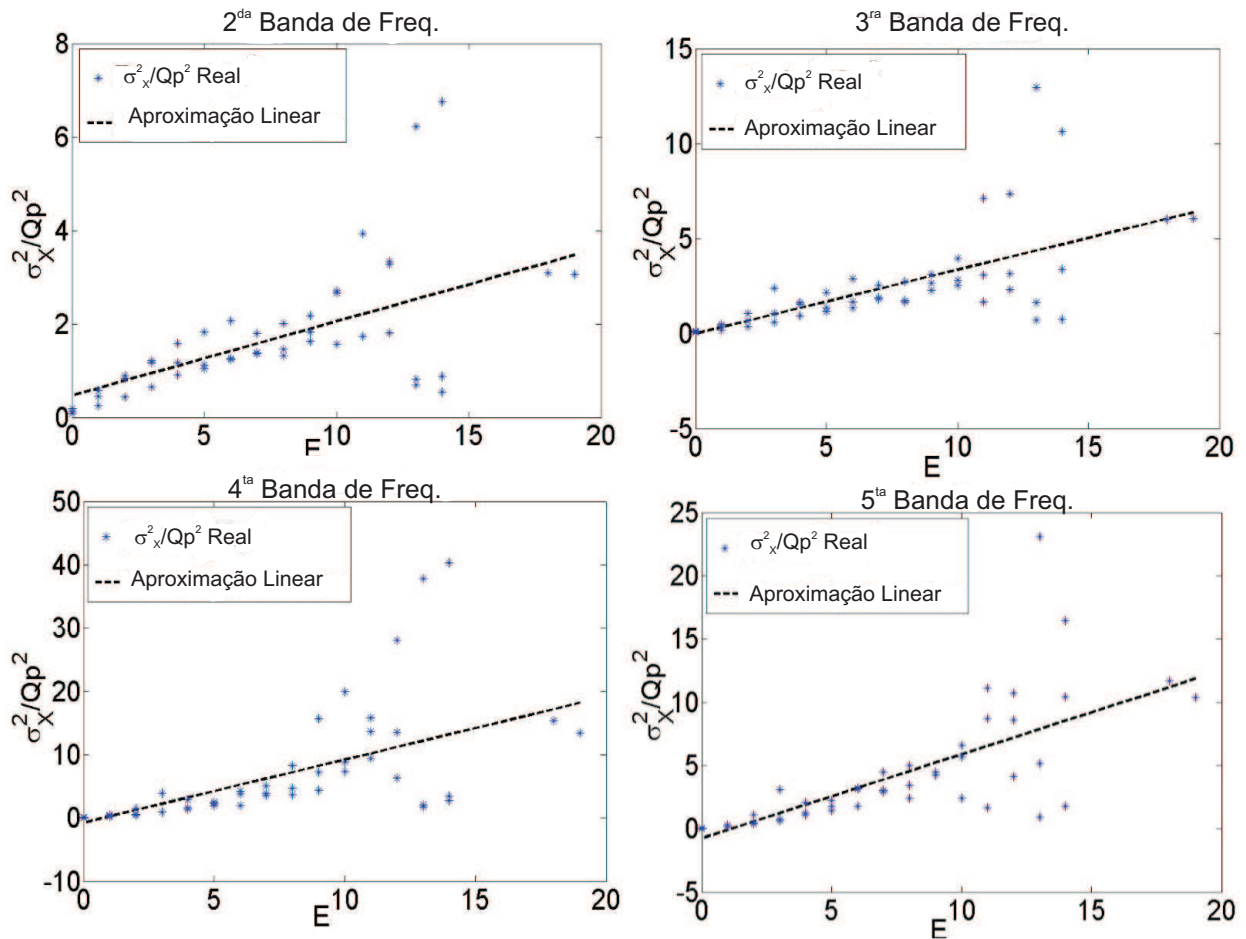


Figura 5.2: Estimação da variância dos coeficientes residuais Laplacianos. Valores reais de σ_x^2/QP_t^2 vs. E e sua aproximação linear usando H.263+ para 4 bandas de frequência.

Os valores de ρ obtidos do treinamento podem ser armazenados em tabelas pré-calculadas tanto no codificador como no decodificador. Os valores obtidos para um bloco 4×4 são mostrados na Tabela 5.1, após serem aproximados.

Tabela 5.1: Valores do fator de atenuação ρ para um bloco 4×4

	Banda de frequência						
	0 (DC)	1	2	3	4	5	6
ρ	0.90	0.85	0.75	0.55	0.25	0.15	0.10

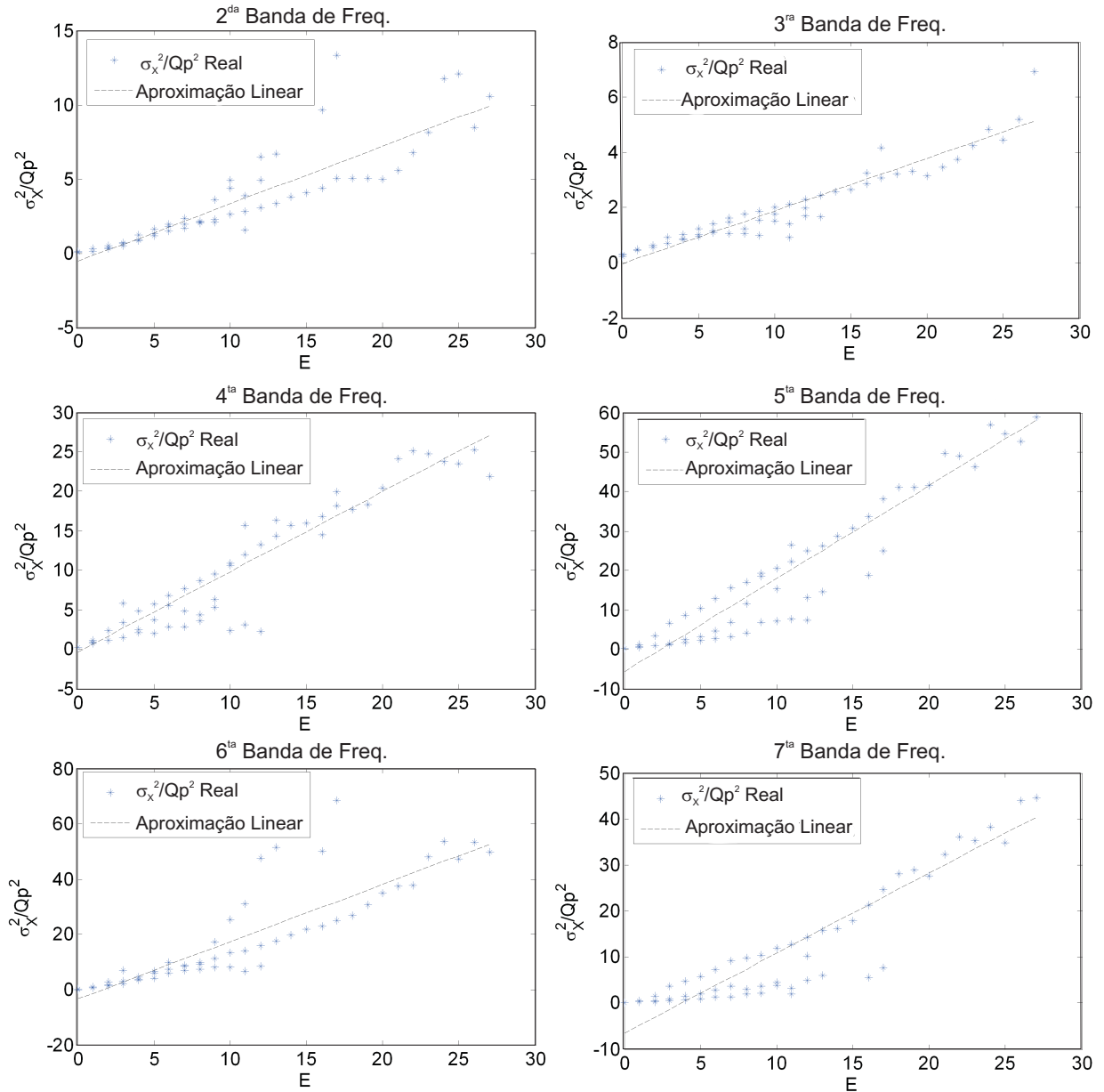


Figura 5.3: Estimação da variância dos coeficientes residuais Laplacianos. Valores reais de σ_x^2/QP_t^2 vs. E e sua aproximação linear usando H.264/AVC para 6 bandas de frequência.

5.2.3 Estimação da variância do ruído Gaussiano

A variância σ_Z^2 é calculada primeiro, nos dados de treinamento, $Z = Y - \rho X$. No nosso modelo assumimos que σ_Z^2 de um macrobloco da camada de realce depende da taxa R necessária para codificar o correspondente macrobloco na camada base, junto com QP_t , BF e E . Uma taxa elevada na camada base indica maior imprecisão na estimação de movimento em baixa resolução. Logo, o quadro resultante da SSR provavelmente deve gerar uma predição menos eficiente no

decodificador, gerando um aumento de σ_Z^2 . Porém, como R também depende de QP_t , foi utilizada uma taxa normalizada $R_n = R \times QP_t^2$, com o objetivo de remover o efeito de QP_t na taxa R . Agora, podemos modelar σ_Z^2 como:

$$\sigma_Z^2 = f_3(QP_t, BF, E, R_n). \quad (5.7)$$

Depois, assumimos que σ_Z^2 é proporcional a σ_X^2 para determinados BF e R_n , e os efeitos de QP_t e E fazem parte de σ_X^2 . Logo, a estimação pode ser modelada como:

$$\sigma_Z^2 = (k_{3,BF}R_n + k_{4,BF})\sigma_X^2 \quad (5.8)$$

Na Figura 5.4 são ilustradas as aproximações lineares de σ_Z^2/σ_X^2 vs. R_n comparadas com os valores reais dos quadros de treinamento, usando H.263+.

5.3 ESCOLHA DOS PARÂMETROS DE CODIFICAÇÃO

A escolha ótima dos parâmetros de quantização e módulo $\{QP_{wz}, M\}$ para os *cosets* dos quadros NRWZ, dado o modelo estatístico $Y = \rho X + Z$, deve seguir os seguintes passos:

- Obter a distorção esperada do codificador convencional para um determinado passo de quantização QP_t . A distorção esperada será denominada como D_t .
- Obter o conjunto de combinações de $\{QP_{wz}, M\}$ que produzam uma distorção igual a D_t . Já que os valores de $\{QP_{wz}, M\}$ são discretos é possível que não exista nenhuma combinação que gere exatamente a distorção desejada, logo se deve escolher as combinações que gerem uma distorção próxima, mas não maior que D_t .
- Dentro do conjunto escolhido, selecionar a opção que represente os valores ótimos de $\{QP_{wz}, M\}$. Ou seja, os valores que produzam menor distorção a menor taxa.

Para realizar os passos acima descritos deve-se fazer uma caracterização das funções de RD para os diferentes cenários de codificação. Estas funções serão dependentes da estatística da

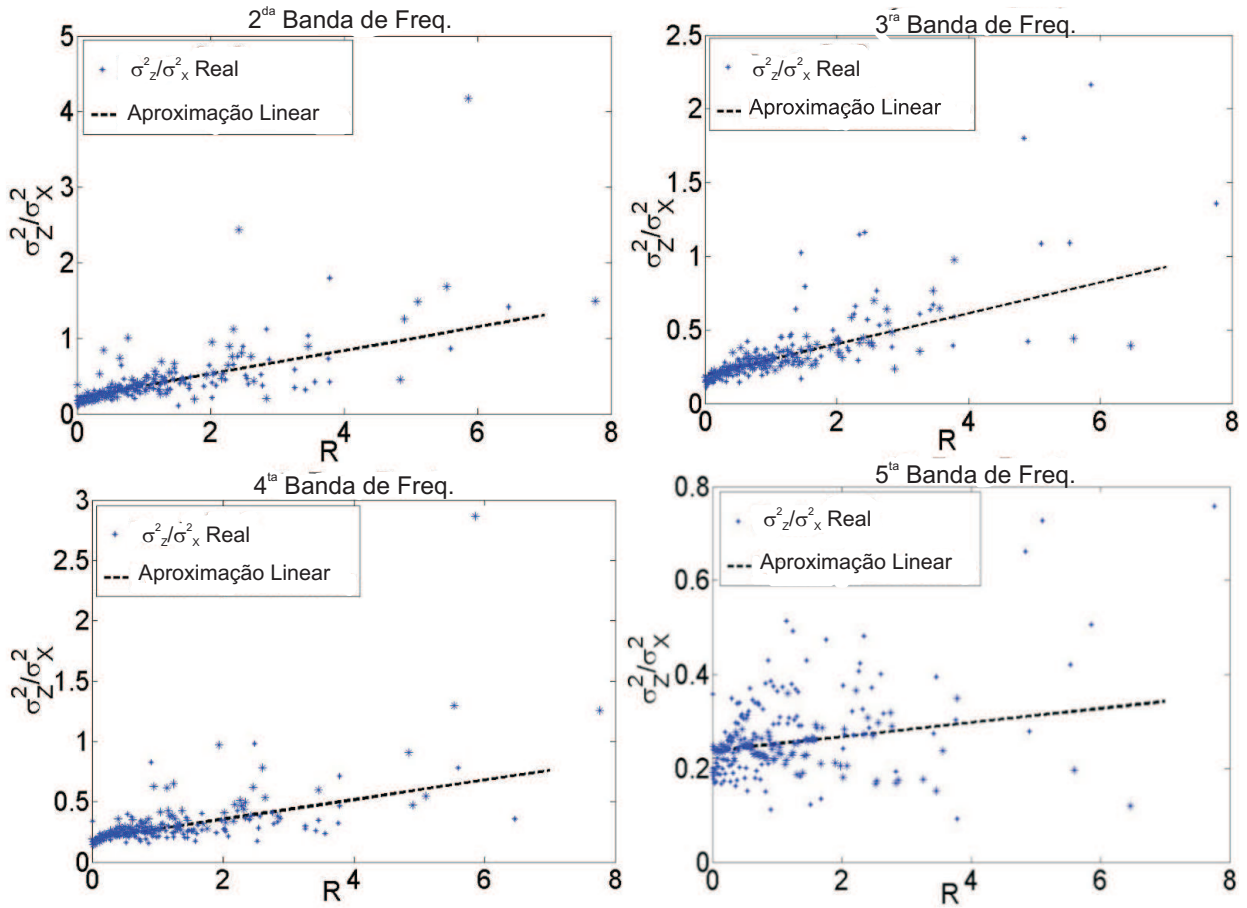


Figura 5.4: Estimação da variância do ruído Gaussiano. Valores reais de σ_X^2/σ_Z^2 vs. R_n e sua aproximação linear usando H.263+ para 4 bandas de frequência.

fonte e do ruído de correlação Wyner-Ziv. Em outras palavras estas funções são dependentes das variâncias σ_X^2 e σ_Z^2 , assim como do fator de atenuação do modelo estatístico ρ . Note que na realidade estes parâmetros serão estimados, conforme descrito na Seção 5.2. Logo a escolha de $\{QP_{wz}, M\}$ será ótima somente se a estimação da correlação estatística for correta.

A seguir, vamos revisar o mecanismo de escolha dos parâmetros de codificação para o modelo mais simples $Y = X + Z$, para facilitar as deduções matemáticas.

5.3.1 Codificação por cosets sem memória seguida de reconstrução MSE com informação lateral

O primeiro passo é obter as expressões para a taxa e distorção esperadas para os *cosets* sem memória descritos na Seção 4.3, para um determinado par $\{QP_{wz}, M\}$. Seja R_{YC} a taxa

assumindo um codificador ideal de entropia para os *cosets*, e D_{YC} a distorção dada a informação lateral y e índice de *coset* c . Pode ser provado (ver Apêndice I, Seção I.1) que os valores esperados são:

$$\begin{aligned}
E(R_{YC}) = & \\
& - \sum_{c \in \Omega_C} \left\{ \sum_{q \in \Omega_Q: \psi(Q, M) = c} \left[m_X^{(0)}(x_h(q)) - m_X^{(0)}(x_l(q)) \right] \right\} \\
& \times \log_2 \left\{ \sum_{q \in \Omega_Q: \psi(Q, M) = c} \left[m_X^{(0)}(x_h(q)) - m_X^{(0)}(x_l(q)) \right] \right\}, \quad (5.9)
\end{aligned}$$

$$\begin{aligned}
E(D_{YC}) = & \sigma_x^2 - \\
& \int_{-\infty}^{\infty} \left\{ \sum_{c \in \Omega_C} \frac{\left(\sum_{q \in \Omega_Q: \psi(Q, M) = c} [m_{X|Y}^{(1)}(x_h(q), y) - m_{X|Y}^{(1)}(x_l(q), y)] \right)^2}{\left(\sum_{q \in \Omega_Q: \psi(Q, M) = c} [m_{X|Y}^{(0)}(x_h(q), y) - m_{X|Y}^{(0)}(x_l(q), y)] \right)} \right\} \\
& \times f_Y(y) dy. \quad (5.10)
\end{aligned}$$

onde definimos $m_X^{(i)}(x) = \int_{-\infty}^x \nu^i f_X(\nu) d\nu$ e $m_{X|Y}^{(i)}(x, y) = \int_{-\infty}^x \nu^i f_{X|Y}(\nu, y) d\nu$.

Uma escolha possível é utilizar codificação a taxa zero, onde nenhuma informação é transmitida (i.e. $QP \rightarrow \infty$ ou $M = 1$). Logo se a taxa é 0, pode ser provado (ver Apêndice I, Seção I.2) que a distorção esperada baseada na reconstrução ótima utilizando somente Y é dada por:

$$\begin{aligned}
E(D_Y) = & \sigma_X^2 - \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} x f_{X|Y}(x, y) dx \right)^2 f_Y(y) dy \\
= & \sigma_X^2 - \int_{-\infty}^{\infty} m_{X|Y}^{(1)}(\infty, y)^2 f_Y(y) dy. \quad (5.11)
\end{aligned}$$

5.3.2 Fonte Laplaciana com Ruído aditivo Gaussiano

Usando o modelo de uma fonte Laplaciana, X , com ruído Gaussiano, Z , temos:

$$f_X(x) = \frac{1}{\sqrt{2}\sigma_x} e^{-\left|\frac{x\sqrt{2}}{\sigma_x}\right|}, \quad (5.12)$$

$$f_Z(z) = \frac{1}{\sqrt{2\pi}\sigma_z} e^{-\frac{1}{2}\left|\frac{z}{\sigma_z}\right|^2}. \quad (5.13)$$

Definindo $\beta(x) = e^{\frac{\sqrt{2}x}{\sigma_x}}$, temos

$$m_X^{(0)}(x) = \begin{cases} \frac{\beta(x)}{2}, & x \leq 0 \\ 1 - \frac{1}{2\beta(x)}, & x > 0 \end{cases}$$

$$m_X^{(1)}(x) = \begin{cases} \frac{\beta(x)}{2\sqrt{2}}(\sqrt{2}x - \sigma_x), & x \leq 0 \\ -\frac{1}{2\sqrt{2}\beta(x)}(\sqrt{2}x + \sigma_x), & x > 0 \end{cases} \quad (5.14)$$

Os momentos descritos por (5.14) podem ser usados no cálculo de $E(R_{YC})$ em (5.9). Se definimos:

$$\gamma_1(x) = \operatorname{erf}\left(\frac{\sigma_x x - \sqrt{2}\sigma_z^2}{\sqrt{2}\sigma_x\sigma_z}\right),$$

$$\gamma_2(x) = \operatorname{erf}\left(\frac{\sigma_x x + \sqrt{2}\sigma_z^2}{\sqrt{2}\sigma_x\sigma_z}\right) \quad (5.15)$$

onde:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt, \quad (5.16)$$

então, como $Y = X + Z$, temos:

$$f_{XY}(x, y) = \frac{1}{2\sqrt{\pi}\sigma_x\sigma_z} e^{-\left|\frac{x\sqrt{2}}{\sigma_x}\right|} e^{-\frac{1}{2}\left(\frac{y-x}{\sigma_z}\right)^2}, \quad (5.17)$$

$$f_Y(y) = \int_{-\infty}^{-\infty} f_{XY}(x, y) dx = \frac{1}{2\sqrt{2}\beta(y)\sigma_x} e^{\frac{\sigma_x^2}{\sigma_z^2}} [\gamma_1(y) + 1.0 - \beta(y)^2(\gamma_2(y) - 1.0)], \quad (5.18)$$

$$f_{X|Y}(x, y) = \frac{f_{XY}(x, y)}{f_Y(y)} = \frac{\sqrt{2}\beta(y)}{\sqrt{\pi}\sigma_z} \frac{e^{-\left|\frac{x\sqrt{2}}{\sigma_x}\right| - \frac{1}{2}\left(\frac{y-x}{\sigma_z}\right)^2 - \frac{\sigma_x^2}{\sigma_z^2}}}{[\gamma_1(y) + 1.0 - \beta(y)^2(\gamma_2(y) - 1.0)]}. \quad (5.19)$$

Dado $f_{X|Y}(x, y)$, os momentos $m_{X|Y}^{(i)}$ podem ser obtidos e usados no cálculo de $E(D_{YC})$ em (5.10).

$$m_{X|Y}^{(0)}(x, y) = \begin{cases} \frac{1}{[\gamma_1(y) + 1.0 - \beta(y)^2(\gamma_2(y) - 1.0)]} \beta(y)^2 [1 - \operatorname{erf}\left(\frac{\sigma_X(y-x) + \sqrt{2}\sigma_Z^2}{\sqrt{2}\sigma_X\sigma_Z}\right)], & x \leq 0 \\ 1 - \frac{1}{[\gamma_1(y) + 1.0 - \beta(y)^2(\gamma_2(y) - 1.0)]} [1 + \operatorname{erf}\left(\frac{\sigma_X(y-x) + \sqrt{2}\sigma_Z^2}{\sqrt{2}\sigma_X\sigma_Z}\right)], & x > 0 \end{cases} \quad (5.20)$$

$$m_{X|Y}^{(1)}(x, y) = \begin{cases} \frac{\beta(y)^2 [y + \sqrt{2}\frac{\sigma_X^2}{\sigma_Z}] [\operatorname{erf}\left(\frac{\sigma_X(y-x) + \sqrt{2}\sigma_Z^2}{\sqrt{2}\sigma_X\sigma_Z}\right) - 1] + \frac{\sqrt{2}}{\sigma_Z} \sigma_Z \beta(x)^2 e^{-\frac{1/2(\sigma_X(y-x) - \sqrt{2}\sigma_Z^2)^2}{\sigma_X^2\sigma_Y^2}}}{[\gamma_1(y) + 1.0 - \beta(y)^2(\gamma_2(y) - 1.0)]}, & x \leq 0 \\ \frac{\beta(y)^2 [y + \sqrt{2}\frac{\sigma_X^2}{\sigma_Z}] (\gamma_2(y) - 1) + [y + \sqrt{2}\frac{\sigma_X^2}{\sigma_Z}] [\operatorname{erf}\left(\frac{\sigma_X(y-x) + \sqrt{2}\sigma_Z^2}{\sqrt{2}\sigma_X\sigma_Z}\right) - \gamma_1(y)] + \frac{\sqrt{2}}{\sigma_Z} \sigma_Z e^{-\frac{1/2(\sigma_X(y-x) - \sqrt{2}\sigma_Z^2)^2}{\sigma_X^2\sigma_Y^2}}}{[\gamma_1(y) + 1.0 - \beta(y)^2(\gamma_2(y) - 1.0)]}, & x > 0 \end{cases} \quad (5.21)$$

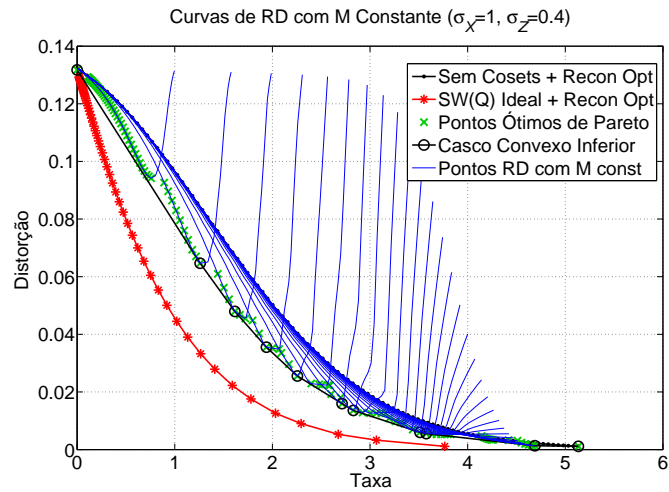
A função $\operatorname{erf}()$ pode ser calculada baseada em aproximações de polinômios de ordem 9 [86]. Assim, todos os valores esperados de taxa e distorção podem ser calculados usando integração numérica de $f_Y(y)$.

5.4 SELEÇÃO ÓTIMA DOS PARÂMETROS DE CODIFICAÇÃO

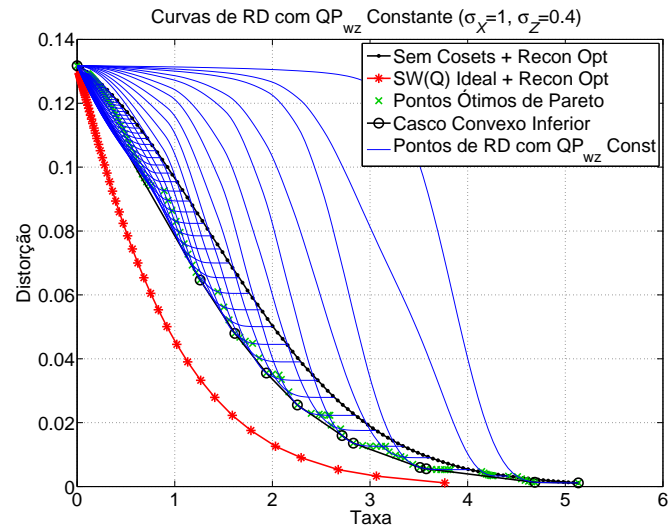
Mediante as equações (5.9) e (5.10) podem ser calculados os pontos de RD para o conjunto de combinações possíveis de $\{QP_{wz}, M\}$, dadas as estatísticas da fonte e do ruído de correlação σ_X e σ_Z , respectivamente. Na Figura 5.5, são apresentados os pontos de RD obtidos para um quantizador do tipo zona morta (em inglês *deadzone*) [87], para o caso específico de $\sigma_X = 1$ e $\sigma_X = 0.4$. Na Figura 5.5(a) cada curva de RD descrita como '*Pontos de RD de M const*' é gerada fixando M e mudando o QP em intervalos de 0.05. Porém, para a análise, assumimos QP contínuo. Assim, o caso onde $QP \rightarrow \infty$ para qualquer M corresponde à codificação de

taxa zero, e retorna o ponto de início de todas as curvas $\{0, E(D_Y)\}$, com $E(D_Y)$ expressada por (5.11). Alternativamente, este ponto pode ser visto como uma curva com $M = 1$. A Figura 5.5(b) mostra exatamente os mesmos resultados utilizando QP constante e mudando M de 1 até 60. Assumindo M contínuo, a medida que $M \rightarrow \infty$ o codificador tem um desempenho similar ao codificador convencional sem utilizar *cosets* mas ainda fazendo a reconstrução ótima baseada na informação lateral Y . Em ambas figuras, a curva para codificar convencionalmente Q mediante um codificador de entropia sem utilizar *cosets*, mas usando reconstrução ótima, é chamada de '*Sem cosets + Recon Opt*' (as equações podem ser encontradas no Apêndice I, Seção I.4). A curva correspondente à codificação ideal Slepian-Wolf dos coeficientes quantizados (Q) seguidos de reconstrução ótima é também mostrada ('*SW(Q) Ideal + Recon Opt*'), para detalhes ver Apêndice I, Seção I.3. Note que esta curva também representa os pontos inferiores do casco convexo dos pontos de RD obtidos mediante a codificação ideal Slepian-Wolf dos *cosets* C para todas as combinações de $\{QP_{wz}, M\}$, e representa o limite teórico do desempenho do presente *framework*.

Observando a Figura 5.5 pode ser verificado que nem todos os valores de QP_{wz} e M geram códigos adequados, já que podem não ser melhores que a codificação convencional ('*Sem cosets + Recon Opt*'). Porém, também pode ser verificado na Figura, que se $\{QP_{wz}, M\}$ forem escolhidos corretamente, a codificação utilizando *cosets* pode ter um melhor desempenho que a codificação convencional. Uma escolha sub-ótima da combinação $\{QP_{wz}, M\}$ pode ser feita achando o conjunto ótimo de Pareto [88]. Este conjunto é definido como o grupo de elementos onde a cada instante não existe nenhum elemento superior ao elemento pertencente ao conjunto. No nosso caso, os pontos de Pareto são os pontos que a uma determinada taxa possuem uma distorção menos que qualquer outro ponto das curvas. Estes pontos estão marcados como ' \times ' nas Figuras 5.5(a) e 5.5(b). Uma estratégia de escolha de parâmetros de codificação seria escolher o ponto de Pareto mais próximo dada uma distorção desejada D_t . Porém, a estratégia que retorna o melhor desempenho de RD é operar no casco convexo inferior do conjunto de pontos gerados por todas as combinações de $\{QP_{wz}, M\}$. Este grupo de pontos é um subconjunto dos pontos de Pareto. Estes pontos estão marcados como ' \circ ' nas Figuras 5.5(a) e 5.5(b). Como pode ser visto, não existe uma combinação de $\{QP_{wz}, M\}$ que pertença ao casco convexo inferior para qualquer valor de distorção e taxa, devido a que os valores de $\{QP_{wz}, M\}$ são discretos. Logo a estratégia



(a)



(b)

Figura 5.5: Curvas de RD obtidas para calcular o casco convexo inferior. Mostrando as curvas de codificação ideal Slepian-Wolf (SW), codificação convencional sem *cosets* com reconstrução ótima, e: (a) curvas RD com M constante e variando QP_{wz} , (b) curvas de RD com QP_{wz} constante e variando M .

adotada neste trabalho foi escolher o ponto do casco convexo inferior se ele existir, caso contrário utilizar o ponto de Pareto mais próximo. Note que ainda existe como melhorar a codificação, já que existe uma diferença entre a codificação por *cosets* e a codificação ideal Slepian-Wolf. Isto é esperado já que a codificação ideal Slepian-Wolf trabalha com a entropia condicional de $H(Q|Y)$,

enquanto as curvas da codificação por *cosets* são baseadas na entropia de $H(C)$. Porém, como estamos considerando o cenário onde não existe um canal de retorno, é razoável que a fronteira de Slepian-Wolf não seja alcançada.

5.4.1 Correspondência de Distorção

O objetivo da correspondência de distorção pode ser expresso em termos de (5.9) e (5.10). Se QP_t é o passo de quantização desejado durante a codificação convencional, usada para os quadros chave, logo é melhor expressar a distorção desejada D_t em termos de QP_t . A distorção esperada para um codificador convencional seguido de reconstrução ótima por MSE sem informação lateral é dada por (ver Apendice I Seção I.4):

$$\begin{aligned} E(D_Q) &= \sigma_X^2 - \sum_{q \in \Omega_Q} \frac{\left(\int_{x_l(q)}^{x_h(q)} x f_X(x) dx \right)^2}{\left(\int_{x_l(q)}^{x_h(q)} f_X(x) dx \right)^2} \\ &= \sigma_X^2 - \sum_{q \in \Omega_Q} \frac{\left(m_X^{(1)}(x_h(q)) - m_X^{(1)}(x_l(q)) \right)^2}{\left(m_X^{(0)}(x_h(q)) - m_X^{(0)}(x_l(q)) \right)^2} \end{aligned} \quad (5.22)$$

De (5.22) obtemos D_t para um determinado QP_t . Logo, podemos realizar a procura do código ótimo com uma distorção próxima e não superior a D_t . Na prática seria computacionalmente custoso realizar estas operações durante a codificação e decodificação. Porém, o mapeamento de QP_t a $\{QP_{wz}, M\}$ pode ser pré-calculado e armazenado em tabelas para diferentes valores de σ_Z e com $\sigma_X = 1$. Estas tabelas devem ser calculadas para diferentes valores de QP_t . Note que as tabelas estão normalizadas para $\sigma_X = 1$. Para utilizar estas tabelas, basta normalizar QP_t por σ_X antes de realizar a procura, e escalar o valor resultante de QP_{wz} . Note que também é necessário arredondar QP_{wz} para a mesma precisão que QP_t . Um exemplo destas tabelas é mostrado na Tabela 5.2. No trabalho criamos tabelas para $\sigma_Z \in \{2.0, 1.2, 1.1, 1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$, com o passo de quantização desejado normalizado $QP_t^* \in [0, 2.9]$ em incrementos de 0.05.

Voltando para o modelo estatístico $Y = \rho X + Z$, o procedimento para a escolha dos parâmetros de codificação $\{QP_{wz}, M\}$ utilizando as tabelas pré-calculadas é a seguinte: (i) estimar $\sigma_Z/(\rho \cdot \sigma_X)$ para determinar qual será a tabela a ser consultada; (ii) encontrar a entrada

mais próxima a distorção desejada normalizada $QP_t^* = QP_t/\sigma_X$; (iii) ler QP_{wz}^* normalizado e M ; e (iv) modificar QP_{wz}^* para obter o valor final $QP_{wz} = QP_{wz}^* \times \sigma_X$. Os valores de QP permitidos dependem do codificador utilizado. No nosso caso, H.264 permite variar o QP de 0.625 a 224 em escala logarítmica (estes valores são mapeados a números inteiros de 1 a 51 [6]).

Os mecanismos descritos no presente capítulo realizam a estimação da correlação estatística dos sinais, a estimação do ponto de operação do codec, e a seleção ótima dos parâmetros de codificação para a geração dos *cosets* sem a necessidade de utilizar um canal de retorno. No capítulo seguinte será detalhada a codificação de entropia dos *cosets*, gerando o *bit-stream* da camada de realce.

Tabela 5.2: Tabela de mapeamento para $\sigma_Z=1.1$

QP_t^*	QP_{wz}^*	M	QP_t^*	QP_{wz}^*	M
0.00	0.2	27	1.50	1.87	2
0.05	0.2	27	1.55	1.98	2
0.10	0.2	27	1.60	2.11	2
0.15	0.2	27	1.65	2.03	2
0.20	0.2	22	1.70	2.27	2
0.25	0.24	19	1.75	2.51	2
0.30	0.3	13	1.80	2.76	2
0.35	0.34	12	1.85	3.07	2
0.40	0.4	12	1.90	3.55	2
0.45	0.44	11	1.95	4.58	2
0.50	0.5	9	2.00	∞	1
0.55	0.54	9	2.05	∞	1
0.60	0.6	8	2.10	∞	1
0.65	0.66	8	2.15	∞	1
0.70	0.7	7	2.20	∞	1
0.75	0.77	6	2.25	∞	1
0.80	0.84	5	2.30	∞	1
0.85	0.9	4	2.35	∞	1
0.90	0.95	4	2.40	∞	1
1.00	1.02	4	2.45	∞	1
1.05	1.1	4	2.50	∞	1
1.10	1.17	3	2.55	∞	1
1.15	1.14	3	2.60	∞	1
1.20	1.27	3	2.65	∞	1
1.25	1.37	3	2.70	∞	1
1.30	1.47	3	2.75	∞	1
1.35	1.56	3	2.80	∞	1
1.40	1.66	3	2.85	∞	1
1.45	1.76	2	2.90	∞	1

6 CODIFICAÇÃO DA CAMADA DE REALCE

A arquitetura do modo de codificação Wyner-Ziv com resolução espacial mista, apresentado no Capítulo 4, gera duas camadas. A primeira é a camada base, e a segunda é a camada de realce ou camada Wyner-Ziv. A camada Wyner-Ziv é baseada na criação de *cosets* sem memória do quadro residual obtido do quadro original e o quadro NRWZ reconstruído interpolado. Para realizar uma codificação competitiva, os índices dos *cosets* são comprimidos sem perdas, ou seja, passam por um codificador de entropia. Como foi mostrado na Seção 4.3, é possível utilizar o codificador de entropia presente no codificador de vídeo tradicional para os *cosets*. Mas um codificador específico pode apresentar uma maior taxa de compressão. Se, por um lado, o uso de um codificador externo específico para os *cosets* pode melhorar o desempenho, por outro lado, utilizar um codificador externo aumentaria as modificações necessárias para incluir o modo de codificação Wyner-Ziv em um codificador híbrido tradicional. Assim, neste trabalho optamos por aperfeiçoar o codificador existente no codec tradicional para a camada de realce.

O codificador convencional híbrido utilizado foi o H.264/AVC, o qual possui dois codificadores de entropia: o codificador CABAC (*Context-Based Adaptive Binary Arithmetic Coding*) e o codificador CAVLC (*Context-Based Adaptive Variable Length Coding*). Sendo, que o CABAC está disponível somente no perfil principal (*Main Profile*) e no perfil alto (*High Profile*), e o CAVLC em todos os perfis (*Baseline, Main, High e Extended Profiles*) [6]. Logo, optamos por modificar o CAVLC. Assim, foi criado o D-CAVLC, do inglês *Distributed CAVLC*. Note que modificações similares podem ser feitas em qualquer outro codificador de entropia presente no codificador híbrido utilizado.

Neste Capítulo, inicialmente vamos descrever como a codificação CAVLC funciona, e depois detalharemos o D-CAVLC utilizado para a camada de realce.

6.1 CODIFICAÇÃO DE ENTROPIA NO H.264

No padrão H.264 a codificação de entropia usa códigos de tamanho variável ou codificação aritmética, dependendo do modo de codificação escolhido e do perfil utilizado [6]. As informações que precisam ser codificadas pelo codificador de entropia, para cada macrobloco 16×16 , são as seguintes:

- tipo do macrobloco;
- padrão do bloco codificado (em inglês *coded block pattern* ou *cbp*), que indica qual bloco 8×8 possui coeficientes diferentes de zero;
- o delta QP , ou variação do passo de quantização, utilizado somente se a otimização de RD estiver ativa;
- o índice do quadro de referência para a predição *inter*-quadro;
- os vetores de movimento, transmitidos também em forma preditiva;
- os coeficientes residuais para cada bloco 4×4 ou 2×2 .

No perfil de base (*Baseline Profile*) a codificação do bloco residual é realizada por CAVLC e as outras informações de cabeçalho são codificadas utilizando códigos Exp-Golomb [44]. Na codificação dos blocos residuais, o CAVLC codifica os coeficientes DC separadamente dos coeficientes AC. Os coeficientes DC de cada bloco de 4×4 pixels, dentro de um macrobloco de 16×16 , são agrupados e codificados em blocos de 4×4 para o componente de luminância e blocos de 2×2 pixels para os componentes de croma.

6.1.1 Codificação Exp-Golomb

Os códigos exponenciais de Golomb, ou simplesmente códigos Exp-Golomb [44], são códigos de tamanho variável com construção regular. Os códigos Exp-Golomb são construídos da seguinte forma:

$$[Nzeros][1][INFO], \quad (6.1)$$

onde *INFO* é um campo de *N*-bits o qual possui a informação. A palavra mais provável é codificada sem zeros ao início e sem informação, ou seja por um único bit igual a 1. Os dois códigos seguintes são codificados com um bit no campo de informação, os quatro seguintes possuem dois bits no campo de informação, e assim sucessivamente. Este processo é exemplificado na Tabela 6.1.

Tabela 6.1: Códigos Exp-Golomb

Número do Código	Código
0	1
1	010
2	011
3	00100
4	00101
5	00110
6	00111
7	0001000
8	0001001
...	...

A decodificação é feita da seguinte maneira:

- Ler os zeros antes do bit 1;
- Ler *N*-bits de informação, onde *N* é a quantidade de zeros lidos;
- o número do código é igual a $2^N + INFO - 1$.

Logo, o mapeamento entre os símbolos a serem codificados e seus respectivos códigos deve ser feito de tal forma que associe códigos menores aos símbolos mais frequentes. Por exemplo, o tipo de macrobloco mais frequente deve ser associado ao número de código 0. Existem, várias tabelas de mapeamento do H.264 para cada tipo de informação a ser codificada com códigos Exp-Golomb. Para o caso específico do cbp existem duas tabelas de mapeamento: uma tabela é utilizada quando a quadro foi do tipo *intra* e a outra quando o quadro for do tipo *inter*.

6.1.2 Seqüências de Zeros

Os coeficientes residuais quantizados precisam ser codificados da forma mais compacta possível. Por isso, O CAVLC trabalha com seqüências de zeros, ou em inglês *zero-runs*. Isto ocorre porque os coeficientes quantizados de um bloco residual normalmente são constituídos por um grande numero de zeros e uma pequena quantidade de coeficientes diferentes de zero. Logo, antes de realizar a codificação no CAVLC, os coeficientes são agrupados em pares do tipo {seqüências de zeros, nível}. A seqüências de zeros indica a quantidade de zeros que precede um determinado coeficiente diferente de zero, e o nível determina o valor desse coeficiente.

Para que a codificação por seqüências de zeros seja eficiente os coeficientes quantizados do bloco residual devem ser reordenados para poder agrupar a maior quantidade de zeros antes de cada nível. Pela característica da DCT os coeficientes residuais de alta freqüência possuem valores mais próximos de zero que os coeficientes de baixa freqüência. Portanto, o reordenamento dos coeficientes é feita em zig-zag conforme a Figura 6.1.

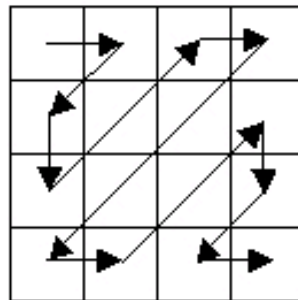


Figura 6.1: Ordem zig-zag de um bloco 4×4 .

6.1.3 CAVLC

O codificador de entropia CAVLC no codec H.264/AVC é aperfeiçoado para codificar seqüências de zeros do bloco residual ordenados em zig-zag. Os blocos podem ser de tamanho 4×4 ou 2×2 . O tamanho 2×2 é usado somente para codificar os coeficientes DC da componente de crominância, para todos os outros casos o tamanho de bloco é 4×4 . A codificação CAVLC usa certas características dos blocos residuais quantizados para codificar a informação. Por exemplo, os coeficientes residuais de alta freqüência são comumente seqüências de ± 1 , portanto, o módulo CAVLC codifica os coeficientes ± 1 de alta freqüência de forma separada. Esses coeficientes são

chamados de *trailing ones*. A codificação CAVLC codifica as informações na seguinte ordem:

1. o número de coeficientes diferentes de zero e *trailing ones*;
2. o sinal de cada *trailing one*;
3. o nível dos coeficientes diferentes de zero restantes;
4. o número total de zeros antes dos coeficientes;
5. cada seqüência de zeros antes de cada coeficiente diferentes de zero.

No primeiro passo, O CAVLC codifica a quantidade de coeficientes diferentes de zero e a quantidade de *trailing ones* como um único código. O número de coeficientes diferente de zero varia de 0 a 16, enquanto a quantidade de *trailing ones* varia de 0 a 3 (somente os últimos ± 1 são considerados como *trailing ones*). Note que 0 é uma possibilidade para o número de coeficientes porque o cbp é codificado para blocos 8×8 . O CAVLC seleciona os códigos entre quatro tabelas diferentes. Três tabelas contém códigos de tamanho variável e uma contém códigos de tamanho fixo. A escolha das tabelas depende do número de coeficientes diferentes de zero nos blocos previamente codificados posicionados acima e à esquerda no bloco atual. Cada uma das tabelas de códigos com tamanho variável possui 62 entradas. A primeira usa códigos menores para valores pequenos, a segunda para valores médios e a terceira para valores grandes. A codificação de tamanho fixo usa palavras de 6 bits para cada código (4 bits para o número de coeficientes e 2 para a quantidade de *trailing ones*).

Os sinais de cada *trailing one* são codificados no segundo passo com 1 bit (0 = positivo, 1 = negativo).

No terceiro passo, o nível de cada coeficiente diferente de zero, que não foi classificado como *trailing one*, é codificado a partir do coeficiente de mais alta freqüência até o coeficiente de menor freqüência. Para codificar os níveis, existem 7 tabelas diferentes. As tabelas são selecionadas dependendo do valor do último coeficiente codificado, conforme o algoritmo seguinte:

1. iniciar a codificação com a primeira tabela;
2. codificar o coeficiente de mais alta freqüência diferente de zero;
3. se o valor codificado é maior que um limiar predefinido, utilizar a próxima tabela para o próximo coeficiente.

A quantidade total de zeros que precede cada coeficiente diferente de zero é codificada no quarto passo. O fato de codificar a quantidade total de zeros implica que a seqüência de zeros do início do bloco em ordem zig-zag não precisa ser codificada. Se a quantidade de coeficientes diferentes de zero indica que todos os 16 coeficientes são diferentes de zero, então o número total de zeros não é codificado porque o decodificador sabe que será igual a zero. Logo, existem 15 tabelas de códigos diferentes para codificar o número total de zeros. Cada tabela depende do número de coeficientes diferentes de zero, que pode variar de 1 a 15.

No último passo são codificadas cada seqüência de zeros que precede a cada coeficiente. Com duas exceções:

- se não existem mais zeros a serem codificados, ou seja se a soma de todas as seqüências codificadas até o momento é igual à quantidade total de zeros, a seqüência de zeros não é codificada;
- também não é codificada a seqüência de zeros após o último coeficiente diferente de zero.

Para realizar a codificação o CAVLC escolhe entre 7 tabelas diferentes, de códigos com tamanho variável, dependendo da quantidade de zeros restantes no bloco.

6.2 CODIFICAÇÃO DA CAMADA WYNER-ZIV

A geração do *bit-stream* da camada de realce é feita de maneira similar à codificação de entropia da camada base. Porém, somente é necessário codificar o cbp e os dados residuais. Não existe informação de cabeçalho, como tipo do macrobloco, vetores de movimento, etc. Os parâmetros de codificação $\{QP_{wz}, M\}$ não são transmitidos, já que são determinados tanto no codificador como no decodificador a partir de informação da camada base (ver Capítulo 5). Uma escolha possível é codificar à taxa zero, o que significa utilizar $QP \rightarrow \infty$ ou $M = 1$. Portanto, o número de *cosets* gerados num bloco 4×4 é variável. Como foi mencionado, na Seção 5.2, os parâmetros de codificação são gerados para cada banda de freqüência (BF) no bloco.

Quando a seleção de $\{QP_{wz}, M\}$ feita pelo mecanismo de estimação indique que nenhum *coset* será gerado no macrobloco inteiro de 16×16 , ou seja quando $M = 1$ para todas as BFs de

cada bloco 4×4 , nenhuma informação é codificada, nem o cbp. Isto ocorre porque o decodificador possui um mecanismo similar de estimação. Logo, é possível deduzir no decodificador que nenhum *coset* foi gerado no macrobloco sem necessidade do cbp. Se uma ou mais *BFs* são selecionadas para criação de *cosets* ($M \neq 1$) para pelo menos um bloco 4×4 dentro do macrobloco 16×16 , então o cbp e a informação residual são codificados.

A codificação do cbp é feita utilizando os códigos Exp-Golomb, de maneira similar à camada base, com a única diferença que a tabela de mapeamento utilizada para todos os quadros é a tabela de predição *intra*. Como na seleção de parâmetros de codificação, é feita uma estimação da quantidade de *cosets* que deve ser criada, se for necessário codificar o cbp, o mais provável é que ele seja diferente de zero. A tabela de mapeamento *inter* utiliza códigos menores para cbp próximos a zero, enquanto a tabela de mapeamento *intra* utiliza códigos menores para valores maiores do cbp. Por isso, utiliza-se a tabela *intra* para os *cosets*. A informação residual é codificada pelo D-CAVLC.

6.2.1 D-CAVLC

Nesta seção vamos detalhar as modificações feitas na codificação CAVLC no padrão H.264/AVC para melhorar seu desempenho na camada de realce.

6.2.1.1 Número de coeficientes e *trailing ones*

Ao codificar os índices dos *cosets* temos a informação adicional da quantidade máxima de *cosets* diferentes de zero dentro de um bloco de 4×4 pixels. Em outras palavras, quando $M \neq 1$ durante a escolha dos parâmetros. Vamos nomear a quantidade máxima de *cosets* de max_C . Idealmente max_C pode variar de 0 a 16. A partir do valor de max_C , as três tabelas de códigos de tamanho variável, utilizadas para codificar o número de coeficientes diferentes de zero e *trailing ones*, podem ser aperfeiçoadas. Podemos criar três grupos de tabelas. O primeiro grupo, por exemplo, deve retornar códigos pequenos para valores menores, similar a primeira tabela do CAVLC padrão. Cada grupo deve possuir diferentes tabelas com a quantidade de entradas decrescente, logo com códigos menores. A quantidade de entradas é determinada por max_C .

Como os parâmetros de codificação são selecionados por *BF*, em um bloco 4×4 podemos

criar 7 tabelas diferentes em cada grupo. Na prática, não é necessário criar todas as tabelas possíveis, já que o ganho é limitado. Logo, foram criados três grupos de tabelas de códigos de tamanho variável, onde cada grupo possui 5 tabelas diferentes. As tabelas originais do CAVLC, com 62 entradas, fazem parte de cada grupo, junto de 4 tabelas novas com $max_C \in \{1, 3, 6, 10\}$. Durante a codificação, o grupo de tabelas é selecionado com o mesmo critério que no CAVLC padrão, baseado nos coeficientes diferentes de zero dos blocos vizinhos. Dentro do grupo, cada tabela é selecionada a partir de max_C . Um exemplo de uma tabela de codificação é apresentado na Tabela 6.2. O conjunto completo de tabelas pode ser encontrado no Apêndice II.

Tabela 6.2: Tabela de Codificação (Primeiro Grupo, $1 \leq max_C \leq 3$).

	Trailing Ones			
Num Coeff	0	1	2	3
0	1	-	-	-
1	0001	01	-	-
2	000010	000011	001	-
3	0000001	0000011	00000001	0000010

6.2.1.2 Níveis dos Coeficientes

Como mencionado anteriormente, o CAVLC utiliza 7 tabelas para codificar os níveis, sendo que a codificação começa utilizando a primeira tabela feita para valores próximos a zero. Ao codificar os níveis dos coeficientes diferentes de zero, para a camada Wyner-Ziv temos a informação do módulo M usado para criar um determinado *coset* dentro de uma BF . Um valor alto de M indica que é melhor usar tabelas coeficientes com valores médios ou altos. Portanto, se M é maior que um limiar (10 em nossa implementação), a codificação dos níveis de coeficientes começa a partir da segunda tabela ao invés da primeira. Note que já existe um mecanismo similar no CAVLC padrão. Quando existem mais de 10 coeficientes diferentes de zero e menos que 3 *trailing ones*, a codificação também começa a partir da segunda tabela. O valor do limiar foi determinado após diferentes testes.

6.2.1.3 Quantidade total de zeros

De forma similar à codificação do número de coeficientes e *trailing ones*, ao codificar o número total de zeros temos a informação adicional de max_C . Esta informação também indica o valor máximo do número total de zeros. Assim, podemos criar novas tabelas juntamente com as 15 tabelas originais do CAVLC para codificar o número total de zeros. Além disso, se o número de coeficientes diferentes de zero é igual a max_C , o número total de zeros não precisa ser codificado. Na nossa implementação criamos 4 grupos de tabelas diferentes para serem utilizadas com as tabelas originais. Para a criação das tabelas, os 4 limiares utilizados para o valor máximo do número total de zeros foram 1,2,5 e 9. As tabelas podem ser encontradas no Apêndice II.

Em capítulos anteriores foi apresentada a arquitetura do codificador distribuído com resolução mista, e os mecanismos para escolha dos parâmetros de codificação e estimação da correlação estatística dos sinais. Neste capítulo foi detalhada a codificação da camada Wyner-Ziv ou camada de realce. No Capítulo 7 serão apresentados os testes realizados com o codificador proposto, mostrando seu desempenho e a redução no tempo de codificação.

7 EXPERIMENTOS

Este capítulo apresenta os resultados obtidos nos testes para avaliar o desempenho do codificador de vídeo de complexidade reversa com resolução espacial mista (MR-DVC) e o detalhe das condições dos testes de desempenho e do treinamento prévio feito para determinar o modelo de estimação da correlação estatística dos sinais (ver Capítulo 5). A medida objetiva de distorção utilizada foi a PSNR, apesar de não refletir perfeitamente a qualidade visual subjetiva do sinal, é a mais usada na literatura. A PSNR é calculada pela equação (3.1).

Também foram realizados testes calculando o tempo de codificação, para poder comparar a complexidade (esforço computacional) requerida para codificar uma seqüência de vídeo com um codificador convencional e um codificador Wyner-Ziv, uma vez que o tempo requerido está diretamente relacionado à quantidade de operações realizadas no algoritmo de codificação. Esta metodologia é utilizada em outros trabalhos em DVC [89].

O modo de codificação Wyner-Ziv com resolução mista foi implementado no software KTA do padrão estado-da-arte em codificação de vídeo H.264/AVC [90]. Para as simulações foi usada estimação rápida de movimento [6]. O codificador de entropia CAVLC foi utilizado para a camada base do modo Wyner-Ziv e para as simulações de codificação convencional híbrida. Para a camada de realce foi utilizado o D-CAVLC (ver Capítulo 6). A função de otimização de RD (em inglês *Rate-Distortion Optimization*, ou RDO) foi desativada para o modo de complexidade reversa. O RDO é uma técnica não-normativa do padrão H.264/AVC, que minimiza a função de custo de taxa-distorção permitindo variações no passo de quantização. Esta técnica é usada para selecionar o modo ótimo para cada macrobloco [6]. O RDO é extremamente custoso em termos de esforço computacional, portanto, como o objetivo é codificar a baixa complexidade, não foi utilizada a função de otimização de RD. A janela de busca para a estimação de movimento em todas as simulações foi 16×16 pixels, e foram usados no máximo 2 quadros de referência. O modo direto dos quadros B foi o modo espacial [6].

7.1 SEQÜÊNCIAS DE VÍDEOS PARA TESTES

As seqüências de testes utilizadas neste trabalho fazem parte de um grupo de seqüências de vídeo que são comumente utilizadas para realizar simulações e testes para medir o desempenho e eficiência na codificação de vídeo. Cada seqüência possui 300 quadros capturados a 30 quadros por segundo. As seqüências estão na resolução CIF, no formato $YUV\ 4 : 2 : 0$, o que significa que a componente de luminância Y possui $352 \times 288\ pixels$ e cada uma das componentes de crominância (U e V) possuem $176 \times 144\ pixels$. É importante destacar que as três seqüências recomendadas pela ITU (*International Telecommunication Union*) para testes de compressão de vídeo em resolução CIF: *Foreman*, *Silent* e *Container* [91], fazem parte no grupo selecionado neste trabalho.

Para uma melhor análise dos resultados as seqüências de testes foram agrupadas em três categorias: baixo movimento, movimento médio e alto movimento. Para realizar a classificação se optou por utilizar como critério a fração de blocos codificados com predição *inter*-quadro [92]. Para isso, primeiro devemos realizar a codificação das seqüências utilizando um codec híbrido tradicional, no modo *IPPP*, com passo de quantização (QP) igual a 1 e com restrição de taxa a 4.8 Mbps [92], onde *IPPP* significa que o primeiro quadro é *intra* e todos os outros são quadros do tipo P (ver Seção 3.1). Depois, devemos obter a razão da quantidade de macroblocos que foram codificados utilizando predição *inter*-quadro do total de macroblocos. Note que mesmo o quadro sendo do tipo P o codificador pode optar por codificar um determinado macrobloco com modo *intra*, e que o primeiro quadro do tipo I não deve ser considerado. Assim, quando a fração de blocos *inter* for próxima a 1 (um) o vídeo é classificado como de baixo movimento, já que praticamente 100% dos macrolobocos são preditos de quadros anteriores. Por exemplo, se a fração for próxima a 0.5 o vídeo é classificado como de alto movimento, porque somente 50% dos macroblocos podem ser obtidos com predição temporal.

Na Tabela 7.1 é mostrada a lista das seqüências de vídeos utilizadas neste trabalho, uma breve descrição de cada uma delas e sua classificação. Na Figura 7.1 é mostrado um quadro de cada seqüência.

Tabela 7.1: Sequências de vídeos utilizadas nos experimentos

Vídeo	Movimento	Fração de blocos P	Descrição
<i>Akiyo</i>	Baixo	0.9666	Uma apresentadora de jornal falando
<i>Container</i>	Baixo	0.9246	Um navio de carga se deslocando lentamente
<i>Hall Monitor</i>	Baixo	0.9146	Cena de um corredor com poucas pessoas andando
<i>Silent</i>	Médio	0.8637	Uma pessoa demonstrando linguagem de sinais
<i>Mother-Daughter</i>	Médio	0.8423	Uma mãe e filha falando
<i>Foreman</i>	Alto	0.5947	Um mestre de obras falando
<i>Mobile</i>	Alto	0.5722	Trem de brinquedo em movimento com um calendário atrás dele
<i>Coastguard</i>	Alto	0.5225	Um navio da guarda costeira se deslocando
<i>Soccer</i>	Alto	0.5484	Um treino de futebol

7.2 DESEMPENHO DO D-CAVLC

Conforme indicado no Capítulo 6, neste trabalho foram feitas modificações no codificador CAVLC do padrão H.264/AVC para melhorar seu desempenho na codificação da camada Wyner-Ziv. Para calcular o ganho de taxa foram feitos testes com as 9 seqüências, codificando 299 quadros, utilizando o CAVLC e o D-CAVLC. O percentual no ganho de taxas foi feito pelo método recomendado pela ITU [93]. Na Tabela 7.2 é mostrada a redução na taxa de bits na camada Wyner-Ziv ao utilizar o D-CAVLC. Para estes testes o modo de codificação reversa Wyner-Ziv foi colocada para operar em modo $IbPbP$, onde um b indica um quadro NRWZ codificado tipo B com um quarto da resolução (fator de dizimação 2×2). Como pode ser visto, o ganho é significativo na maioria das seqüências, especialmente nas seqüências de alto movimento.

Tabela 7.2: Ganho da taxa de bits do *bit-stream* da camada Wyner-Ziv

Seqüências								
Movimento Alto				Movimento Médio		Movimento Baixo		
<i>Foreman</i>	<i>Mobile</i>	<i>Soccer</i>	<i>Coastguard</i>	<i>Silent</i>	<i>Mother-Daughter</i>	<i>Hall Monitor</i>	<i>Container</i>	<i>Akiyo</i>
5.91%	3.25%	2.50%	7.01%	2.02%	2.43%	1.98%	2.05%	1.51%



(a)



(b)



(c)

Figura 7.1: Quadros de exemplo das seqüências utilizadas. (a) Baixo Movimento, da esquerda para direita: *Akiyo*, *Container* e *Hall Monitor*. (b) Movimento Médio, da esquerda para direita: *Mother-Daughter* e *Silent*. (c) Alto Movimento, da esquerda para direita: *Coastguard*, *Foreman*, *Mobile* e *Soccer*

7.3 ANÁLISE DE COMPLEXIDADE DE CODIFICAÇÃO

Na arquitetura proposta utilizando resolução mista, a diminuição da complexidade é atingida devido à codificação em baixa resolução dos quadros NRWZ. A quantidade de operações na codificação de um quadro NRWZ deve ser menor que a quantidade de operações de um quadro à resolução original. Para medida de complexidade optamos por utilizar o tempo de codificação. Esta medida é somente um indicativo da complexidade de codificação, já que depende de diversos fatores como a implementação do software, a arquitetura em que o software é executado, do hardware do sistema, do sistema operacional, etc. Espera-se, que o tempo utilizado na estimação

de movimento nos quadros NRWZ seja menor do que o tempo utilizado para quadros com resolução espacial normal. Assim, para realizar uma comparação objetiva de complexidade de codificação foi medido o tempo médio de codificar uma determinada seqüência de vídeo para diferentes valores de QP . Foram considerados 15 valores de QP e 299 quadros por seqüência.

Tabela 7.3: Tempo de codificação do H.264/AVC codec. (EM: Estimação de Movimento)

<i>Modo IBIBI</i>			
	Tempo Total (segundos)	Tempo EM (segundos)	EM / Tempo Total
Foreman	165.25	129.01	78.0%
Coastguard	170.95	132.78	77.6%
Silent	162.52	123.76	76.1%
MÉDIA	166.24	128.52	77.3%
<i>Modo IPBPB</i>			
	Tempo Total (segundos)	Tempo EM (segundos)	EM / Tempo Total
Foreman	303.12	234.03	77.2%
Coastguard	319.61	249.02	77.9%
Silent	268.90	200.89	74.7%
MÉDIA	297.21	227.98	76.7%
<i>Modo IP_nPP_nP</i>			
	Tempo Total (segundos)	Tempo EM (segundos)	EM / Tempo Total
Foreman	254.66	198.26	77.8%
Mother and Daughter	216.01	160.04	74.1%
MÉDIA	235.33	179.15	76.1%

Na Tabela 7.3 é apresentado o tempo de codificação total médio e o tempo utilizado para estimação de movimento do codec H.264 convencional, na implementação KTA, usando um Intel Pentium D 915 Dual Core 2.80 GHz com 1 GB DDR2 de RAM. Para obter dados mais confiáveis o processo de codificação foi feito com prioridade de tempo real, no sistema operacional *Windows XP*. Os modos de codificação utilizados foram o modo IP_nPP_nP , onde P_n indica um quadro tipo P que não é usado como referência, o modo $IPBPB$, onde os quadros B também não são usados como referência, e o modo $IBIB$. Note que a estimação de movimento, mesmo sendo a

estimação rápida [6], é responsável por aproximadamente 76% do tempo total de codificação em todos os casos.

O tempo de codificação para o modo com resolução mista proposto é mostrado na Tabela 7.4. Este modo de codificação foi testado em *IbIbI*, *IbPbP* e *IpPpP*. Os quadros *p* e *b* indicam quadros NRWZ com fator de dizimação 2×2 .

Tabela 7.4: Tempo de codificação do modo com resolução mista no H.264/AVC (EM: Estimação de Movimento)

<i>Modo IbIbI</i>			
	Tempo Total (seg)	Tempo EM (seg)	Redução do Tempo Total comparado com <i>IBIBI</i>
Foreman	63.39	33.43	63.6%
Coastguard	63.08	32.44	63.1%
Silent	58.56	28.80	63.9%
MÉDIA	61.51	31.57	63.2%
<i>Modo IbPbP</i>			
	Tempo Total (seg)	Tempo EM (seg)	Redução do Tempo Total comparado com <i>IBPBP</i>
Foreman	171.65	133.73	43.3%
Coastguard	184.06	142.86	42.4%
Silent	153.53	113.73	42.9%
MÉDIA	169.75	130.10	42.6%
<i>Modo IpPpP</i>			
	Tempo Total (seg)	Tempo EM (seg)	Redução do Tempo Total comparado com <i>IP_nPP_nP</i>
Foreman	162.73	125.90	36.1%
Mother and Daughter	134.75	98.70	37.6%
MÉDIA	148.74	112.30	36.79%

Pode-se observar que o codec em resolução mista utilizando quadros chave *intra* consegue reduzir aproximadamente 63% do tempo de codificação. Quando os quadros chave são do tipo *P*

a redução de complexidade é aproximadamente 42%. No modo I_pPpP a redução é de 36% comparado ao IP_nPP_nP . Para comparar como se pode adaptar a redução da complexidade requerida, na Tabela 7.5 mostra-se o tempo de codificação de uma seqüência utilizando dois quadros NRWZ, no modo $IbbPbbP$. Note que existe uma redução no tempo de codificação comparado com o modo $IbPbP$, e que se comparamos sua codificação à codificação normal $IBBPBBP$ a redução do tempo de codificação é em torno de 66%.

Tabela 7.5: Tempo de codificação do modo com resolução mista no H.264/AVC (EM: Estimação de Movimento, FPS: *frame per second*)

<i>Modo IbbPbbP</i>			
	Tempo Total (seg)	Tempo EM (seg)	Redução do Tempo Total comparado com <i>IBBPBBP</i>
Foreman	117.97	78.65	66.4%
Silent	97.30	58.81	67.1%
MÉDIA	107.64	68.58	66.8%

7.4 ESTIMAÇÃO DA CORRELAÇÃO ESTATÍSTICA

Para obter o modelo de estimação da correlação estatística entre Y e X , é preciso estimar σ_X , σ_Z e ρ . Este processo é realizado mediante as equações (5.4),(5.6) e (5.8), conforme descrito no Capítulo 5. As constantes de cada equação são obtidas utilizando um grupo de quadros de treinamento.

Durante o treinamento foram usados os 15 primeiros quadros de três seqüências diferentes, totalizando 45 quadros. As seqüências utilizadas foram: *Silent*, *Mobile* e *Akiyo*. A seqüência *Silent* é de movimento médio, a seqüência *Mobile* possui um movimento alto, e muita informação de alta frequência, por exemplo, bordas e contornos, enquanto a seqüência *Akiyo* é de baixo movimento. Os 45 quadros de treinamento não fizeram parte dos testes de desempenho na codificação. O grupo de testes, para a verificação do desempenho em termos de taxa-distorção, é formado pelos 300 quadros de cada uma das 6 seqüências restantes, e os 285 quadros finais das seqüências *Silent*, *Mobile* e *Akiyo*.

7.5 DESEMPENHO DO MODO DE CODIFICAÇÃO WYNER-ZIV COM RESOLUÇÃO MISTA

Para os testes do desempenho do codificador Wyner-Ziv com resolução mista, realizamos comparações com o codificador H.264/AVC convencional e com outras arquiteturas de DVC. Para realizar as comparações no desempenho de codificação, as curvas de PSNR foram geradas conforme a recomendação da ITU [91], onde a PSNR é obtida a partir da média da PSNR de cada quadro.

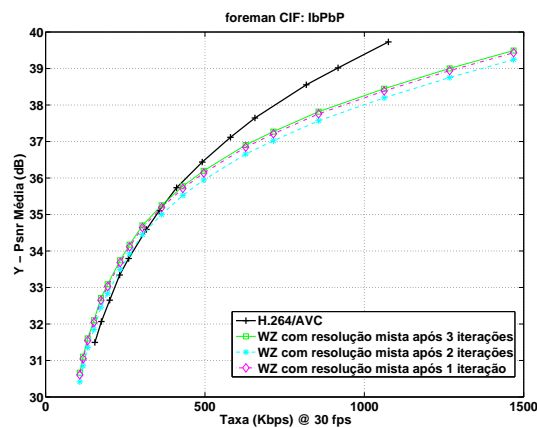
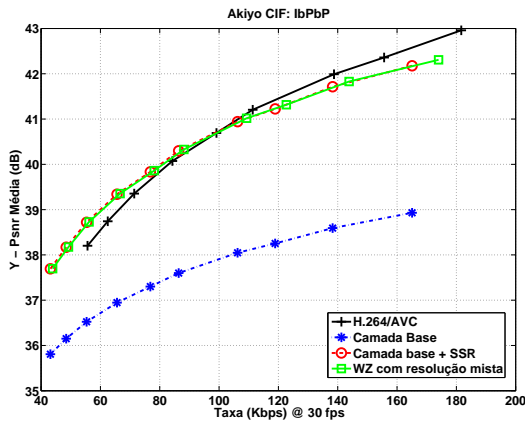


Figura 7.2: Resultados de PSNR média para a componente de luminância da seqüência *Foreman* no modo *IbPbP*. Comparando H.264 convencional no modo *IBPBP* e o modo de codificação Wyner-Ziv depois de 1,2 e 3 iterações.

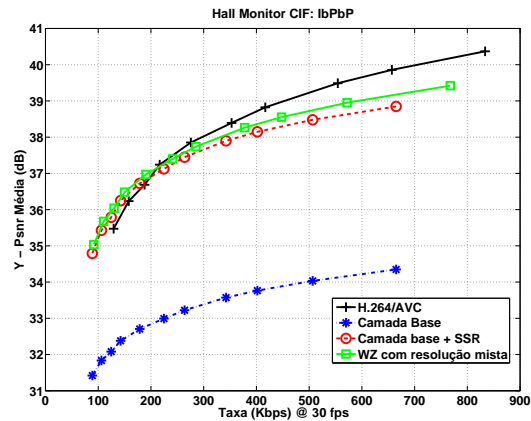
7.5.1 Comparação com codificação convencional híbrida

Na Fig. 7.2 é mostrado o desempenho da codificação para a seqüência *Foreman* de: (i) o H.264/AVC convencional no modo *IBPBP*; (ii) o modo de codificação WZ *IbPbP* utilizando a informação lateral depois de uma, duas e três iterações. A codificação é feita nas componentes de luminância *Y* e crominância *UV* da seqüência, sendo que inicialmente mostramos os resultados onde a PSNR foi calculada considerando somente a componente *Y*. Pode-se observar que o modo WZ é competitivo, já que supera a codificação convencional a baixas taxas. Porém, para altas taxas, a codificação convencional supera o modo WZ. A diferença de PSNR entre iterações é pequena, mas existe uma diferença visual subjetiva perceptível. Para esta seqüência em particular,

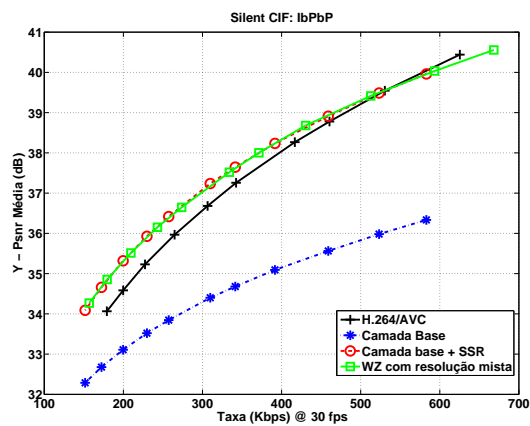
mais iterações não representam um ganho significativo. Em todos os testes o limiar utilizado na geração da informação lateral (ver Seção 4.4) foi $T_i = \{500, 80, 50, 8, 5\}$, onde o índice i representa o número da iteração.



(a)



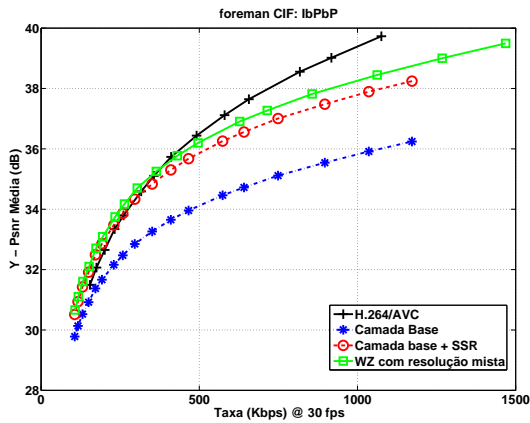
(b)



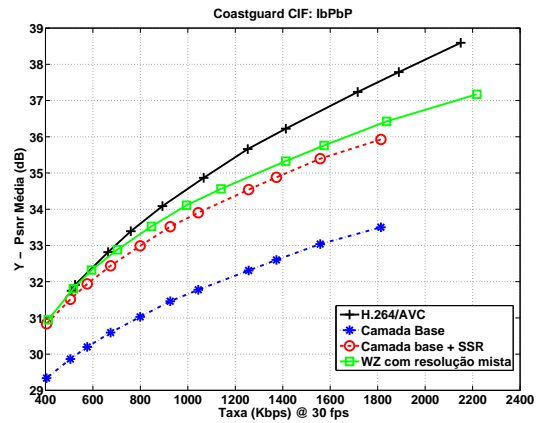
(c)

Figura 7.3: Resultados de PSNR média para a componente de luminância para seqüências de baixo e médio movimento no modo *IbPbP*. Comparando H.264 convencional no modo *IBPBP*, o modo de codificação Wyner-Ziv *IbPbP* depois de 3 iterações, a camada base e os quadros chave junto com os quadros SSR. (a) *Akiyo*, (b) *Hall Monitor* e (c) *Silent*

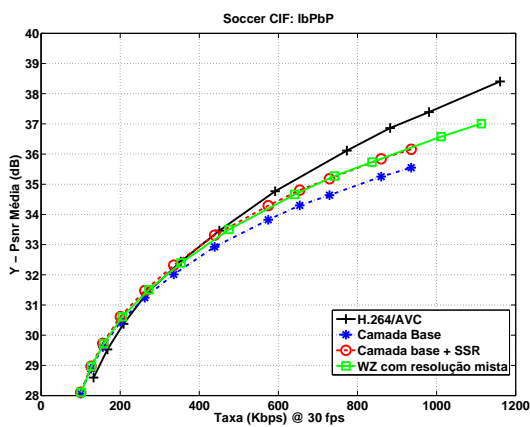
Resultados adicionais para o mesmo modo *IbPbP* são mostrados na Figura 7.3 para seqüências de baixo e médio movimento e na Figura 7.4 para seqüências de alto movimento. Nas Figuras também são mostradas as curvas da camada base de baixa resolução, a qual representa a decodificação a baixa complexidade, e a curva formada a partir dos quadros chave e quadros



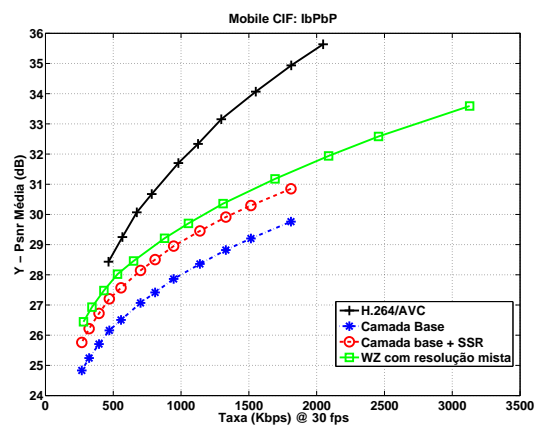
(a)



(b)



(c)



(d)

Figura 7.4: Resultados de PSNR para a componente de luminância para seqüências de alto movimento no modo *IbPbP*. Comparando H.264 convencional no modo *IBPBP*, o modo de codificação Wyner-Ziv depois de 3 iterações, a camada base e os quadros chave junto com os quadros SSR. (a) *Foreman*, (b) *Coastguard*, (c) *Soccer* e (d) *Mobile*

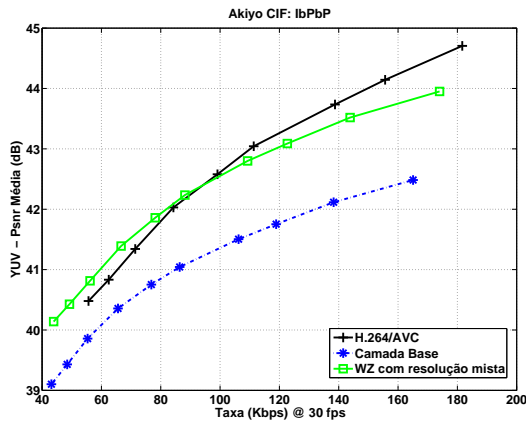
SSR. É importante notar que em alguns casos a curva dos quadros SSR e quadros chave possui grandes variações de PSNR entre quadros, já que os quadros chave e os quadros SSR não necessariamente possuem uma distorção similar. Isto não acontece de forma tão evidente quando a camada WZ é utilizada, já que a mecanismo de escolha de parâmetros de codificação vai tentar gerar uma distorção constante na seqüência completa.

Podemos observar que para as seqüências de baixo e médio movimento o modo WZ com resolução mista consegue ser muito eficiente, superando em alguns casos a codificação

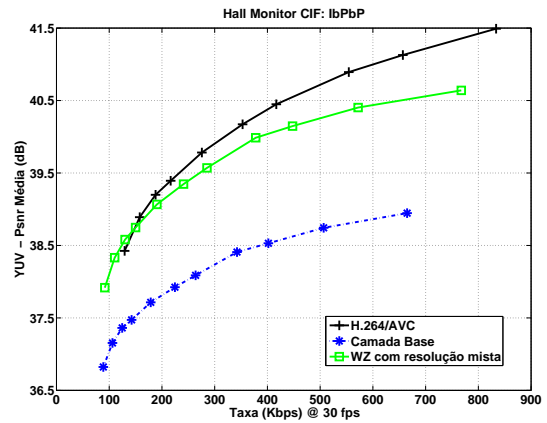
convencional. Isto ocorre principalmente devido à geração adequada da SI. Para as seqüências *Akiyo* e *Silent* o mecanismo de escolha de parâmetros de codificação não envia muitos índices de *cosets* já que a predição indica que não é necessário. Para as seqüências de alto movimento o desempenho é um pouco inferior, especialmente na seqüência *Mobile*. Isto deve-se a que esta seqüência possui uma grande quantidade de informação de alta freqüência e movimento não translacional superior ao que o processo de SSR consegue recuperar. O H.264/AVC possui estimação de movimento com tamanho de blocos adaptativos de tal forma que pode realizar uma melhor estimação de movimento para este tipo de seqüências. Logo, é possível melhorar o resultado do codificador WZ para a seqüência *Mobile* se a geração da informação lateral for implementada com tamanho de blocos variável. Para as seqüências *Foreman*, *Coastguard* e *Soccer*, o modo WZ mesmo tendo o desempenho um pouco inferior, comparado às seqüências de médio e baixo movimento, ainda é competitivo, e consegue superar a codificação híbrida a baixas taxas em algumas seqüências. Os mesmos resultados considerando as componentes de luminância e crominância são mostrados nas Figuras 7.5 e 7.6. Note que as componentes de crominância possuem menor resolução que a componente de luminância e são sinais mais suaves. Assim, a dizimação na crominância não é tão prejudicial como na luminância. Por isso, ao considerar a PSNR de todas as componentes *YUV*, a curva da camada base é mais próxima à curva da codificação convencional.

Resultados para o modo de codificação *IpPpP* são apresentados nas Figuras 7.7 e 7.8 considerando somente a componente de luminância e todas as componentes, respectivamente. A Figura mostra uma seqüência de cada grupo (baixo, médio e alto movimento). Também é possível observar que os resultados para baixo e médio movimento são mais satisfatórios que os resultados de alto movimento. Também, como os quadros *P* convencionais são menos eficientes que os quadros *B*, o modo de codificação com resolução mista no modo *IpPpP* consegue um desempenho aparentemente melhor, ao comparar com a codificação tradicional, do que no modo *IbPbP*. Porém, o modo *IpPpP* é o que apresenta menor redução no tempo de codificação como visto na Seção 7.3.

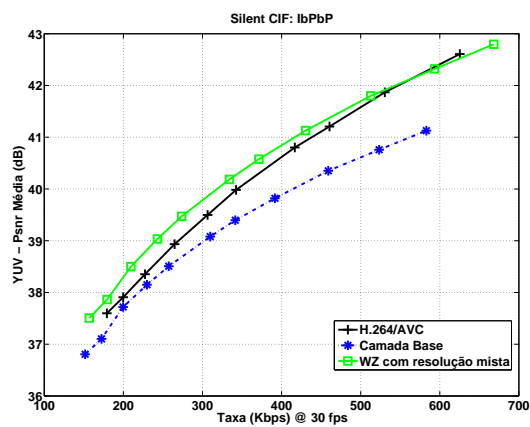
Como mostrado na Seção 7.3, o menor tempo de codificação é obtido utilizando quadros de referência *I*. Nas Figuras 7.9 e 7.10 mostramos resultados do codificador WZ no modo *IbIbI* comparados ao modo *IBIBI* do codec tradicional. Pode-se observar que neste modo



(a)



(b)

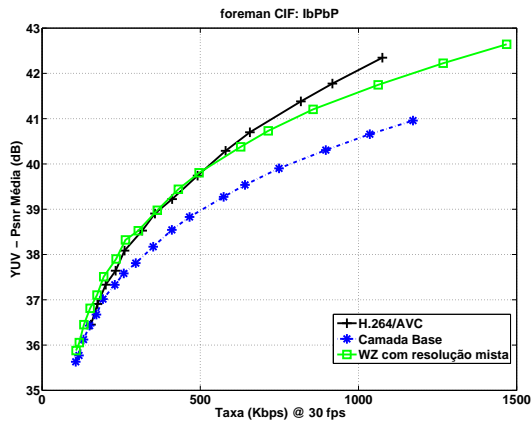


(c)

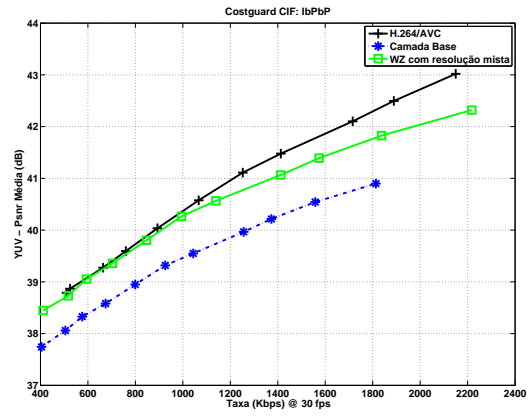
Figura 7.5: Resultados de PSNR para a componente de luminância e crominância para seqüências de baixo e médio movimento no modo *IbPbP*. Comparando H.264 convencional, o modo de codificação Wyner-Ziv depois de 3 iterações e a camada base. (a) *Akiyo*, (b) *Hall Monitor* e (c) *Silent*

o codificador WZ apresenta um resultado inferior aos outros modos, quando comparado com a codificação tradicional. Todavia, neste modo a redução da complexidade supera 60% (ver Tabela 7.4).

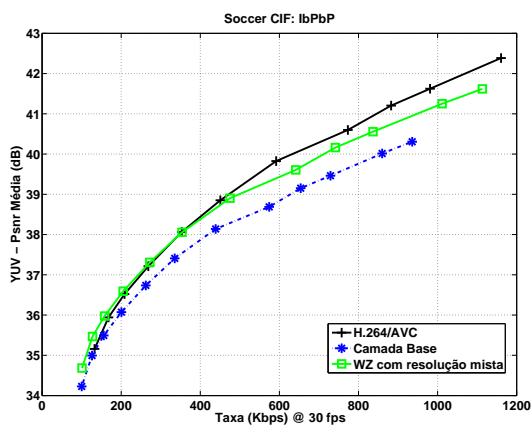
Outra alternativa para reduzir o tempo de codificação é incrementar o número de quadros NRWZ. Porém, é esperado que quanto maior o número de quadros NRWZ, pior será o desempenho do codificador em termos de RD. Isto ocorre porque a geração da informação lateral não é tão eficiente quando não se utiliza quadros contíguos temporalmente. Na Figura



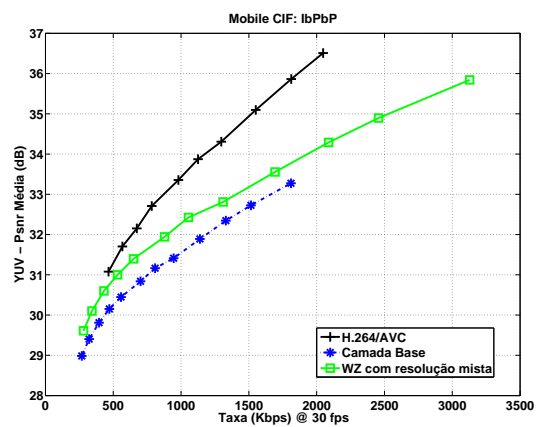
(a)



(b)



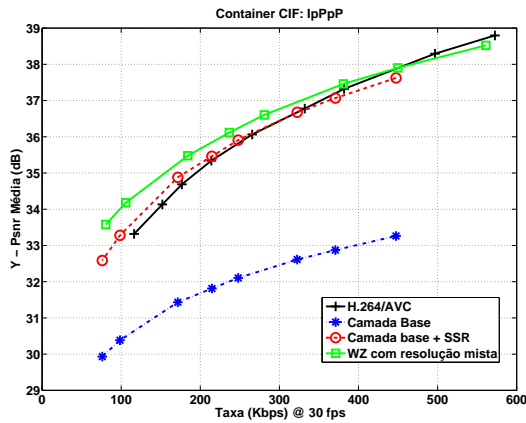
(c)



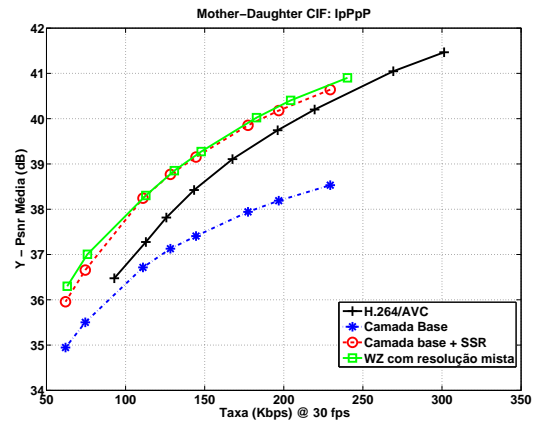
(d)

Figura 7.6: Resultados de PSNR para a componente de luminância e crominância para seqüências de alto movimento no modo IBPBP. Comparando H.264 convencional, o modo de codificação Wyner-Ziv depois de 3 iterações e a camada base. (a) *Foreman*, (b) *Coastguard*, (c) *Soccer* e (d) *Mobile*

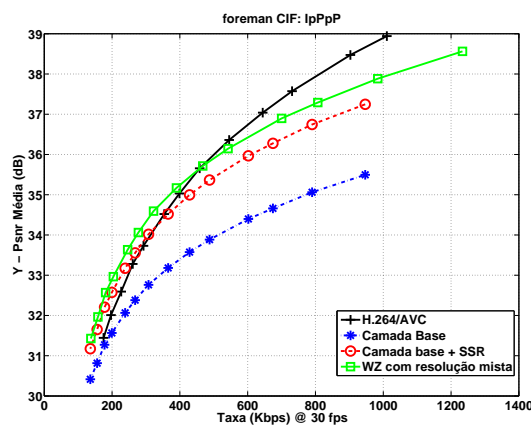
7.11 mostramos o resultados da codificação da seqüência de alto movimento *Foreman* em modo *IbbPbbP* comparado ao modo *IBBPBBP* do H.264 tradicional. Como esperado, o desempenho em termos de RD do codificador WZ é significativamente inferior ao codificador tradicional, mas possui uma queda na complexidade de aproximadamente 66% (ver Seção 7.3). O tempo em codificar uma seqüência *IBBPBBP* em um codificador tradicional é superior ao tempo de codificar a mesma seqüência no modo *IBPBP*, já que os quadros *B* são mais custosos que os quadros *P*. No caso do codificador WZ proposto o modo *IbbPbbP* é mais rápido que o modo



(a)



(b)

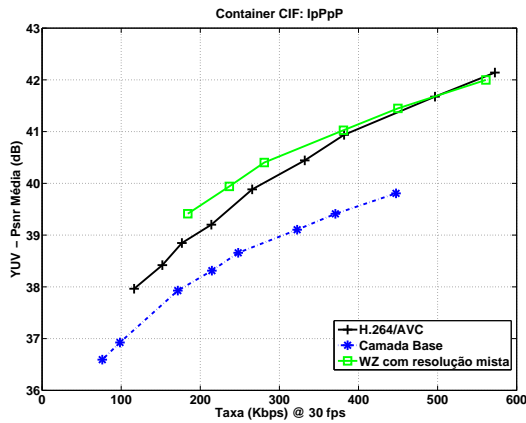


(c)

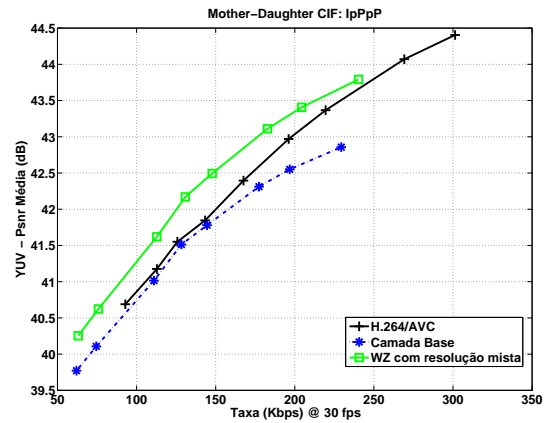
Figura 7.7: Resultados de PSNR para a componente de luminância no modo $IpPpP$. Comparando H.264 convencional, o modo de codificação Wyner-Ziv depois de 3 iterações, a camada base e os quadros chave junto com os quadros SSR. (a) *Container* - baixo movimento, (b) *Mother-Daughter* - movimento médio, (c) *Foreman* - alto movimento

$IbPbP$, já que possui mais quadros NRWZ. Logo, é possível adaptar a complexidade mediante o número de quadros NRWZ.

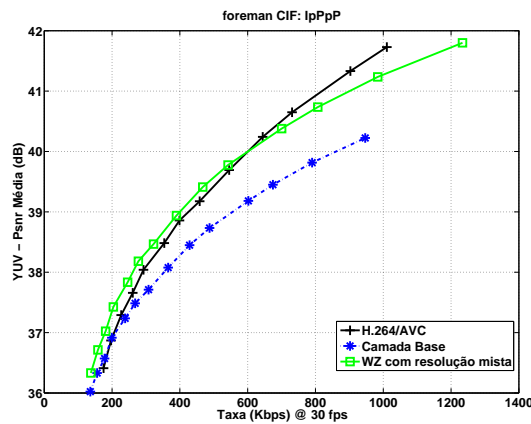
Como foi mencionado anteriormente, a baixas taxas para a maioria das seqüências, o modo de codificação Wyner-Ziv proposto neste trabalho, possui um melhor desempenho em termos de RD do que o H.264/AVC padrão. Isto se deve a correta geração da SI aplicada a um *framework* com resolução mista. A codificação em resolução mista proposta pode ser vista como uma codificação interpolativa. A codificação interpolativa foi proposta para imagens [94] e permite um melhor



(a)



(b)



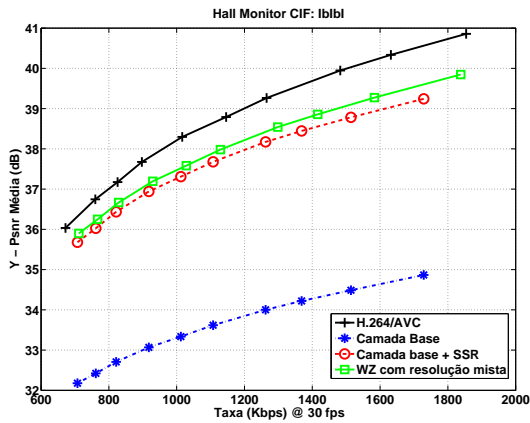
(c)

Figura 7.8: Resultados de PSNR para a componente de luminância e crominância no modo $IpPpP$. Comparando H.264 convencional, o modo de codificação Wyner-Ziv depois de 3 iterações e a camada base. (a) *Container* - baixo movimento, (b) *Mother-Daughter* - movimento médio, (c) *Foreman* - alto movimento

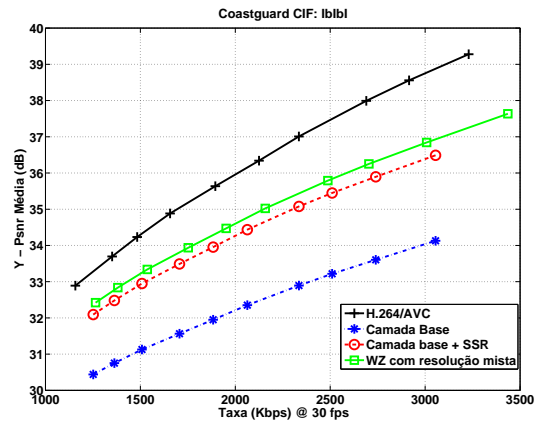
desempenho que a compressão de imagens com resolução original a baixas taxas.

7.5.2 Comparação com outras arquiteturas DVC

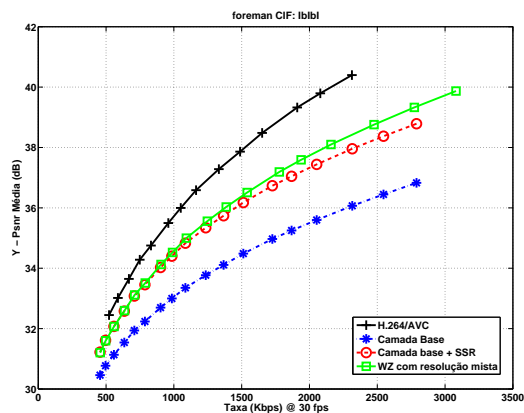
Para uma melhor análise da eficiência do nosso codificador, comparamos o codificador com resolução mista com um dos codecs WZ mais populares na literatura: o codec DISCOVER [67]. O DISCOVER é um codificador distribuído que utiliza a arquitetura de Stanford (ver Seção 3.2.1). Dentro dos codecs com a arquitetura de Stanford, o DISCOVER é um dos codecs que



(a)



(b)

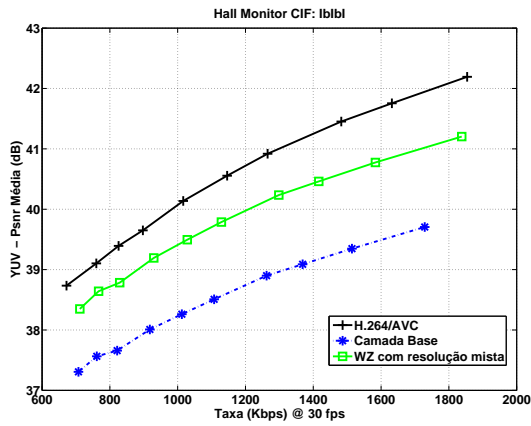


(c)

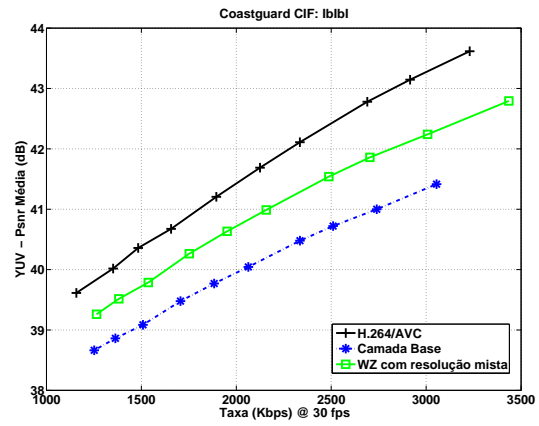
Figura 7.9: Resultados de PSNR para a componente de luminância no modo *Ibbi*. Comparando H.264 convencional no modo *IBIBI*, o modo de codificação Wyner-Ziv depois de 3 iterações e a camada base e os quadros chave junto com os quadros SSR. (a) *Hall Monitor* - baixo movimento, (b) *Coastguard*- alto movimento, (c) *Foreman*- alto movimento

reportam melhor desempenho. O DISCOVER utiliza uma geração de informação lateral mediante interpolação temporal de quadros uni e bi-direcional, refinamento dos vetores de movimento e técnicas de suavização [95, 96]. A codificação de canal do DISCOVER, pode ser feita por *Turbo Codes* ou LDPC, sendo que a implementação com LDPC apresenta um melhor desempenho e foi utilizada nestes testes.

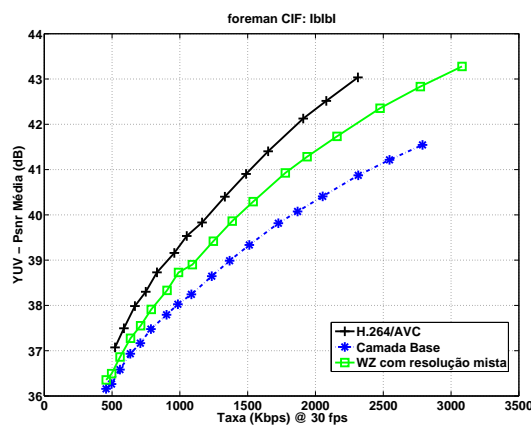
As simulações do DISCOVER foram feitas utilizando estimação de movimento rápida e codificação de entropia CAVLC. Além disso, realizamos testes com o DISCOVER com e sem



(a)



(b)



(c)

Figura 7.10: Resultados de PSNR para a componente de luminância e crominância no modo *IbIbI*. Comparando H.264 convencional no modo *IBIBI*; o modo de codificação Wyner-Ziv depois de 3 iterações; e a camada base. (a) *Hall Monitor* - baixo movimento, (b) *Coastguard*- alto movimento, (c) *Foreman*- alto movimento

RDO. A maioria dos resultados do DISCOVER reportados na literatura utilizam o RDO [67, 95]. O codec DISCOVER somente trabalha com a componente de luminância Y e unicamente permite utilizar quadros chave do tipo *intra*, ou seja, o modo de codificação foi *IZIZI*, onde Z indica um quadro WZ codificado totalmente de forma distribuída. Também é importante notar que o DISCOVER utiliza tabelas de quantização para os quadros Z que são dependentes da seqüência. Em outras palavras, dada uma determinada seqüência, o DISCOVER utiliza uma determinada tabela de quantização. Esta tabela de quantização não é feita para maximizar o desempenho, mas

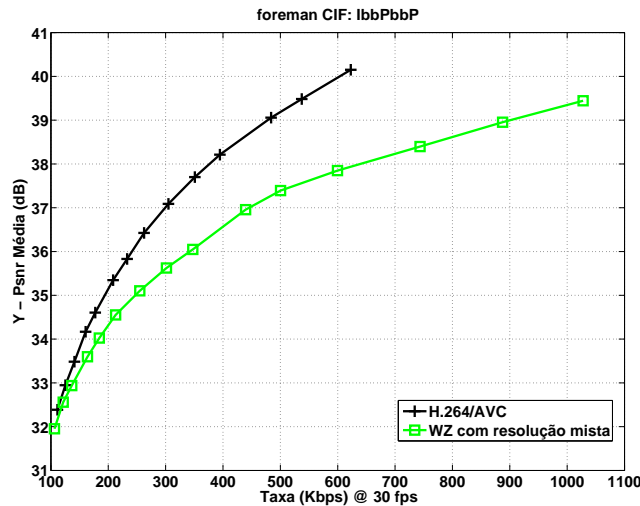


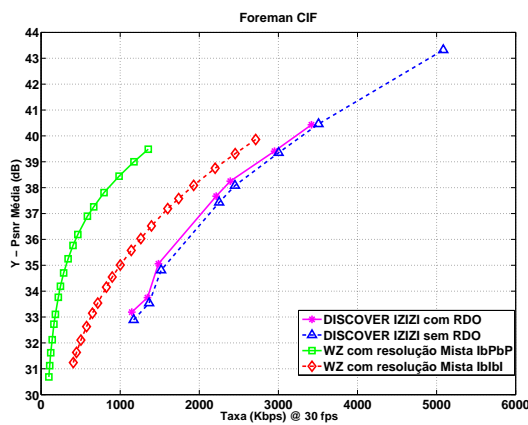
Figura 7.11: Resultados de PSNR para a componente de luminância da seqüência *Foreman* no modo *IbbPbbP*. Comparando H.264 convencional e o modo de codificação Wyner-Ziv.

para obter uma qualidade visual subjetiva uniforme da seqüência decodificada. O DISCOVER também utiliza um canal de retorno. O modo de codificação distribuído proposto neste trabalho, não utiliza canal de retorno e a seleção dos parâmetros de quantização para os quadros NRWZ é feita de forma automática.

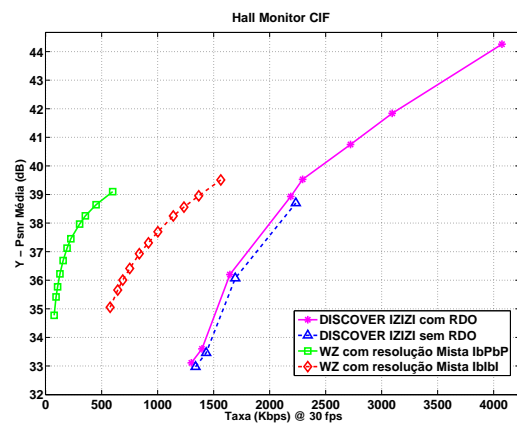
Em termos de complexidade utilizando o RDO, o codec DISCOVER codifica uma seqüência CIF em 165 segundos, em média. Este tempo é comparável ao tempo do codificador proposto nos modos *IbPbP* e *IpPpP* sem RDO. Por outro lado, sem RDO, o DISCOVER pode codificar uma seqüência inteira em aproximadamente 20 segundos. A arquitetura de resolução mista não consegue atingir uma complexidade tão baixa. No caso, de utilizar quadros chave do tipo *intra* (*IbIbI*) o codificador com resolução mista utiliza aproximadamente 60 segundos. Porém, como pode ser visto na Figura 7.12, o modo proposto possui um desempenho em termos de *RD* superior ao DISCOVER. Na Figura 7.12 mostramos a curvas de RD: (i) o codificador com resolução mista no modo *IbPbP* e (ii) no modo *IbIbI*, (iii) o DISCOVER no modo *IZIZI* com RDO e (iv) sem RDO. Note que tanto a PSNR como a taxa são calculadas somente para a componente de luminância.

Nas seqüências *Foreman* e *Hall Monitor* o codificador com resolução mista em ambas as configurações supera o DISCOVER por uma margem significativa. Na seqüência *Foreman* o modo de codificação na configuração *IbIbI* supera o DISCOVER por 1 a 2 dB, e o modo na

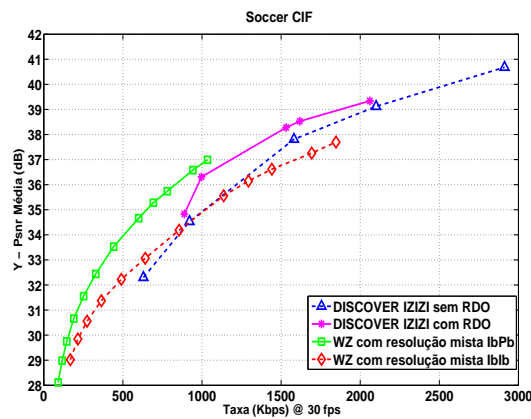
configuração *IbPbP* por mais de 5 dB. Na seqüência *Hall Monitor* a diferença é ainda maior. De todos os testes realizados, o único onde o DISCOVER consegue superar o codificador proposto é na seqüência *Soccer*, onde o DISCOVER supera o modo de resolução mista somente em *IbIbI* para altas taxas. Isto deve se dar porque a estimação dos parâmetros $\{QP_{wz}, M\}$ não foi satisfatória na seqüência *Soccer* no modo *IbIbI*, enquanto o DISCOVER utiliza um canal de retorno. Contudo, o modo *IbPbP* tem um melhor desempenho que o DISCOVER também nesta seqüência.



(a)



(b)



(c)

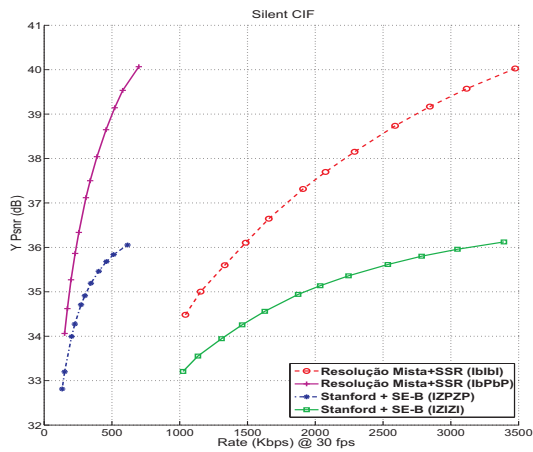
Figura 7.12: Curvas de PSNR para a componente *Y*. Comparando o modo de codificação com resolução mista com outra arquitetura de DVC: (a) Foreman CIF; (b) Hall Monitor CIF, (c) Soccer CIF

Como já foi mencionado, o DISCOVER somente opera com quadros chave *intra*. Para verificar o potencial da arquitetura de Stanford com quadros chave *inter*, implementamos uma

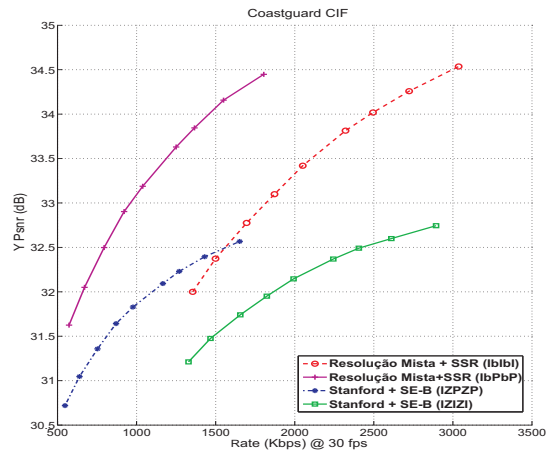
geração de informação temporal muito utilizada nesta arquitetura: algoritmo SE-B [65, 66]. Neste algoritmo, a idéia básica é realizar estimação de movimento entre os dois quadros chave e gerar o quadro interpolado. Inicialmente a estimação de movimento é feita utilizando o quadro anterior como fonte e o posterior como referência obtendo os vetores de movimento MV_F . Depois, um novo processo de estimação de movimento é feito agora com o quadro posterior como fonte e o anterior como referência, gerando MV_B . O quadro intermediário é gerado a partir de uma média entre os quadros compensados utilizando $MV_F/2$ e $MV_B/2$. Basicamente, este método assume que o movimento do quadro que está sendo interpolado é a metade entre o movimento existente entre os quadros chave anterior e posterior. A interpolação SE-B é uma das mais usadas na arquitetura de Stanford, e é usada como referência para comparar geração de informação lateral em outros trabalhos [30, 31, 32]. Realizamos, então, testes de geração de informação lateral comparando o SE-B e o algoritmo de semi super resolução (SSR), assumindo que uma melhor informação lateral possibilita um melhor desempenho do codificador Wyner-Ziv.

Na Figura 7.13, é mostrada a comparação entre o desempenho dos processos de geração de SI. Estas curvas de PSNR foram calculadas a partir da MSE da seqüência inteira. Isto se deve ao fato que seqüências formadas pelos quadros chaves e quadros da geração de SI (SE-B ou SSR) possuem grandes variações de PSNR. Os quadros chaves possuem uma PSNR significativamente melhor que os quadros de SI. Logo, utilizar a PSNR média para este caso particular não é um bom indicativo da qualidade subjetiva do sinal. Nos gráficos são mostrados: (i) os quadros chave decodificados e os quadros resultantes do processo de SSR após uma iteração, para o codificador com resolução mista no modo *IbIbI*; (ii) os quadros chave decodificados e os quadros resultantes do processo de SSR após uma iteração, para o codificador com resolução mista no modo *IbPbP*; (iii) os quadros chave tipo *I* e os quadros interpolados mediante o método SE-B para a arquitetura de Stanford; e (iv) os quadros chave tipo *P* e os quadros interpolados mediante o método SE-B para a arquitetura de Stanford. Foram utilizadas as seqüências de alto movimento *Coastguard*, *Foreman*, *Mobile*, e a seqüência de movimento médio *Silent*.

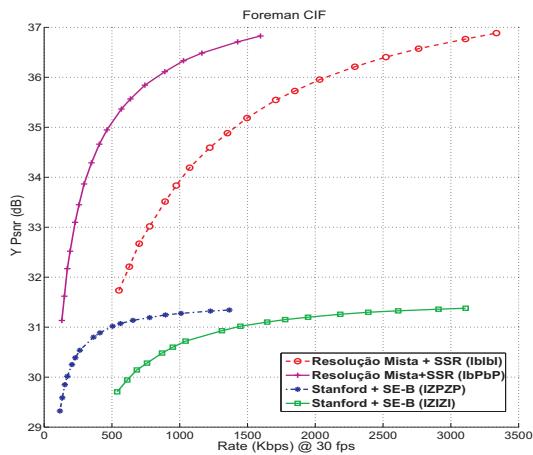
Como era esperado, o nosso método possui um desempenho superior à geração de informação lateral SE-B. Novos métodos de geração de informação lateral para a arquitetura de Stanford podem levar a um melhor desempenho do codificador Wyner-Ziv [30, 31, 32, 97]. Porém, como nosso *framework* trabalha com uma versão em baixa resolução do quadro, a nossa arquitetura



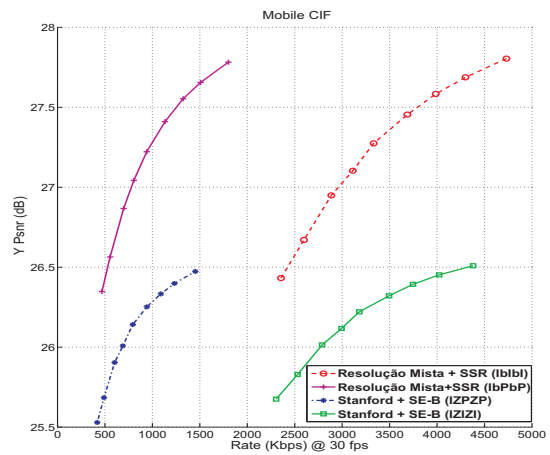
(a)



(b)



(a)



(b)

Figura 7.13: Resultados do processo de semi super resolução para quatro seqüências diferentes. Comparando a semi super resolução para quadros b usando quadros-chave tipo I e tipo P , e a interpolação SE-B para os mesmos quadros chave.

tem o potencial de superar qualquer método de geração de informação lateral da arquitetura de Stanford. Uma melhor informação lateral implica um melhor desempenho de RD do codec [66]. Por outro lado, como foi discutido anteriormente, a arquitetura de Stanford consegue codificar seqüências de vídeo a um custo computacional menor.

No próximo capítulo serão apresentadas as conclusões gerais do trabalho, detalhando as contribuições realizadas na presente pesquisa e as perspectivas para trabalhos futuros.

8 CONCLUSÕES

O trabalho apresentado consiste em um modo de codificação de vídeo com complexidade reversa, ideal para cenários onde a codificação de vídeo é feita por dispositivos móveis com restrição de energia. A codificação com complexidade reversa é baseada na codificação distribuída de vídeo. A redução de complexidade, no processo de codificação, é alcançada mediante a codificação de quadros em baixa resolução, formando a camada base. A camada de realce é gerada a partir de um quadro residual entre o quadro reconstruído interpolado e o quadro original. Os coeficientes transformados do quadro residual são codificados mediante uma técnica distribuída utilizando índices de *cosets* sem memória, sem o uso de um canal de retorno.

O modo de codificação possui resolução mista, já que os quadros-chave são codificados em uma resolução espacial normal. Este modo pode ser utilizado opcionalmente, permitindo que o codificador de vídeo não opere sempre no modo de complexidade reversa. Este modo pode ser ligado somente quando necessário, por exemplo, se o dispositivo móvel está com pouca bateria. O uso de quadros com resolução mista faz com que a codificação Wyner-Ziv possua um desempenho competitivo, em termos de taxa-distorção. A redução no esforço computacional, que implica uma perda de desempenho, pode ser feita de forma adaptável mudando a quantidade de quadros em baixa resolução e o fator de dizimação.

O único canal de comunicação entre o decodificador e codificador é o sinal de vídeo codificado em duas camadas. Não existe canal de retorno nem informação auxiliar (*hash*). Logo, o decodificador pode reproduzir a seqüência de vídeo imediatamente após esta ser recebida ou armazenar a seqüência para reproduzi-la em qualquer outro momento. A arquitetura também permite que o decodificador consiga reproduzir uma seqüência com baixa complexidade de decodificação sem perda de quadros, simplesmente interpolando os quadros da camada base.

Foi feito um estudo de complexidade, baseado no tempo de codificação, para verificar a redução do esforço computacional que a arquitetura proposta possibilita. Foi mostrado que a codificação com resolução mista pode obter uma complexidade similar com um desempenho superior à arquitetura de codificação Wyner-Ziv de Stanford [17, 67], usando quadros-chave

com codificação do tipo *inter*. Por outro lado, é improvável que consigamos obter uma redução de complexidade tão drástica como a obtida pela arquitetura de Stanford com quadros chave do tipo *intra*, já que na codificação com resolução mista a estimação de movimento é sempre necessária. Porém, o uso de codificadores de vídeo em hardware, ou o aumento de capacidade dos dispositivos móveis, pode fazer que uma drástica redução de complexidade não seja necessária. Assim, a arquitetura desenvolvida permite uma melhor adaptabilidade da redução de complexidade em troca do desempenho de codificação.

Para se atingir um bom desempenho de taxa-distorção foi apresentado um novo método de geração de informação lateral que usa um fator de confiabilidade (Seção 4.4) para controlar a adição de alta frequência. Este método possibilita que o nosso codificador com resolução mista consiga alcançar um desempenho até mesmo superior à codificação convencional para seqüências de baixo conteúdo de alta frequência e baixo movimento. Em outras palavras, o método de geração de informação lateral consegue recuperar satisfatoriamente perdas moderadas de informação de alta frequência. A geração da informação lateral pode ser feita de forma iterativa melhorando a sua qualidade visual, mas a um custo de maior complexidade de decodificação, já que cada iteração é um novo processo de decodificação.

8.1 APRESENTAÇÃO DAS CONTRIBUIÇÕES DO DOUTORADO

A concepção e implementação do *framework* no padrão H.264/AVC, é uma das principais contribuições desta tese. Este *framework* permite codificação em duas camadas com redução de complexidade. A decodificação também possui um modo de baixa complexidade, e quando a camada de realce é codificada, o codec é competitivo ao ser comparado com a codificação convencional híbrida. Foi mostrado, por meio dos experimentos, que utilizando um modo de operação com resolução mista é possível que a codificação distribuída supere a codificação convencional, embora em um caso restrito a baixas taxas, e principalmente caso a seqüência de vídeo possua baixo ou médio movimento. O modo de operação com resolução mista pode reduzir as redundâncias temporais de uma seqüência, podendo aproveitar o conteúdo de alta frequência de alguns quadros, os quadros-chave, para reutilizar em outros, os quadros Wyner-Ziv. Nos

experimentos, a codificação Wyner-Ziv foi comparada com a codificação convencional operando nas mesmas condições, sem restrições, diferentemente de comparações com a codificação tradicional restringindo o uso somente de vetores de movimento nulos (iguais a zero), o que é mais comumente encontrado na literatura para codificadores Wyner-Ziv implementados no H.264/AVC [67, 95].

Outra contribuição importante deste trabalho é o método de estimação da correlação estatística dos sinais. Foi realizada uma busca na literatura e parece não haver um método similar que tenha sido apresentado anteriormente, já que a grande maioria das arquiteturas de codificação distribuída utiliza um canal de retorno. Alguns estudos apresentaram um mecanismo para a alocação de taxa da arquitetura de Stanford sem utilizar canal de retorno [98, 99, 100]. Porém, normalmente uma simples diferença nos valores dos pixels entre o quadro original e o quadro de referência é utilizada para prever a probabilidade de erro nos planos de bits [98, 99], obtendo um desempenho similar ao utilizar o canal de retorno somente em algumas seqüências de baixo movimento. O método de estimação da correlação estatística proposto neste trabalho é mais robusto. O método de estimação da correlação é usado para a escolha dos parâmetros de codificação. Logo, ele é responsável pela alocação automática da taxa. No caso de se obter uma perfeita estimação, os parâmetros escolhidos para o código de canal mediante *cosets* sem memória serão ótimos para o modelo estatístico proposto. Assim, nosso método apresenta resultados satisfatórios para seqüências de baixo e médio movimento e inclusive para algumas seqüências de alto movimento. A arquitetura PRISM [61], que não utiliza canal de retorno, assume que a correlação entre os sinais e o ruído Wyner-Ziv é conhecida tanto no codificador e decodificador. Nosso método pode ser aplicado, com algumas modificações, na arquitetura PRISM. Como arquitetura PRISM não é separada em camadas, logo é necessário substituir a taxa de codificação da camada base por outro parâmetro, como por exemplo, a taxa utilizada em blocos vizinhos, para realizar a estimação. Também será necessário fazer um novo treinamento.

Como foi mencionado, na Seção 3.3, a codificação distribuída não oferece normalmente um controle de taxa ou distorção. Porém, nosso mecanismo de estimação de correlação aplicada na escolha dos parâmetros ótimos, também permite obter um valor esperado da distorção e taxa, eliminando a incerteza do ponto de operação da codificação distribuída. Com os valores estimados de taxa e distorção é possível elaborar funções de controle e/ou restrição de taxa no

codificador distribuído, de forma similar aos mecanismos de restrição de taxa na codificação híbrida convencional [6].

Foi também apresentado um novo modelo para o sinal de informação lateral, onde é considerado um fator de atenuação. Este modelo pode ser visto como uma generalização do modelo clássico, onde a informação lateral é modelada como o sinal original adicionado de um sinal de ruído. No modelo proposto a informação lateral é descrita pela adição do sinal de ruído e o sinal original escalonado pelo fator de atenuação.

Para a geração da informação lateral foi implementado um método chamado de semi super resolução, onde quadros em alta resolução são usados para aumentar a resolução espacial de um quadro em baixa resolução. Como mostrado em outros trabalhos de super resolução utilizando técnicas similares [83, 101], o método de semi super resolução pode ser utilizado em diferentes cenários, que não necessariamente estejam relacionados com a codificação distribuída. O método proposto neste trabalho possui um fator de confiabilidade e a possibilidade de utilizar várias iterações, com o objetivo de melhorar a qualidade de quadro resultante do processo de semi super resolução.

Finalmente, também modificamos um codificador de entropia, inicialmente otimizado para trabalhar com coeficientes residuais, para codificar índices de *cosets*. Estas modificações podem ser utilizadas em outros codificadores que utilizem *cosets* ou síndrome.

Uma lista das publicações obtidas neste trabalho, e outras contribuições, pode ser encontrada no Apêndice III.

8.2 PERSPECTIVAS PARA TRABALHOS FUTUROS

Estudos futuros podem melhorar ou estender a pesquisa realizada nas seguintes áreas:

- Estender o trabalho para que possa ser usado em caso de codificação com vistas múltiplas (em inglês *multi-view systems*). Em outras palavras, quando uma seqüência de vídeo é gravada simultaneamente por várias câmeras em posições diferentes, com objetivo de obter uma versão tridimensional do sinal, ou do arranjo ótimo das câmeras. Em sistemas com

vistas múltiplas as principais exigências são: (i) codificação a baixa complexidade, já que os dispositivos devem normalmente transmitir em tempo real, (ii) resistência a erros de transmissão, já que a codificação requer uma comunicação confiável entre todos os dispositivos, e (iii) alta taxa de compressão, por existir limitação de banda no canal de transmissão. Assim, este é outro cenário onde a codificação distribuída pode ser aplicada. Inclusive já existem trabalhos onde uma arquitetura similar à arquitetura de Stanford é aplicada a um sistema de vistas múltiplas [102].

- Implementar um mecanismo automático de alocação de recursos, o qual pode variar a quantidade de quadros Wyner-Ziv e o fator de dizimação de forma dinâmica durante a codificação de uma seqüência de vídeo a medida que a bateria, ou outros recursos, diminui.
- Melhorar a função de semi super resolução utilizando blocos de tamanho variável, e combinando a técnica com técnicas Bayesianas de super resolução.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] PRATT, W. K. *Digital Image Processing*. 3rd. ed. : Willey, 2001.
- [2] COTE, G. et al. H.263+: Video coding at low bit-rates. *IEEE Trans. Circuits Syst. Video Technology*, v. 8, n. 7, pp. 849–866, Novembro 1998.
- [3] HASKELL, B.; PURI, A.; NETRAVALI, A. *Digital Video: An Introduction to MPEG-2*. : Chapman & Hall, 1996.
- [4] WALSH, A.; BOURGES-SéVENIER, M. *MPEG-4 Jump Start*. : Prentice-Hall, 2002.
- [5] PEREIRA, F.; EBRAHIMI, T. *The MPEG-4 Book*. : IMSC Press, 2002.
- [6] WEIGAND, 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, 2003.
- [7] SCHWARZ, H.; MARPE, D.; WIEGAND, T. SNR-scalable extension of H.264/AVC. *In Proc of IEEE International Conf on Image Processing*, pp. 3113–3116, 2004.
- [8] SCHWARZ, H.; MARPE, D.; WIEGAND, T. MCTF and scalability extension of H.264/AVC. *Picture Coding Symposium*, San Francisco, USA, Dezembro 2004.
- [9] SCHWARZ, H.; MARPE, D.; WIEGAND, T. Combined scalability support for the scalable extension of H.264/AVC. *In Proc. of IEEE International Conf on Multimedia and Expo.*, Amsterdam, The Netherlands, Julho 2005.
- [10] DOMANSKI, M.; BLASZAK, L.; MACKOWIAK, S. AVC video coders with spatial and temporal scalability. *Picture Coding Symposium*, Saint-Malo, France, Abril 2003.
- [11] YAN, R. et al. Efficient video coding with hybrid spatial and fine-grain SNR scalabilities. *In Proc of SPIE Visual Communications and Image Processing*, v. 4671, pp. 850–859, 2002.
- [12] A.LEONTARIS; COSMAN, P. Drift-resistant SNR scalable video coding. *IEEE Trans on Image Processing*, v. 15, n. 8, pp. 2191–2197, Agosto 2006.

- [13] COOK, G. et al. Rate-distortion analysis of motion-compensated rate scalable video. *IEEE Trans on Image Processing*, v. 15, n. 8, pp. 2170–2190, Agosto 2006.
- [14] SLEPIAN, J.; WOLF, J. Noiseless coding of correlated information sources. *IEEE Trans on Information Theory*, v. 19, n. 4, pp. 471–480, Julho 1973.
- [15] WYNER, A.; ZIV, J. The rate-distortion function for source coding with side information at the decoder. *IEEE Trans on Information Theory*, v. 2, n. 1, pp. 1–10, Janeiro 1976.
- [16] PRADHAN, S. S.; RAMCHANDRAN, K. Distributed source coding using syndromes (DISCUS): design and construction. in *Proc. IEEE Data Compression Conf.*, pp. 158–167, 1999.
- [17] AARON, A.; ZHANG, R.; GIROD, B. Transform-domain Wyner-Ziv codec for video. *Proc. SPIE Visual Communications and Image Processing*, v. 5308, pp. 520–528, San Jose, Janeiro 2004.
- [18] PURI, R.; RAMCHANDRAN, K. PRISM: A new robust video coding architecture based on distributed compression principles. *Allerton Conference on Communications, Control and Computing*, 2002.
- [19] XU, Q.; XIONG, Z. Layered Wyner-Ziv video coding. *Proc. SPIE Visual Communications and Image Processing*, San Jose, Janeiro 2004.
- [20] XU, Q.; XIONG, Z. Layered Wyner-Ziv video coding. *IEEE Trans on Image Processing*, v. 15, n. 12, pp. 3791–3809, Dezembro 2006.
- [21] WANG, H.; CHEUNG, N. M.; ORTEGA, A. A framework for adaptive scalable video coding using Wyner-Ziv techniques. *EURASIP Journal on Applied Signal Processing*, pp. 1–18, 2006.
- [22] TAGLIASACCHI, M.; MAJUMDAR, A.; RAMCHANDRAN, K. A distributed-source-coding based robust spatio-temporal scalable video codec. *Picture Coding Symposium, USA*, San Francisco, Dezembro 2004.

- [23] WANG, X.; ORCHARD, M. T. Design of trellis codes for source coding with side information at the decoder. *Proceedings of IEEE Data Compression Conference*, pp. 361–370, 2001.
- [24] AARON, A.; GIROD, B. Compression with side information using turbo codes. *Proceedings of IEEE Data Compression Conference*, pp. 252–261, 2002.
- [25] GIROD, B. et al. Distributed video coding. *Proceedings of the IEEE*, v. 93, n. 1, pp. 71–83, Janeiro 2005.
- [26] SEHGAL, A. et al. Scalable video coding using Wyner-Ziv codes. *Picture Coding Symposium*, Dezembro 2004.
- [27] PEIXOTO, E.; QUEIROZ, R. L.; MUKHERJEE, D. Mobile video communications using a wyner-ziv transcoder. *Proc. of SPIE Visual Communications and Image Processing*, San Jose, 2007.
- [28] PEIXOTO, E. *Transcodificador de vídeo Wyner-Ziv/H.263 para comunicação entre dispositivos móveis*. : Dissertação de Mestrado em Engenharia Elétrica com ênfase em Telecomunicações, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 91p., 2008.
- [29] AARON, A. M. et al. Wyner-Ziv coding for video: applications to compression and error resilience. *Proceedings of the IEEE Data Compression Conference*, pp. 93–102, 2003.
- [30] ARTIGAS, X.; TORRES, L. Iterative generation of motion-compensated side information for distributed video coding. in *Proc. IEEE International Conference on Image Processing*, v. 1, pp. I-833–6, 2005.
- [31] ASCENSO, J.; BRITES, C.; PEREIRA, F. Motion compensated refinement for low complexity pixel based distributed video coding. in *Proc. of IEEE Conference on Advanced Video and Signal Based Surveillance*, pp. 593–598, Setembro 2005.
- [32] WEERAKKODY, W. et al. An iterative refinement technique for side information generation in DVC. in *Proc. of IEEE International Conference on Multimedia and Expo*, pp. 164–167, Julho 2007.

- [33] AARON, A.; RANE, S.; GIROD, B. Wyner-ziv video coding with hash-based motion compensation at the receiver. *in Proc. IEEE International Conference on Image Processing*, Outubro 2004.
- [34] MARTINIAN, E. et al. Hybrid distributed video coding using SCA codes. *IEEE Workshop on Multimedia Signal Processing (MMSP)*, pp. 258–261, Outubro 2006.
- [35] COVER, T. M.; THOMAS, J. A. *Elements of Information Theory*. 2nd. ed. : Wiley, 2006.
- [36] PAPOULIS, A.; PILLAI, S. U. *Probability, Random Variables and Stochastic Processes*. 4th. ed. : Mc Grall Hill, 2002.
- [37] SHANNON, C. E. A mathematical theory of communication. *Bell System Technical Journal*, v. 27, n. 17, pp. 379–423,623–656, July and October 1948.
- [38] JAYANT, N.; NOLL, P. *Digital Coding of Waveforms*. : Prentice-Hall, 1984.
- [39] MACKAY, D. *Information Theory, Inference and Learning Algorithms*. : Cambridge University Press, 2003.
- [40] LIN, S.; COSTELLO, D. J. *Error Control Coding*. 2nd. ed. : Prentice-Hall, 2004.
- [41] PROAKIS, J. G. *Digital Communications*. : McGraw-Hill, 1983.
- [42] PRADHAN, S.; CHOU, J.; RAMCHANDRAN, K. Duality between source coding and channel coding and its extension to the side information case. *IEEE Transactions on Information Theory*, v. 14, n. 5, pp. 1181–1203, 2003.
- [43] ZAMIR, R. The rate loss in the wyner-ziv problem. *IEEE Transactions on Information Theory*, v. 42, n. 6, pp. 2073–2084, 1996.
- [44] RICHARDSON, I. E. G. *H.264 and MPEG-4 video compression*. : Wiley, 2003.
- [45] RICHARDSON, I. E. G. *Video Codec Design*. : Wiley, 2002.
- [46] FONSECA, T. A. da. *Redução de Complexidade na Compressão de Vídeo de Alta Resolução*. : Dissertação de Mestrado em Engenharia Elétrica com ênfase em Telecomunicações, Publicação PPGENE.DM 323/08, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 78p., 2008.

- [47] FONSECA, Y. L. T. A.; QUEIROZ, R. L. Open-loop prediction in h.264/avc for high definition sequences. *Simpósio Brasileiro de Telecomunicações*, Setembro 2007.
- [48] INTERNATIONAL TELECOMMUNICATION UNION. H.261: Video codec for audiovisual services at $p \times 64$ kbits. *ITU-T Recommendation H.261*, Março, 1993.
- [49] INTERNATIONAL TELECOMMUNICATION UNION. H.263: Video coding for low bit rate communication. *ITU-T Recommendation H.263*, 1996.
- [50] INTERNATIONAL ORGANISATION FOR STANDARDISATION. Coding of moving pictures and associated audio for digital storage media at up to about 1,5 mbit/s. *ISO/IEC 11172-1*, 1993.
- [51] INTERNATIONAL ORGANISATION FOR STANDARDISATION; INTERNATIONAL TELECOMMUNICATION UNION. Generic coding of moving pictures and associated audio information: Video. *ITU-T Recommendation H.262, ISO/IEC 13818-2*, 2000.
- [52] INTERNATIONAL ORGANISATION FOR STANDARDISATION. Coding of audiovisual objects - part 2: Visual. *ISO/IEC 14496-2*, 2001.
- [53] JOINT VIDEO TEAM. H.264: Advanced video coding for generic audiovisual services, Coding of audio-visual objects - part 10: Advanced video coding. *ITU-T Recommendation H.264, ISO/IEC 14496-10*, 2003.
- [54] QUEIROZ, R. L. et al. Fringe benefits of the H.264/AVC. *International Telecommunication Symposium*, pp. 208–212, 2006.
- [55] PRADHAN, S.; RAMCHANDRAN, K. Distributed source coding: symmetric rates and applications to sensor networks. *Proc. IEEE Data Compression Conference*, pp. 363–372, 2000.
- [56] PRADHAN, S.; RAMCHANDRAN, K. Group-theoretic construction and analysis of generalized coset codes for symmetric/asymmetric distributed source coding. *Conference Information Sciences and Systems*, NJ, 2000.

- [57] PRADHAN, S.; RAMCHANDRAN, K. Geometric proof of rate-distortion function of gaussian sources with side information at the decoder. *Proc. IEEE International Symposium Information Theory*, pp. 351, 2000.
- [58] PRADHAN, S.; KUSUMA, J.; RAMCHANDRAN, K. Distributed compression in a dense microsensor network. *IEEE Signal Processing Magazine*, v. 19, n. 2, pp. 51–60, 2002.
- [59] AARON, A.; ZHANG, R.; GIROD, B. Wyner-Ziv coding of motion video. In *Proc. Asilomar Conference on Signals and Systems*, Novembro 2002.
- [60] PURI, R.; RAMCHANDRAN, K. PRISM: A “reversed” multimedia coding paradigm. *Proc. IEEE International Conference on Image Processing*, v. 1, pp. I–617–20, Setembro 2003.
- [61] PURI, R.; MAJUMDAR, A.; RAMCHANDRAN, K. PRISM: A video coding paradigm with motion estimation at the decoder. *IEEE Transactions on Image Processing*, v. 16, n. 10, pp. 2436–2448, 2007.
- [62] AARON, A.; SETTON, E.; GIROD, B. Towards practical Wyner-Ziv coding of video. *Proc. IEEE International Conference on Image Processing*, Setembro 2003.
- [63] AARON, A.; VARODAYAN, D.; GIROD, B. Wyner-Ziv residual coding of video. *Proc. International Picture Coding Symposium*, Abril 2006.
- [64] AARON, A.; VARODAYAN, D.; GIROD, B. Rate-adaptive distributed source coding using low-density parity-checks codes. *Proc. Asilomar Conference on Signals, Systems and Computing*, Novembro 2005.
- [65] LI, Z.; DELP, E. J. Wyner-Ziv video side estimator: conventional motion search methods revisited. *Proc. IEEE International Conf on Image Processing*, v. 1, pp. 825–828, Setembro 2005.
- [66] LI, Z.; LIU, L.; DELP, E. J. Rate distortion analysis of motion side estimation in Wyner-Ziv video coding. *IEEE Transactions on Image Processing*, v. 16, n. 1, pp. 98–113, Janeiro 2007.
- [67] ARTIGAS, X. et al. The discover codec: Architecture, techniques and evaluation. *Picture Coding Symposium*, Novembro 2007.

- [68] AARON, A. et al. Systematic lossy forward error protection for video waveforms. *Proc. IEEE International Conference on Image Processing*, pp. I-609–I-612, 2003.
- [69] SEHGAL, A.; AHUJA, N. Robust predictive coding and the Wyner-Ziv problem. *Proc. IEEE Data Compression Conference*, pp. 103–112, 2003.
- [70] RANE, S.; AARON, A.; GIROD, B. Systematic lossy forward error protection for error resilient digital video broadcasting. *Proc. SPIE Visual Communication and Image Processing Conference*, 2004.
- [71] RANE, S.; AARON, A.; GIROD, B. Systematic lossy forward error protection for error resilient digital video broadcasting – a Wyner-Ziv coding approach. *Proc. IEEE International Conference on Image Processing*, Outubro 2004.
- [72] STEINBERG, Y.; MERHAV, N. On successive refinement for the Wyner-Ziv problem. *IEEE Transactions on Information Theory*, v. 50, n. 8, pp. 1636–1654, 2004.
- [73] GALANT, M.; CÔTÉ, G.; KOSSSENTINI, F. An efficient computation-constrained block-based motion estimation algorithm for low bit rate video coding. *IEEE Transactions on Image Processing*, v. 8, Dezembro 1999.
- [74] ZHU, C. A novel hexagon-based search algorithm for fast block motion estimation. *Proc. IEEE International Conference on Acoustic, Speech and Signal Processing*, pp. 1593–1596, Maio 2001.
- [75] MACCHIAVELLO, B.; PEIXOTO, E.; QUEIROZ, R. L. A video coding framework with spatial scalability. *Simpósio Brasileiro de Telecomunicações*, Setembro 2007.
- [76] SCHWARZ, H.; MARPE, D.; WIEGAND, T. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 17, n. 9, pp. 1103–1120, Setembro 2007.
- [77] GONZALEZ R. E. WOODS, S. L. E. R. C. *Digital Image Processing using Matlab*. : Prentice-Hall, 2004.

- [78] SULLIVAN, G.; TOPIWALA, P.; LUTHRA, A. The H.264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions. *Proc. SPIE Conference on Applications of Digital Image Processing*, Agosto 2004.
- [79] MUKHERJEE, D. A robust reversed complexity Wyner-Ziv video codec introducing sign-modulated codes. *HP Labs Technical Report, HPL-2006-80*, Maio 2006.
- [80] BRITES, C.; ASCENSO, J.; PEREIRA, F. Studying temporal correlation noise modeling for pixel based wyner-ziv video coding. *IEEE international conference on Image Processing*, pp. 273–276, Outubro, 2006.
- [81] FREEMAN, W.; JONES, T.; PASZTOR, E. Example-based super-resolution. *IEEE Computer Graphics and Applications*, v. 22, pp. 56–65, 2002.
- [82] BRANDI, F.; QUEIROZ, R. L. de; MUKHERJEE, D. Super resolution of video using key frames. *Proc. of IEEE International Symposium on Circuits and Systems*, Maio 2008.
- [83] BRANDI, F.; QUEIROZ, R. L. de; MUKHERJEE, D. Super resolution of video using key frames and motion estimation. *Proc. of International Conference on Image Processing*, Outubro 2008.
- [84] GARCIA-FRIAS, J.; XIONG, Z. Distributed source and joint source-channel coding: from theory to practice. *Proc. IEEE International Conference on Acoustic, Speech and Signal Processing*, pp. V-1093–V-1095, 2005.
- [85] MUKHERJEE, D. Parameter selection for Wyner-Ziv coding of laplacian sources. *Proc. of SPIE Visual Communications and Image Processing*, San Jose, 2007.
- [86] PRESS, W. H. et al. *Numerical Recipes in C*. 2nd. ed. : Cambridge University Press, 1992.
- [87] SAYOOD, K. *Introduction to Data Compression*. 2nd. ed. : Morgan Kaufmann, 2000.
- [88] OSBORNE, M. J.; RUBENSTEIN, A. *A Course in Game Theory*. : MIT Press, 1994.
- [89] PEREIRA, F. et al. Application scenarios and functionalities for DVC. *Distributed coding for video services - DISCOVER group, Document Deliverable-D19*, Agosto, 2007.

- [90] SUHRING, K. *KTA Software*. <http://iphome.hhi.de/suehring/tml/>, último acceso em 13/06/2008 2008.
- [91] TAN, T.; SULLIVAN, G.; WEDI, T. Recommended simulation common conditions for coding efficiency experiments. *ITU-T Video Coding Experts Group ITU-T SG16 Q.16 Document VCEG-AE010, 31st Meeting, Marocco, Jan 2007*.
- [92] WANG, Y.; CLAYPOOL, M.; KINICKI, R. Impact of reference distance for motion compensation prediction on video quality. *Proceedings of ACM/SPIE Multimedia Computing and Networking*, San Jose, California, 2007.
- [93] BJONTEGAARD, G. Calculation of average psnr differences between RD curves. *ITU-T SC16/Q6, 13th VCEG Meeting Doc. VCEG-M33*, April, 2001.
- [94] ZENG, B.; VENETSANOPOULOS, A. N. A JPEG-based interpolative image coding scheme. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, v. 5, 1993.
- [95] BRITES, C. et al. Evaluating a feedback channel based transform domain Wyner-Ziv video codec. *IEEE Transactions on Image Processing*, v. 16, n. 1, pp. 98–113, 2007.
- [96] ASCENSO, J.; BRITES, C.; PEREIRA, F. Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding. *Proceedings of EURASIP Conference on Speech, Image Processing, Multimedia Communications and Services*, Julho, 2005.
- [97] PEIXOTO, E.; QUEIROZ, R. L.; MUKHERJEE, D. On side information generation for Wyner-Ziv video coding. *Simpósio Brasileiro de Telecomunicações*, 2008.
- [98] MORBEE, M. et al. Rate allocation algorithm for pixel-domain distributed video coding without feedback channel. *Proceedings of the IEEE ICASSP*, pp. 521–524, Hawaii, 2007.
- [99] SHENG, T. et al. Rate allocation for transform domain Wyner-Ziv video coding without feedback. *Proceedings of the 16th ACM international conference on Multimedia*, pp. 701–704, 2008.

- [100] YAACOUB, C.; FARAH, J.; PESQUET-POPESCU, B. Feedback channel suppression in distributed video coding with adaptive rate allocation and quantization for multiuser applications. *EUSASIP Journal on Wireless Communications and Networking*, v. 2008, pp. 1–13, 2008.
- [101] BRANDI, F. *Super-resolução utilizando quadros chave em seqüências de vídeo de resolução mista.* : Dissertação de Mestrado em Engenharia Elétrica com ênfase em Telecomunicações, Publicação PPGENE.DM 362A/09, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 90p., 2009.
- [102] GUO, X. et al. Wyner-Ziv-Based multiview video coding. *IEEE Transactions on circuits and systems for video technology*, v. 18, n. 6, pp. 713–724, 2008.

APÊNDICES

I. CARACTERIZAÇÃO DAS FUNÇÕES DE TAXA E DISTORÇÃO

I.1 CODIFICAÇÃO POR COSETS SEM MEMORIA SEGUIDA DE RECONSTRUÇÃO MSE COM INFORMAÇÃO LATERAL

Assumindo um codificador de entropia ideal para os *cosets*, como mostrado na Figura I.1, a taxa esperada é a entropia da fonte C , a qual pode ser expressa como:

$$\begin{aligned} E(R_{YC}) &= H(C) = - \sum_{c \in \Omega_C} p_c(c) \log_2 p_c(c) \\ &= - \sum_{c \in \Omega_C} \left\{ \sum_{q \in \Omega_Q: \psi(Q,M)=c} \int_{x_l(q)}^{x_h(q)} f_X(x) dx \right\} \times \log_2 \left\{ \sum_{q \in \Omega_Q: \psi(Q,M)=c} \int_{x_l(q)}^{x_h(q)} f_X(x) dx \right\} \end{aligned}$$

A equação acima pode ser re-escrita como:

$$\begin{aligned} E(R_{YC}) &= \\ &= - \sum_{c \in \Omega_C} \left\{ \sum_{q \in \Omega_Q: \psi(Q,M)=c} \left[m_X^{(0)}(x_h(q)) - m_X^{(0)}(x_l(q)) \right] \right\} \\ &\quad \times \log_2 \left\{ \sum_{q \in \Omega_Q: \psi(Q,M)=c} \left[m_X^{(0)}(x_h(q)) - m_X^{(0)}(x_l(q)) \right] \right\} \end{aligned}$$

onde $m_X^{(i)}(x) = \int_{-\infty}^x \nu^i f_X(\nu) d\nu$.

Se a reconstrução for feita pela função do menor erro quadrático médio (MSE) em (4.5), a distorção esperada D_{YC} , dada a informação lateral y e o índice de *coset* c , é dada por:

$$\begin{aligned} E(D_{YC}|Y = y, C = c) &= E([X - \hat{X}_{YC}(y, c)]^2 | Y = y, C = c) \\ &= E(X^2 | Y = y, C = c) - \hat{X}_{YC}(y, c)^2, \end{aligned}$$

onde $\hat{X}_{YC}(y, c) = E(X|Y = y, C = c)$. Marginalizando sobre y e c , temos:

$$E(D_{YC}) = E(X^2) - \int_{-\infty}^{\infty} \left\{ \sum_{c \in \Omega_C} \hat{X}_{YC}(y, c)^2 p_{C|Y}(C = c|Y = y) \right\} f_Y(y) dy.$$

A equação (I.1) pode ser expressa como:

$$E(D_{YC}) = \sigma_X^2 - \int_{-\infty}^{\infty} \left\{ \sum_{c \in \Omega_C} \left(\frac{\sum_{q \in \Omega_Q: \psi(Q, M) = c} \int_{x_l(q)}^{x_h(q)} x f_{X|Y}(x, y) dx}{\sum_{q \in \Omega_Q: \psi(Q, M) = c} \int_{x_l(q)}^{x_h(q)} f_{X|Y}(x, y) dx} \right)^2 \right\} p_{C|Y}(C = c|Y = y) f_Y(y) dy,$$

onde $p_{C|Y}(C = c|Y = y)$ é a função massa de probabilidade condicional de C dado Y , que é

$$p_{C|Y}(C = c|Y = y) = \sum_{q \in \Omega_Q: \psi(Q, M) = c} \int_{x_l(q)}^{x_h(q)} f_{X|Y}(x, y) dx.$$

Logo, temos:

$$E(D_{YC}) = \sigma_X^2 - \int_{-\infty}^{\infty} \left\{ \sum_{c \in \Omega_C} \frac{\left(\sum_{q \in \Omega_Q: \psi(Q, M) = c} \int_{x_l(q)}^{x_h(q)} x f_{X|Y}(x, y) dx \right)^2}{\left(\sum_{q \in \Omega_Q: \psi(Q, M) = c} \int_{x_l(q)}^{x_h(q)} f_{X|Y}(x, y) dx \right)} \right\} \times f_Y(y) dy.$$

Definindo:

$$m_{X|Y}^{(i)}(x, y) = \int_{-\infty}^x \nu^i f_{X|Y}(\nu, y) d\nu,$$

podemos re-escrever $E(D_{YC})$ como:

$$E(D_{YC}) = \sigma_X^2 - \int_{-\infty}^{\infty} \left\{ \sum_{c \in \Omega_C} \frac{\left(\sum_{q \in \Omega_Q: \psi(Q,M)=c} [m_{X|Y}^{(1)}(x_h(q), y) - m_{X|Y}^{(1)}(x_l(q), y)] \right)^2}{\left(\sum_{q \in \Omega_Q: \psi(Q,M)=c} [m_{X|Y}^{(0)}(x_h(q), y) - m_{X|Y}^{(0)}(x_l(q), y)] \right)} \right\} \times f_Y(y) dy.$$

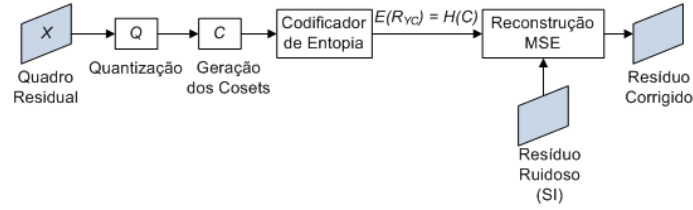


Figura I.1: Codificação de cosets com reconstrução MSE.

I.2 CODIFICAÇÃO A TAXA ZERO SEGUIDA DE RECONSTRUÇÃO MSE COM INFORMAÇÃO LATERAL

Uma escolha possível é simplesmente utilizar taxa zero (Figura I.2), onde nenhuma informação é transmitida (i.e. $QP \rightarrow \infty$ or $M = 1$). Neste caso, o decodificador realiza a reconstrução $\hat{X}_Y(y)$, baseada no menor erro quadrático médio, a partir da função:

$$\begin{aligned} \hat{X}_Y(y) = E(X|Y = y) &= \int_{-\infty}^{\infty} x f_{X|Y}(x, y) dx \\ &= m_{X|Y}^{(1)}(\infty, y) \end{aligned}$$

Logo, a distorção esperada D_Y a taxa zero é dada por:

$$\begin{aligned} E(D_Y) &= \sigma_X^2 - \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} x f_{X|Y}(x, y) dx \right)^2 f_Y(y) dy \\ &= \sigma_X^2 - \int_{-\infty}^{\infty} m_{X|Y}^{(1)}(\infty, y)^2 f_Y(y) dy. \end{aligned}$$



Figura I.2: Codificação a taxa zero e reconstrução MSE.

I.3 CODIFICAÇÃO SLEPIAN-WOLF IDEAL COM RECONSTRUÇÃO MSE E INFORMAÇÃO LATERAL

Agora vamos a considerar o cenário onde a taxa e distorção esperadas são as obtidas da codificação ideal Slepian-Wolf dos *bins* quantizados, como descrito na Figura I.3. Note, que o conjunto ótimo da codificação ideal Slepian-Wolf da fonte C é obtido quando $M \rightarrow \infty$, i.e. quando realizamos codificação Slepian-Wolf diretamente nos *bins* quantizados. Assim, o codificador Slepian-Wolf ideal retorna uma taxa não maior a $H(Q|Y)$ para enviar os *bins* quantizados sem erros. Uma vez que os bins sem erros sejam recebidos, a reconstrução também será feita pelo mínimo MSE. Então, a taxa esperada é dada por:

$$E(R_{YQ}) = H(Q|Y) = - \int_{-\infty}^{\infty} \left\{ \sum_{q \in \Omega_Q} p_{Q|Y}(Q = q|Y = y) \log_2 p_{Q|Y}(Q = q|Y = y) \right\} \times f_Y(y) dy.$$

A qual pode ser re-escrita como:

$$E(R_{YQ}) = \int_{-\infty}^{\infty} \left\{ \sum_{q \in \Omega_Q} [m_{X|Y}^0(x_h(q), y) - m_{X|Y}^0(x_l(q), y)] \times \log_2 [m_{X|Y}^0(x_h(q), y) - m_{X|Y}^0(x_l(q), y)] \right\} f_Y(y) dy.$$

A distorção esperada D_{YQ} é a distorção gerada pela reconstrução MSE dado o *bin* quantizado q e a informação lateral y . Esta função de reconstrução $\hat{X}_{YQ}(y, q)$ é expressa como:

$$\hat{X}_{YQ}(y, q) = E(X|Y = y, Q = q) = \frac{\int_{x_l(q)}^{x_h(q)} x f_{X|Y}(x, y) dx}{\int_{x_l(q)}^{x_h(q)} f_{X|Y}(x, y) dx}.$$

A equação acima pode ser escrita da seguinte maneira:

$$\hat{X}_{YQ}(y, q) = \frac{m_{X|Y}^{(1)}(x_h(q), y) - m_{X|Y}^{(1)}(x_l(q), y)}{m_{X|Y}^{(0)}(x_h(q), y) - m_{X|Y}^{(0)}(x_l(q), y)}$$

Usando esta reconstrução, a distorção esperada (D_{YQ}), com *bins* livres de ruído, é dada por:

$$\begin{aligned} E(D_{YQ}) &= \sigma_X^2 - \int_{-\infty}^{\infty} \left\{ \sum_{q \in \Omega_Q} \frac{\left(\int_{x_l(q)}^{x_h(q)} x f_{X|Y}(x, y) dx \right)^2}{\int_{x_l(q)}^{x_h(q)} f_{X|Y}(x, y) dx} \right\} f_Y(y) dy \\ &= \sigma_X^2 - \left[\int_{-\infty}^{\infty} \left\{ \sum_{q \in \Omega_Q} \frac{\left(m_{X|Y}^{(1)}(x_h(q), y) - m_{X|Y}^{(1)}(x_l(q), y) \right)^2}{m_{X|Y}^{(0)}(x_h(q), y) - m_{X|Y}^{(0)}(x_l(q), y)} \right\} \right. \\ &\quad \left. \times f_Y(y) dy \right]. \end{aligned}$$

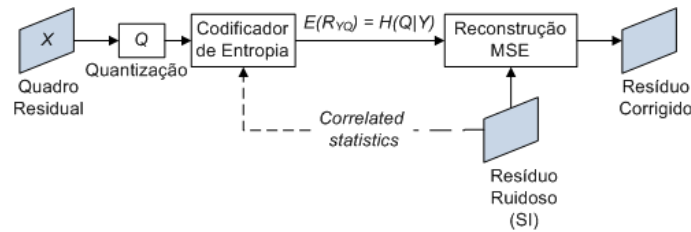


Figura I.3: Codificador Slepian-Wolf ideal com reconstrução MSE.

I.4 CODIFICAÇÃO CONVENCIONAL SEGUIDA DE RECONSTRUÇÃO MSE COM E SEM INFORMAÇÃO LATERAL

O último cenário a ser considerado é a taxa e distorção se não realizamos codificação distribuída nos *bins* quantizados (ver Figura I.4). Neste caso, a taxa esperada é simplesmente a entropia de Q :

$$\begin{aligned}
E(R_Q) = H(Q) &= - \sum_{q \in \Omega_Q} p_Q(q) \log_2 p_Q(q) \\
&= - \sum_{q \in \Omega_Q} \{ [m_X^{(0)}(x_h(q)) - m_X^{(0)}(x_l(q))] \\
&\quad \times \log_2 [m_X^{(0)}(x_h(q)) - m_X^{(0)}(x_l(q))] \}
\end{aligned}$$

Se a informação lateral Y estiver disponível, o decodificador pode ainda realizar a decodificação distribuída. Nesse caso, a função de reconstrução e sua correspondente distorção esperada são dadas pelas mesmas equações $\hat{X}_{YQ}(y, q)$ e $E(D_{YQ})$, respectivamente, usadas para o codificador Slepian-Wolf ideal com reconstrução MSE. Por outro lado, se não existe informação lateral disponível, a distorção esperada D_Q é a distorção gerada pela reconstrução MSE somente baseada no *bin* q . Essa função, $\hat{X}_Q(q)$, é dada por:

$$\begin{aligned}
\hat{X}_Q(q) = E(X/Q = q) &= \frac{\int_{x_l(q)}^{x_h(q)} x f_X(x) dx}{\int_{x_l(q)}^{x_h(q)} f_X(x) dx} \\
&= \frac{m_X^{(1)}(x_h(q)) - m_X^{(1)}(x_l(q))}{m_X^{(0)}(x_h(q)) - m_X^{(0)}(x_l(q))},
\end{aligned}$$

e a distorção esperada é expressa por:

$$\begin{aligned}
E(D_Q) &= \sigma_X^2 - \sum_{q \in \Omega_Q} \frac{\left(\int_{x_l(q)}^{x_h(q)} x f_X(x) dx \right)^2}{\left(\int_{x_l(q)}^{x_h(q)} f_X(x) dx \right)} \\
&= \sigma_X^2 - \sum_{q \in \Omega_Q} \frac{\left(m_X^{(1)}(x_h(q)) - m_X^{(1)}(x_l(q)) \right)^2}{\left(m_X^{(0)}(x_h(q)) - m_X^{(0)}(x_l(q)) \right)}.
\end{aligned}$$

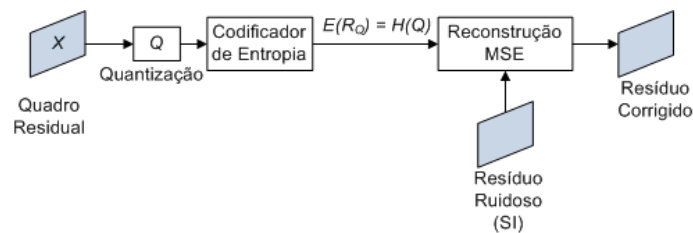


Figura I.4: Codificação convencional com reconstrução MSE

II. TABELAS DO D-CAVLC

II.1 TABELAS DE CODIFICAÇÃO PARA O NÚMERO DE COEFICIENTES E TRAILING ONES

II.1.1 Grupo I

O grupo I é selecionado quando a média da quantidade de coeficientes diferentes de zero do bloco acima e do bloco a esquerda é menor que 2.

Tabela II.1: Tabela de Codificação, grupo I, $10 < max_C \leq 16$. T1s significa número de *Trailing ones*.

NumCoef\T1s	0	1	2	3
0	1	-	-	-
1	000011	01	-	-
2	00000111	0001001	001	-
3	000001001	00000110	0001000	00011
4	000001000	000001011	000000101	000010
5	0000000111	000001010	000000100	0001011
6	00000000111	0000000110	0000001101	00010101
7	000000001001	00000000110	0000001100	00010100
8	000000001000	00000001001	000000001010	000000111
9	0000000000111	000000001011	000000000101	0000000101
10	0000000000110	0000000001101	0000000001111	00000001000
11	000000000000011	00000000001100	00000000001110	0000000000100
12	000000000000010	000000000000100	000000000000110	00000000000101
13	000000000000101	000000000000111	000000000010001	000000000001001
14	0000000000000011	000000000000010	000000000010000	0000000000000011
15	00000000000000001	00000000000000011	00000000000000010	000000000000000101
16	00000000000000000	0000000000000001001	0000000000000010001	0000000000000010000

Tabela II.2: Tabela de Codificação, grupo I, $6 < max_C \leq 10$.

NumCoef\T1s	0	1	2	3
0	1	-	-	-
1	000011	01	-	-
2	000100	000101	001	-
3	0000100	0000101	000000101	00011
4	00000101	00000110	000000111	00000100
5	00000111	000000110	0000000110	000000100
6	0000000100	0000000101	00000000101	0000000111
7	00000000110	00000000100	0000000001100	00000000111
8	0000000001110	0000000001101	0000000001111	0000000001001
9	0000000001010	0000000000100	0000000001011	0000000001000
10	0000000000101	0000000000110	0000000000001	0000000000111

Tabela II.3: Tabela de Codificação, grupo I, $3 < max_C \leq 6$.

NumCoef\T1s	0	1	2	3
0	1	-	-	-
1	00011	01	-	-
2	00010	000010	001	-
3	000011	0000011	000000011	000010
4	00000010	000000010	0000000010	00000011
5	0000000011	00000000010	000000000010	00000000011
6	000000000001	000000000011	0000000000001	000000000001

Tabela II.4: Tabela de Codificação, grupo I, $1 < max_C \leq 3$.

NumCoef\T1s	0	1	2	3
0	1	-	-	-
1	0001	01	-	-
2	000010	000011	001	-
3	0000001	0000011	00000001	0000010

Tabela II.5: Tabela de Codificação, grupo I, $0 \leq max_C \leq 1$.

NumCoef\T1s	0	1	2	3
0	1	-	-	-
1	001	01	-	-

II.1.2 Grupo II

O grupo II é escolhido se a média da quantidade de coeficientes diferentes de zero do bloco acima e do bloco a esquerda é menor que 4 e maior ou igual a 2.

Tabela II.6: Tabela de Codificação, grupo II, $10 < max_C \leq 16$.

NumCoeff \ T1s	0	1	2	3
0	11	-	-	-
1	000011	011	-	-
2	000010	00011	010	-
3	001001	001000	001010	101
4	1000001	001011	100101	0011
5	00000111	1000000	1000010	00010
6	00000110	1000011	1001101	10001
7	000001001	10011101	10011100	100100
8	000001000	000001011	000000101	1001100
9	0000000111	000001010	000000100	10011111
10	0000000110	0000001101	0000001100	10011110
11	00000000101	00000000111	00000001001	000000111
12	00000000100	00000000110	00000001000	0000000101
13	000000000011	000000000010	000000000100	000000000111
14	0000000000011	0000000000101	0000000000010	0000000001101
15	00000000000001	00000000000000	000000000000111	0000000001100
16	000000000000101	000000000000100	0000000000001101	000000000001100

Tabela II.7: Tabela de Codificação, grupo II, $6 < max_C \leq 10$.

NumCoeff\T1s	0	1	2	3
0	10	-	-	-
1	000101	11	-	-
2	00111	001001	011	-
3	0100	001000	0000100	0101
4	00110	001011	0000101	001010
5	000100	000110	00000101	000111
6	0000110	0000111	000000100	00000110
7	00000100	00000111	000000110	000000101
8	0000000100	000000111	0000000110	0000000101
9	0000000111	00000000100	00000000110	00000000101
10	00000000110	00000000011	00000000001	00000000010

Tabela II.8: Tabela de Codificação, grupo II, $3 < max_C \leq 6$.

NumCoeff\T1s	0	1	2	3
0	10	-	-	-
1	0100	11	-	-
2	011	00111	0101	-
3	00110	00101	000111	00100
4	000100	000101	000011	000110
5	0000101	0000100	0000010	0000011
6	00000011	00000010	000000001	00000001

Tabela II.9: Tabela de Codificação, grupo II, $1 < max_C \leq 3$.

NumCoeff\T1s	0	1	2	3
0	1	-	-	-
1	0001	01	-	-
2	000010	000011	001	-
3	0000011	0000001	00000001	00000010

Tabela II.10: Tabela de Codificação, grupo II, $0 \leq max_C \leq 1$.

NumCoeff\T1s	0	1	2	3
0	1	-	-	-
1	001	01	-	-

II.1.3 Grupo III

Este grupo é escolhido se a média da quantidade de coeficientes diferentes de zero do bloco acima e do bloco a esquerda é menor que 8 e maior ou igual a 5. Se a média da quantidade de coeficientes dos blocos vizinhos for maior ou igual a 8, então se utiliza os códigos de tamanho fixo de 6 bits.

Tabela II.11: Tabela de Codificação, grupo III, $10 < max_C \leq 16$.

NumCoeff\T1s	0	1	2	3
0	0011	-	-	-
1	0000011	0010	-	-
2	0000010	101110	1101	-
3	000011	101001	010110	1100
4	000010	101000	010001	1111
5	101101	101011	010000	1110
6	101100	101010	010011	1001
7	101111	010101	010010	1000
8	0110101	010100	011101	00011
9	0110100	010111	011100	00010
10	0110111	0110110	0110000	011111
11	01111001	0110001	01111010	0110011
12	01111000	01111011	01100101	01100100
13	000000011	000000010	000000100	000000111
14	0000000011	000000101	0000001101	0000001100
15	0000000010	00000000011	00000000010	00000000001
16	0000000000001	0000000000001	00000000000001	00000000000000

Tabela II.12: Tabela de Codificação, grupo III, $6 < max_C \leq 10$.

NumCoeff\T1s	0	1	2	3
0	1000	-	-	-
1	001111	1100	-	-
2	01010	01101	1011	-
3	1001	1111	001101	1110
4	1010	01000	001100	01100
5	1101	01001	001110	01111
6	01011	01110	001011	001001
7	001010	001000	000110	000101
8	000100	000111	0000100	0000110
9	0000101	0000111	00000100	00000101
10	00000111	00000011	00000010	00000110

Tabela II.13: Tabela de Codificação, grupo III, $3 < max_C \leq 6$.

NumCoeff\T1s	0	1	2	3
0	1000	-	-	-
1	1100	1001	-	-
2	1010	1101	1011	-
3	1110	1111	01101	01001
4	01000	01010	01110	01100
5	01011	00100	01111	00101
6	00110	00010	00101	00011

Tabela II.14: Tabela de Codificação, grupo III, $3 < max_C \leq 6$.

NumCoeff\T1s	0	1	2	3
0	10	-	-	-
1	010	11	-	-
2	0011	0010	011	-
3	00011	00001	000001	00010

Tabela II.15: Tabela de Codificação, grupo III, $0 \leq max_C \leq 1$.

NumCoeff\T1s	0	1	2	3
0	11	-	-	-
1	01	10	-	-

II.2 TABELAS DE CODIFICAÇÃO PARA O NÚMERO TOTAL DE ZEROS

As tabelas abaixo indicam os códigos utilizados para codificar o número total de zeros. Note que se a quantidade de coeficientes for zero ou igual ao número máximo de coeficientes que um bloco pode possuir, então não é necessário codificar o número total de zeros.

Tabela II.16: Códigos quando a quantidade Total de zeros máxima igual a 1.

NumCoeff\TotZeros	1
<i>0</i>	1
<i>1</i>	0

Tabela II.17: Códigos quando a quantidade Total de zeros máxima igual a 2.

NumCoeff\TotZeros	1	2
<i>0</i>	1	0
<i>1</i>	01	1
<i>2</i>	00	-

Tabela II.18: Códigos quando a quantidade Total de zeros máxima igual a 5.

NumCoeff\TotZeros	1	2	3	4	5
<i>0</i>	1	001	001	00	0
<i>1</i>	011	01	01	01	1
<i>2</i>	010	11	11	1	-
<i>3</i>	001	10	10	-	-
<i>4</i>	0001	000	-	-	-
<i>5</i>	0000	-	-	-	-

Tabela II.19: Códigos quando a quantidade Total de zeros máxima igual a 9.

NumCoeff\TotZeros	1	2	3	4	5	6	7	8	9
<i>0</i>	1	111	000	0001	000	000	001	00	0
<i>1</i>	011	110	001	011	001	001	01	01	1
<i>2</i>	010	101	010	0011	011	01	11	1	-
<i>3</i>	0011	100	011	0010	11	11	10	-	-
<i>4</i>	0010	011	100	10	10	10	-	-	-
<i>5</i>	00011	010	101	11	010	-	-	-	-
<i>6</i>	00010	001	110	010	-	-	-	-	-
<i>7</i>	00001	0001	111	-	-	-	-	-	-
<i>8</i>	000001	0000	-	-	-	-	-	-	-
<i>9</i>	000000	-	-	-	-	-	-	-	-

Tabela II.20: Códigos quando a quantidade Total de zeros máxima igual a 15.

NumCoeff\TotZeros	1	2	3	4	5	6	7	
<i>0</i>	1	111	0010	111101	01000	101100	111000	
<i>1</i>	011	101	1101	1110	01010	101101	111001	
<i>2</i>	010	011	000	0110	01011	1010	11101	
<i>3</i>	0011	001	010	1010	1110	001	1001	
<i>4</i>	0010	000	1011	000	011	010	1111	
<i>5</i>	00011	1000	1111	100	100	000	00	
<i>6</i>	00010	0101	011	110	1111	110	01	
<i>7</i>	000011	1001	100	1011	110	111	101	
<i>8</i>	000010	1100	0011	010	101	100	110	
<i>9</i>	0000011	01000	1110	001	001	011	100	
<i>10</i>	0000010	11011	1010	0111	000	10111	-	
<i>11</i>	00000001	11010	11000	1111	01001	-	-	
<i>12</i>	00000000	010010	110011	111100	-	-	-	
<i>13</i>	00000011	0100111	110010	-	-	-	-	
<i>14</i>	000000101	0100110	-	-	-	-	-	
<i>15</i>	000000100	-	-	-	-	-	-	
NumCoeff\TotZeros	8	9	10	11	12	13	14	15
<i>0</i>	101000	111000	10000	11000	1000	100	00	0
<i>1</i>	101001	111001	10001	11001	1001	101	01	1
<i>2</i>	10101	11101	1001	1101	101	11	1	-
<i>3</i>	1011	1111	101	111	0	0	-	-
<i>4</i>	110	00	01	0	11	-	-	-
<i>5</i>	00	01	11	10		-	-	-
<i>6</i>	111	10	00	-	-	-	-	-
<i>7</i>	01	110	-	-	-	-	-	-
<i>8</i>	100	-	-	-	-	-	-	-
<i>9</i>	-	-	-	-	-	-	-	-
<i>10</i>	-	-	-	-	-	-	-	-
<i>11</i>	-	-	-	-	-	-	-	-
<i>12</i>	-	-	-	-	-	-	-	-
<i>13</i>	-	-	-	-	-	-	-	-
<i>14</i>	-	-	-	-	-	-	-	-
<i>15</i>	-	-	-	-	-	-	-	-

III. PRODUÇÃO BIBLIOGRÁFICA

III.1 PUBLICAÇÕES EM REVISTA

III.1.1 Codificador distribuído com resolução mista

1. MACCHIAVELLO, B. L. E.; QUEIROZ, R. L. MUKHERJEE, D. Iterative Side-Information Generation in a Mixed Resolution Wyner-Ziv Framework. *a ser publicado em IEEE Transactions on Circuits and Systems for Video Technology*, 2009.
2. MACCHIAVELLO, B. L. E.; BRANDI, F.; PEIXOTO, E.; QUEIROZ, R. L.; MUKHERJEE, D. Side-Information Generation for Temporally and Spatially Scalable Wyner-Ziv Codecs. *Eurasip Journal on Image and Video Processing*, v. 2009, pp.1–12,2009.

III.1.2 Outras Contribuições

1. ANDRADE, M. M.; MACCHIAVELLO, B. L. E.; NASCIMENTO, F. A. O.; ROCHA, A. F.; VASCONCELOS, D. F.; JESUS, P. C.; CARVALHO, H. S. Algoritmo híbrido para segmentação do ventrículo esquerdo em imagens de ecocardiografia bidimensional. *Revista Brasileira de Engenharia Biomédica*, v. 22, p. 23-33, 2006.
2. ISHIHARA, J. Y. ; TERRA, M. H. ; MACCHIAVELLO, B. L. E. . H_{∞} Filtering for Rectangular Discrete-time Descriptor Systems. *a ser publicado em Automatica (Oxford)*, 2009.

III.2 PUBLICAÇÕES EM CONGRESSO

III.2.1 Codificador distribuído com resolução mista

1. MACCHIAVELLO, B. L. E.; QUEIROZ, R. L.; MUKHERJEE, D. Parameter Estimation for a H.264 Based Distributed Video Coder. *Proc. of the IEEE International Conference on Image Processing*, p. 1124-1127, 2008.
2. MACCHIAVELLO, B. L. E.; BRANDI, F.; QUEIROZ, R. L.; MUKHERJEE, D. Super-resolution

applied to Distributed Video Coding with Spatial Scalability. *Anais do Simpósio Brasileiro de Telecomunicações*, Rio de Janeiro, 2008.

3. MACCHIAVELLO, B. L. E.; MUKHERJEE, D.; QUEIROZ, R. L. A Statistical Model for a Mixed Resolution Wyner-Ziv Framework. *Proceedings of the 2007 Picture Coding Symposium*, Lisboa, 2007.
4. MACCHIAVELLO, B. L. E.; MUKHERJEE, D.; QUEIROZ, R. L. Motion-Based Side-Information Generation for a Scalable Wyner-Ziv Video Coder. *Proc. of the IEEE International Conference on Image Processing*, p. VI - 413-VI - 416, 2007.
5. MUKHERJEE, D.; MACCHIAVELLO, B. L. E.; QUEIROZ, R. L. A simple reversed-complexity Wyner-Ziv video coding mode. *Proceedings of SPIE Visual Communications and Image Processing* v. 6508. p. 65081Y1-65081Y12, 2007.
6. MACCHIAVELLO, B. L. E.; PEIXOTO, E.; QUEIROZ, R. L. A Video Coding Framework with Spatial Scalability. *Anais do XXV Simpósio Brasileiro de Telecomunicações*, Recife, 2007.

III.2.2 Outras Contribuições

1. REIS, M. C.; ROCHA, A. F.; VASCONCELOS, D. F.; MACCHIAVELLO, B. L. E.; NASCIMENTO, F. A. O.; CARVALHO, J. L. A.; SALOMONI, S.; CAMAPUM, J. F. Semi-automatic detection of the left ventricular border. *Proceedings of the IEEE 30th International Conference on Engineering in Medicine and Biology Society*, Vancouver, 2008.
2. MACCHIAVELLO, B. L. E.; ANDRADE, M. M.; NASCIMENTO, F. A. O.; CARVALHO, H. S.; VASCONCELOS, D. F.; ROCHA, A. F.; MELO JUNIOR, S.A. Algoritmo de segmentação de ECO2D dinâmica durante um ciclo completo. *Anais do XXVIII Congresso Ibero Latino-Americano sobre Métodos Computacionais em Engenharia*, Porto, 2007.
3. ISHIHARA, J. Y.; MACCHIAVELLO, B. L. E.; TERRA, M. H. H-infinity Estimation and Array Algorithms for Descriptor Systems. *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, 2006.
4. MACCHIAVELLO, B. L. E.; ISHIHARA, J. Y.; TERRA, M. H. Algoritmos Array para estimadores H-infinito na forma de Informação. *Anais do XVI Congresso Brasileiro de Automática*, p. 3068-3073, 2006.

5. MACCHIAVELLO, B. L. E.; ANDRADE, M. M.; NASCIMENTO, F. A. O.; ROCHA, A. F.; CARVALHO, H. S. Sistema Automático para a Geração da Curva de Variação de Área do Ventriculo Esquerdo no Ciclo Cardíaco Completo. *Anais do III Congresso Latino-Americano de Engenharia Biomedica*, v. 5. p. 1259-1262, 2004.
6. ANDRADE, M. M.; MACCHIAVELLO, B. L. E.; NASCIMENTO, F. A. O.; ROCHA, A. F.; CARVALHO, H. S.; VASCONCELOS, D. F. Segmentação Automática do Ventriculo Esquedo em Imagens de Ecocardiografia Bidimensional. *Anais do III Congresso Latino-Americano de Engenharia Biomedica*, p. 1387-1390, 2004.