

4 Codificador WZ-LIA

4.1. Motivação

O método mais utilizado na codificação distribuída de vídeo é o método de codificação no domínio do pixel (PDWZ – *Pixel Domain Wyner-Ziv Codec*) [69, 72], onde não se aplica qualquer tipo de transformada nos frames, tanto para o codificador Wyner-Ziv, a partir de agora abreviado por WZ, quanto para o codificador *intraframe*. Entretanto, conforme os resultados apresentados por Girod et al. [52, 68], utilizando-se um codificador distribuído de vídeo no domínio da transformada (TDWZ - *Transform Domain Wyner-Ziv Codec*), obtém-se um melhor desempenho. Inclusive, nesses mesmos artigos [52], foram apresentados resultados comparativos entre o método tido como o mais eficiente até o momento para codificação distribuída de vídeo e os métodos tradicionais de codificação, onde se observa que o desempenho taxa-distorção (RD: *Rate-Distortion*) do método TDWZ supera o da codificação H.264 *intraframe*, mas ainda está um pouco abaixo do desempenho obtido pela codificação H.264 *interframe* com uma estrutura de GOP igual a I-B-I (sequência em que os frames ímpares são *intraframes* e os pares *interframes* com predição bidirecional) [51, 84].

Já a codificação por meio da transformada é uma ferramenta comumente usada na codificação tradicional de vídeo para explorar a correlação espacial dentro de uma imagem ou frame de uma sequência de vídeo. Geralmente, a transformada é aplicada sobre blocos de $N \times N$ amostras de um frame, decorrelatando espacialmente as amostras dentro de um bloco, isto é, convertendo valores de pixels correlatados em coeficientes transformados independentes. Visto que, dentro de um bloco, as amostras vizinhas são geralmente fortemente correlatadas, é possível representar com maior eficiência essas amostras no domínio da frequência. A decorrelação dentro de um bloco de $N \times N$ amostras permite que a energia do bloco seja concentrada num pequeno

número de coeficientes transformados de valores maiores. O coeficiente chamado DC (por corresponder à menor frequência espacial) e geralmente os coeficientes próximos aos coeficientes DC englobam a maior parte da energia do bloco de amostras $N \times N$. A quantidade de bits necessários para codificar um frame pode ser reduzida caso sejam considerados somente aqueles coeficientes de baixa frequência (e de valores altos) e desprezando os coeficientes DCT remanescentes, visto que estes possuem amplitudes bem menores. Geralmente, o impacto na qualidade do frame decodificado por somente codificar os coeficientes de baixa frequência não é notado pelo sistema visual humano (HVS – *Human Visual System*), embora isto dependa fortemente do conteúdo do vídeo [85-86], ou seja, o que geralmente não se aplica a blocos com alta textura.

Conforme mencionado no capítulo 2, a codificação por transformada é uma ferramenta que pode ser utilizada na codificação distribuída de vídeo com o mesmo propósito de seu uso na codificação tradicional de vídeo, ou seja, para explorar a correlação entre valores de amostras vizinhas. Esta correlação espacial não é explorada no método de codificação distribuída PDWZ e assim, um melhor desempenho RD pode ser alcançado se a ferramenta de codificação por transformada for utilizada. No entanto, o uso desta ferramenta não deve comprometer o requerimento de codificação de baixa complexidade necessário às diversas aplicações emergentes, como por exemplo, redes de vídeo-vigilância wireless de baixa potência.

4.2. Arquitetura do Codec WZ-LIA

A arquitetura do codificador distribuído WZ-LIA, proposto neste trabalho, cujo diagrama de blocos é descrito pela figura 4.1, é baseada no esquema de codificação do *estado da arte* em DVC apresentado em [68], mas com modificações e otimizações de diversas ferramentas, principalmente em seus componentes mais essenciais, tais como no decodificador de canal (o qual utiliza o código LIA proposto descrito no capítulo 3), no módulo de geração da informação lateral e na modelagem do canal de correlação. Assim, buscou-se combinar e aperfeiçoar as técnicas de codificação empregadas comumente no projeto DVC para melhoria do desempenho dos módulos básicos de

processamento tanto da codificação quanto da decodificação e para lidar com os problemas associados à estimação *online* dos parâmetros usados na decodificação.

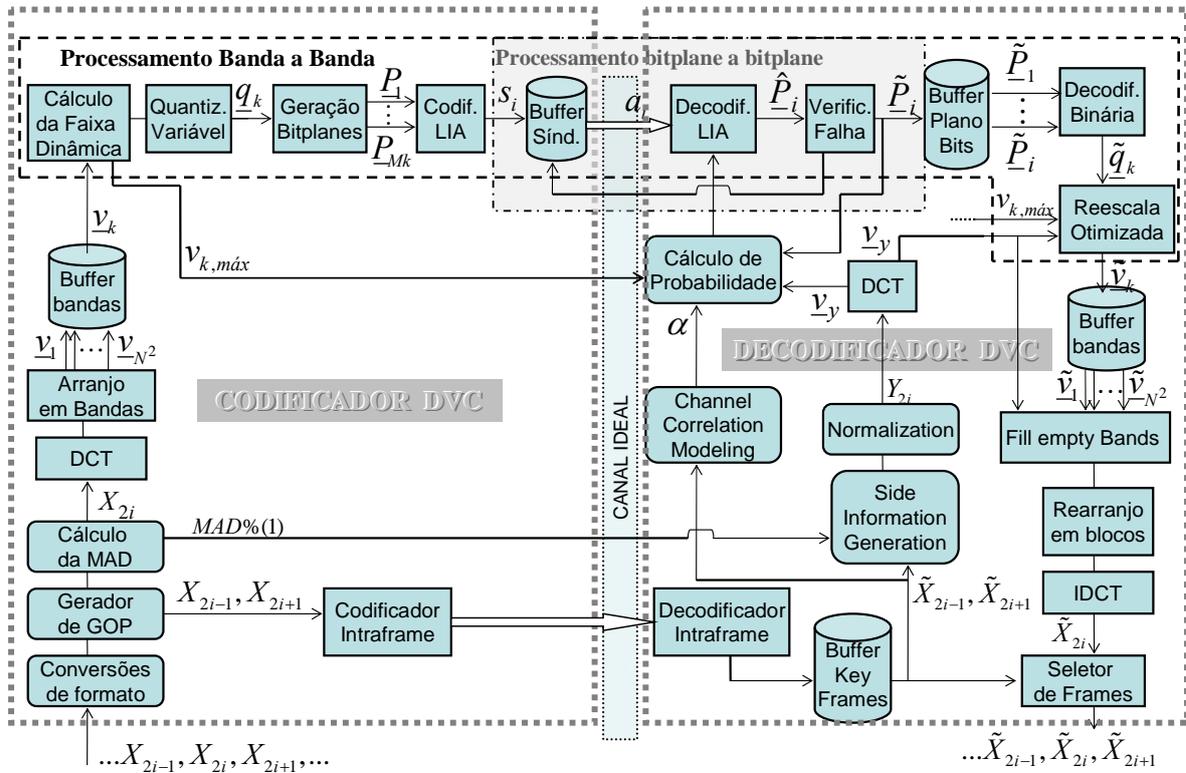


Figura 4.1: Diagrama de blocos da arquitetura de codificação distribuída proposta.

4.3. Procedimento de Codificação

Neste item, o procedimento de codificação ilustrado pela figura 4.1, será descrito sucintamente. Ressalta-se que o detalhamento das técnicas utilizadas, assim como as equações e algoritmos empregados nas mesmas serão descritos na seção 4.5 e a implementação do codec proposto foi feita em Matlab 7.0.

❖ Conversão e geração de GOP:

- Primeiro, uma sequência de vídeo é adequada ao formato especificado, o que é uma abordagem opcional proposta neste trabalho. Depois a sequência é dividida em frames Wyner-Ziv (os frames pares da sequência de vídeo) e frames *key* (os frames ímpares da mesma), que é a abordagem comumente utilizada para GOP em DVC.

❖ Codificação dos frames *key*:

- Antes de iniciar a codificação WZ, a qual faz uso necessariamente da informação lateral para que ocorra uma decodificação bem sucedida, executa-se a codificação e a decodificação dos frames *key*, os quais são submetidos ao mesmo processo de codificação e decodificação dos intraframes utilizados nos codecs convencionais de vídeo, já que para a decodificação de um frame Wyner-Ziv X_{2i} , faz-se necessário a disponibilidade dos frames *key* no decodificador (reconstruídos), ou melhor, dos frames \tilde{X}_{2i-1} e \tilde{X}_{2i-1} .
- As etapas do processo de codificação *intraframe* já são bem conhecidas e, por isso, fogem do foco deste trabalho. Sendo assim estas são apenas citadas resumidamente como: predição *intraframe*; aplicação da transformada DCT; quantização; escaneamento e codificação de comprimento variável (codificação entrópica). Na decodificação ocorre o processo inverso [50, 51].

❖ Geração das Bandas DCT:

- Aplica-se a transformada DCT baseada em blocos $N \times N$ sobre cada frame Wyner-Ziv X_{2i} , conforme definido pelo padrão de codificação de vídeo H.264/MPEG-4 AVC [84]. A transformada DCT 4×4 foi escolhida a fim de proporcionar uma melhor comparação entre os resultados que serão obtidos com base no codec proposto e os disponíveis em [52-54], já que este é o padrão de transformada (e de dimensão do bloco) mais utilizado e mais eficiente na codificação distribuída de vídeo [54]. No codec proposto, há também a possibilidade de alterar o tamanho do bloco DCT de 4×4 para 8×8 ou 16×16 .
- Os coeficientes provenientes da transformada aplicada aos blocos 4×4 do frame X_{2i} , chamados de coeficientes DCT, são agrupados de acordo com sua posição ocupada dentro dos blocos 4×4 , formando bandas de coeficientes DCT (bandas de frequência espacial). Uma vez que se utiliza uma transformada baseada em blocos 4×4 , existem 4^2 posições diferentes dentro do bloco e, portanto, podem ser

formadas 4^2 bandas diferentes de coeficientes DCT. Assim, o número de elementos em cada banda de coeficientes DCT é igual ao número de blocos 4×4 no frame.

- O comprimento de uma banda de coeficientes DCT é dado pela relação entre o tamanho do frame e o número de bandas, no caso, 4^2 (que é igual ao tamanho do bloco processado pela DCT). Esta razão corresponde ao tamanho do vetor de entrada tanto no quantizador proposto quanto no módulo de geração de bitplanes, visto que cada banda (ou vetor) de coeficientes DCT é codificada separadamente, de acordo com a ilustração da figura 4.1.

❖ Quantização Variável:

- Depois da operação de transformada e separação dos coeficientes em bandas, cada banda de coeficientes DCT é uniformemente quantizada com $L_k = 2^{M_k}$ níveis, onde M_k é o número de bits necessário para codificar a k -ésima banda, ou melhor, a variação por banda no número de níveis de quantização implica também em uma variação no número de bits necessário para codificar cada banda. Portanto, a quantização é uma função da banda. O resultado da quantização é o *stream* de símbolos quantizados da banda b_k , denominado q_k .

❖ Geração dos Bitplanes:

- Sobre o *stream* resultante de símbolos de quantização associada à banda b_k de coeficientes DCT, realiza-se uma codificação binária, fazendo-se um mapeamento de M_k bits para cada símbolo. A partir dos códigos binários gerados, extraem-se os M_k *bitplanes* da banda b_k , ou seja, os bits representando os símbolos de quantização são ordenados em *bitplanes* e são alimentados no codificador de canal formado pelo código LIA. Na figura 4.1, os bitplanes da k -ésima banda estão representados por $\underline{P}_1, \underline{P}_2, \dots, \underline{P}_{M_k}$. É importante destacar que para cada símbolo (de quantização) da banda b_k há um código binário de comprimento M_k associado. Assim, havendo n

coeficientes da transformada, haverá n símbolos e n bits em cada bitplane, onde n é o número de blocos no frame.

❖ Geração e Transmissão das Síndromes:

- Ao receber cada bitplane i da banda b_k , o codificador LIA calcula um conjunto de bits de paridade (informação redundante) representando a síndrome completa s_i do bitplane codificado P_i , de acordo com a ilustração da figura 4.1.
- Em seguida, as síndromes s_i associadas a cada bitplane são acumuladas (denominadas a_i), conforme critérios estabelecidos na seção 3.2.5.1, armazenadas em um *buffer* e enviadas por partes para o decodificador. A princípio, enviam-se pequenas quantidades de bits de paridade (representados pelas síndromes acumuladas), caso a decodificação seja considerada mal sucedida, o decodificador pede mais bits ao codificador através do canal de *feedback*. Descrevendo o processo de outra forma, o codificador transmite uma síndrome curta, baseada em um código menos preciso (ou com menor probabilidade de correção de erros) e o decodificador tenta realizar a decodificação. No evento em que a decodificação obtém êxito (ver seção 4.6.3.1), o decodificador sinaliza este fato para o codificador, o qual, então, continua com o próximo bloco de dados da fonte (com o próximo bitplane da mesma banda b_k). Caso contrário, mais bits de paridade são requisitados.

4.4. Procedimento de Decodificação

❖ Geração da Informação Lateral:

- A primeira tarefa executada pelo decodificador DVC é a reconstrução dos frames *key* X_{2i-1} e X_{2i+1} (*intraframes*), os quais são adjacentes (anterior e posterior) ao frame Wyner-Ziv atual X_{2i} .
- Assim, o decodificador executa o processo de geração de uma informação ou frame auxiliar lançando mão dos frames *key* reconstruídos, \tilde{X}_{2i-1} e \tilde{X}_{2i+1} , para criar uma estimativa do frame X_{2i} ,

chamada informação lateral e representada pelo frame Y_{2i} . As ferramentas desenvolvidas aqui e utilizadas nesse processo serão descritas na seção 4.6.7.

❖ Normalização da Informação Lateral:

- Em seguida, a informação lateral é normalizada a fim de torná-la uma melhor estimativa do frame atual. A validação deste módulo pode ser verificada pelos experimentos das seções 4.7.1 e 4.7.2. A fórmula utilizada na normalização é dada por:

$$\underline{Y}_{2i,N} = \beta \cdot \underline{Y}_{2i} / \text{Máx}\{\underline{Y}_{2i}\}$$

em que β representa um fator fixo de normalização, o qual é discutido na seção 4.7.1.

- Depois disso, uma transformada DCT 4x4 é aplicada sobre o frame interpolado normalizado $\underline{Y}_{2i,N}$ obtendo-se $\underline{v}_{y,2i}$, uma estimativa dos coeficientes de \underline{X}_{2i}^{DCT} , os quais serão representados por $\underline{v}_{x,2i}$ ou simplesmente \underline{v}_{2i} .

❖ Estimação dos Parâmetros do Canal de Correlação:

- Considera-se que a estatística do resíduo entre os coeficientes de \underline{v}_{2i} e seus correspondentes em $\underline{v}_{y,2i}$ pode ser modelada por uma distribuição laplaciana e, conforme [54], é similar à estatística do resíduo entre os coeficientes de X_{2i-1} e X_{2i+1} (devido à semelhança entre estes e entre os coeficientes de \underline{v}_{2i} e de sua estimativa $\underline{v}_{y,2i}$). Neste trabalho, o parâmetro laplaciano é estimado em tempo real e oferece duas possibilidades diferentes de estimação: em nível de banda DCT, isto é, para cada banda haverá um parâmetro laplaciano associado, ou em nível de coeficiente, associando um parâmetro laplaciano a cada coeficiente.

❖ Decodificação dos Bitplanes:

- Uma vez que os coeficientes DCT de Y_{2i} , representados por $\underline{v}_{y,2i}$, e as estatísticas do resíduo para uma dada banda b_k de coeficientes DCT são conhecidos, podem ser calculadas as probabilidades condicionais $p(x_i | x_{i-1}, \underline{y})$. Assim, obtêm-se os M_k bitplanes da banda b_k a partir destas duas informações adicionais (ou laterais) e mais a síndrome enviada pelo codificador WZ, através de um procedimento iterativo de decodificação e verificação de erros.

❖ Reconstrução dos índices de Quantização:

- Posteriormente, reordenando os bitplanes em códigos binários (os códigos originais) e decodificando os mesmos, obtêm-se o *stream* de símbolos de quantização reconstruídos da banda b_k do frame X_{2i} , $\tilde{q}_{2i,k}$.

❖ Reconstrução das Bandas de Coeficientes DCT:

- O módulo de reconstrução, representado pelo nome de reescala otimizada na figura 4.1, faz uso do *stream* \tilde{q}_{2i} e dos coeficientes DCT $\underline{v}_{y,2i}$ (da informação lateral) para reconstruir cada banda de coeficientes DCT do frame X_{2i} , obtendo-se \tilde{v}_{2i} .
- Depois de todas as bandas de coeficientes DCT serem reconstruídas e estarem disponíveis no *buffer*, há o preenchimento das bandas vazias, ou melhor, os coeficientes das bandas para as quais nenhum bit foi enviado serão substituídos pelos correspondentes da informação lateral.

❖ Reconstrução do Frame WZ:

- Completando-se os coeficientes de cada banda, reagrupa-se os mesmos em blocos $N \times N$ e, então, realiza-se a transformada inversa da DCT $N \times N$ (representada pelo módulo IDCT na figura 4.1) e obtêm-se, enfim, o frame Wyner-Ziv reconstruído \tilde{X}_{2i} . Nas seções

posteriores, a implementação e a descrição dos principais módulos de codificação e decodificação serão apresentadas com maiores detalhes.

4.5. Métodos e Técnicas do Codec Proposto

4.5.1. Transformada DCT no Codec WZ-LIA

Uma transformada baseada em blocos é aplicada à arquitetura de codificação Wyner-Ziv com o mesmo propósito de seu uso nos métodos tradicionais de codificação de vídeo: descorrelatar as amostras do bloco através da exploração da redundância espacial entre as amostras vizinhas, e compactar a energia do bloco em tão poucos coeficientes quanto possível.

A transformada de Karhunen-Loève (KLT) é a transformada ótima em termos de capacidade de compactação de energia [85]. Entretanto, a transformada KLT é dependente do sinal, isto é, as funções-base da KLT são dependentes do sinal a ser transformado. Para um vasto conjunto de sinais, a transformada discreta de cossenos (DCT) é uma estimativa bem próxima da transformada KLT [86], com a vantagem de que as funções-base DCT são independentes do sinal. De fato, a transformada DCT é amplamente utilizada pelos padrões de codificação de vídeo (padrões H.264/MPEG-4 AVC [84]).

Geralmente, a transformada DCT unidimensional converte um vetor \underline{x} de N x 1 amostras em um novo vetor \underline{X} de N x1 coeficientes transformados (no domínio da frequência), por uma transformação linear dada pela eq. (4.1), onde A é a matriz de transformação N x N .

$$\underline{X}_{N \times 1} = \underline{A}_{N \times N} \cdot \underline{x}_{N \times 1} \quad (4.1)$$

A transformada DCT também pode ser aplicada a uma matriz de amostras N x N . Neste caso, uma transformada DCT bidimensional é empregada, visto que há duas dimensões a serem consideradas. A transformada DCT bidimensional é utilizada aplicando-se a transformada DCT monodimensional duas vezes, uma para a dimensão vertical e outra para a horizontal [85]. Na primeira etapa,

explora-se a correlação horizontal dentro do bloco de $N \times N$ amostras, e na segunda etapa, a transformada DCT unidimensional é aplicada para explorar a correlação vertical.

A transformada empregada no codec WZ-LIA é a DCT baseada em blocos $N \times N$, conforme definido pelo padrão H.264/MPEG-4 AVC [84], a fim de permitir a comparação dos mesmos com o codec proposto. As maiores características da transformada DCT 4x4, utilizada em [84], são as seguintes:

- Ela é uma transformada baseada em blocos 4x4 de números inteiros, isto é, todas as operações podem ser executadas utilizando somente adições, subtrações e deslocamentos de bits, sem perda de precisão.
- Todas as operações podem ser executadas usando aritmética de 16 bits ao invés de 32 em uma DCT de não-inteiros, reduzindo a complexidade computacional.
- A matriz de transformação A é uma matriz 4x4, definida como [84]:

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}. \quad (4.2)$$

- E a matriz de transformação inversa A_{inv} é uma matriz 4x4, definida como:

$$A_{inv} = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix}. \quad (4.3)$$

- Note que as multiplicações por 2 em (4.2) e por 1/2 em (4.3) podem ser implementadas com deslocamentos de 1 bit à esquerda e à direita, respectivamente, permitindo a todos os decodificadores obterem os mesmos resultados. A relação entre as matrizes A_{inv} e A é dada pela eq. (4.4), onde I é a matriz identidade.

$$H_{inv} \begin{bmatrix} 1/4 & 0 & 0 & 0 \\ 0 & 1/5 & 0 & 0 \\ 0 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 1/5 \end{bmatrix} H = I \quad (4.4)$$

No método de codificação WZ no domínio da transformada, a transformada DCT é aplicada a todos os blocos 4x4 (não sobrepostos) do frame X_{2i} , da esquerda para direita e de cima para baixo. A transformada DCT unidimensional é caracterizada pela matriz de transformação dada pela eq. (4.1).

Após aplicar a transformada DCT a um bloco de amostras de tamanho 4x4, as 16 amostras correlatas dentro deste bloco são convertidas em 16 coeficientes (DCT) independentes, no domínio da frequência espacial. Estes coeficientes são arranjados em um bloco 4x4 chamado de bloco de coeficientes DCT, conforme ilustrado na figura 4.2. Os coeficientes DCT no canto superior esquerdo de cada bloco 4x4 são chamados de coeficientes DC. Os 15 coeficientes remanescentes são conhecidos como coeficientes AC e correspondem a frequências espaciais diferentes de zero, sendo que o coeficiente AC localizado no canto inferior direito (neste caso, na posição 16) corresponde à frequência espacial mais alta.

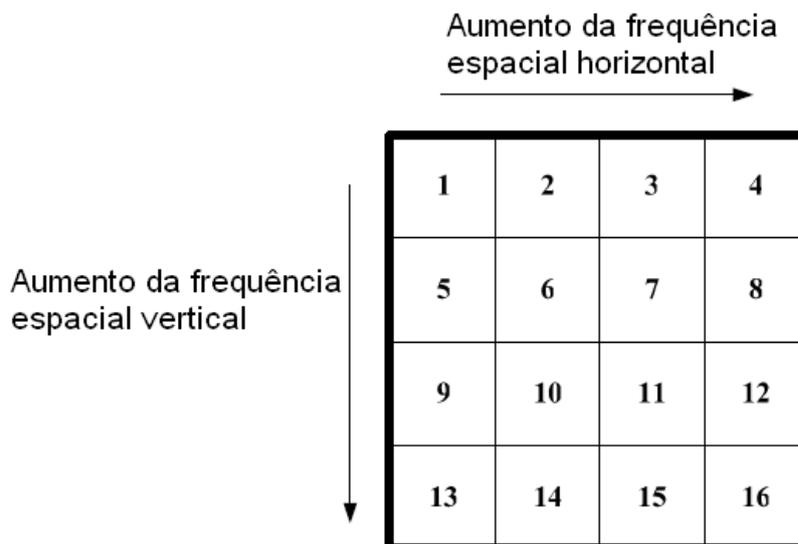


Figura 4.2: Ordem das posições (frequências espaciais) dentro de um bloco 4x4 de coeficientes DCT.

Uma vez que a operação de transformada DCT foi aplicada sobre todos os blocos de amostras 4x4 do frame X_{2i} , os coeficientes DCT são agrupados de

acordo com a posição ocupada pelos mesmos dentro do bloco transformado, conforme figura 4.2, formando as bandas de coeficientes DCT. Em outras palavras, a banda b_k de coeficientes DCT engloba os coeficientes que ocupam a posição k dentro de cada bloco 4x4 de coeficientes (do frame). Na verdade, qualquer ordem é possível para numeração das bandas DCT, já que o processo de codificação e decodificação é executado banda por banda, o que não acontece, por exemplo, quando se utiliza o escaneamento zig-zag para maximizar o impacto subjetivo. A primeira banda de coeficientes DCT b_1 corresponde à banda de coeficientes DC e a banda de coeficientes b_{16} corresponde à banda de coeficientes AC de maior frequência. As equações (4.5) a (4.7) apresentam essas relações de coeficientes e bandas.

$$X_{2i}^{DCT} = \underline{v}_{2i} = \begin{bmatrix} \underline{v}_1 \\ \underline{v}_2 \\ \vdots \\ \underline{v}_{16} \end{bmatrix} \quad (4.5)$$

onde cada banda b_k de coeficientes DCT, \underline{v}_k , é constituída por um vetor, cujo comprimento corresponde ao número de blocos 4x4 no frame, n . Assim:

$$\underline{v}_k = \begin{bmatrix} v_{k,1} & v_{k,2} & \dots & v_{k,n} \end{bmatrix} \quad (4.6)$$

onde $v_{k,j}$ é o coeficiente DCT do j -ésimo bloco 4x4, pertencente à banda k .

No decodificador, uma estimativa do frame X_{2i} , representada na figura 4.1 por Y_{2i} (também conhecida como informação lateral), é obtida através dos frames adjacentes (frames anterior e posterior) ao frame X_{2i} , por meio de uma técnica de interpolação. Em seguida, a estimativa do decodificador Y_{2i}^{DCT} (estimativa dos coeficientes DCT), correspondente a X_{2i}^{DCT} , é obtida pela aplicação de uma transformada DCT 4x4 sobre o frame Y_{2i} . Então, o decodificador LIA utiliza Y_{2i}^{DCT} para obter o *stream* $\tilde{q}_{2i,k}$ de símbolos quantizados reconstruídos associado à banda DCT b_k . A estimativa Y_{2i}^{DCT} juntamente com o *stream* $\tilde{q}_{2i,k}$ são necessários para ajudar na tarefa de reconstrução das bandas de coeficientes DCT, X_{2i}^{DCT} , conforme será descrito na seção 4.7.4. Em seguida, os coeficientes DCT

agrupados em bandas são reagrupados nos respectivos blocos $B_{2i,j}$, definidos como:

$$B_{2i,j} = \begin{bmatrix} v_{1,j} & \cdots & v_{4,j} \\ \vdots & \ddots & \vdots \\ v_{13,j} & \cdots & v_{16,j} \end{bmatrix} \quad (4.7)$$

Enfim, a operação inversa da transformada DCT, conhecida como IDCT, deve ser executada sobre cada bloco representado pela eq. (4.7), a fim de obter o frame X_{2i} reconstruído, denominado \tilde{X}_{2i} . Conforme descrição feita na figura 4.1, a operação IDCT é realizada sobre \tilde{X}_{2i}^{DCT} , ou melhor, sobre cada bloco $B_{2i,j}$ de coeficientes DCT. A transformada inversa IDCT é executada de uma maneira similar à operação de transformada DCT, descrita anteriormente nesta seção. Entretanto, ao invés de se usar a matriz de transformação A , a transformada IDCT emprega a matriz de transformação inversa A_{inv} , descrita pela eq. (4.3).

4.5.2. Quantizador do Codec WZ-LIA

Após realizar a operação de transformada DCT 4x4 no codificador Wyner-Ziv, cada banda b_k de coeficientes DCT é independentemente codificada. Isto significa que há um número de níveis e um intervalo de quantização diferente para cada banda. Quantização é o primeiro passo no processo de codificação da banda de coeficientes DCT. O método de codificação Wyner-Ziv apresentado aqui faz uso de um quantizador uniforme distinto para cada banda, havendo também a possibilidade de ser implementado um quantizador não-uniforme.

4.5.2.1. Quantização dos Coeficientes DC

A banda de coeficientes DC é caracterizada por valores positivos de alta amplitude (valores muito maiores do que o módulo dos coeficientes das outras bandas), já que cada coeficiente DC da transformada expressa a energia média do bloco de amostras 4x4 correspondente. Uma vez que somente valores positivos são alimentados no quantizador, o algoritmo de quantização para a banda de

coeficientes DC pode ser similar ao utilizado na implementação no domínio do pixel. A figura 4.3 ilustra o quantizador escalar uniforme utilizado no procedimento de quantização dos coeficientes DC, onde v representa o eixo de valores que podem ser assumidos pelo coeficiente DC, e os números 0, 1, 2,... acima do eixo v simbolizam os índices de quantização e W a largura dos intervalos de quantização.

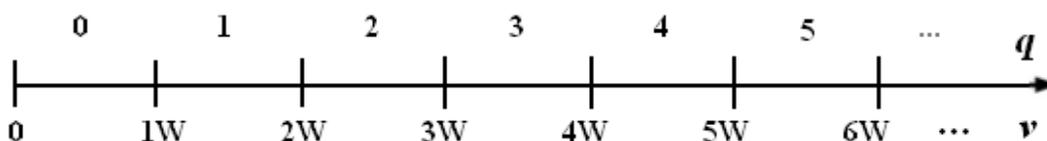


Figura 4.3: Quantizador escalar uniforme para o coeficiente DC com largura W do intervalo de quantização.

As figuras 4.4 a 4.7 exibem a distribuição de coeficientes DC (bandas b_1) das sequências de vídeo *Foreman*, *News*, *Coastguard* e *Soccer* respectivamente. Melhor dizendo, as figuras 4.4 a 4.7 apresentam a distribuição de probabilidade de ocorrer certo valor de coeficiente DC da transformada DCT 4x4, medida com base na sequência inteira. Os coeficientes DC de cada bloco DCT correspondem à frequência mais baixa do mesmo. Observando estas figuras (4.4 e 4.7), nota-se que as sequências *News* possui uma concentração de coeficientes da banda b_1 (banda DC) bem diferente das distribuições das sequências *Foreman*, *Soccer* e *Coastguard*, pois nas últimas há maior concentração de coeficientes próximos ao valor máximo, enquanto que na distribuição da sequência *News* há maior concentração de coeficientes próximos ao valor mínimo, isto porque os valores dos pixels da sequência *Foreman*, *Soccer* e *Coastguard* são bem maiores que os da *News*, ou seja, as cenas de vídeo desta última possui um valor menor de luminância. Vale ressaltar que a apresentação das curvas de distribuição tanto para a banda DC quanto para as bandas AC tem por objetivo justificar o cálculo da faixa dinâmica (ou limites de quantização) proposta neste trabalho.

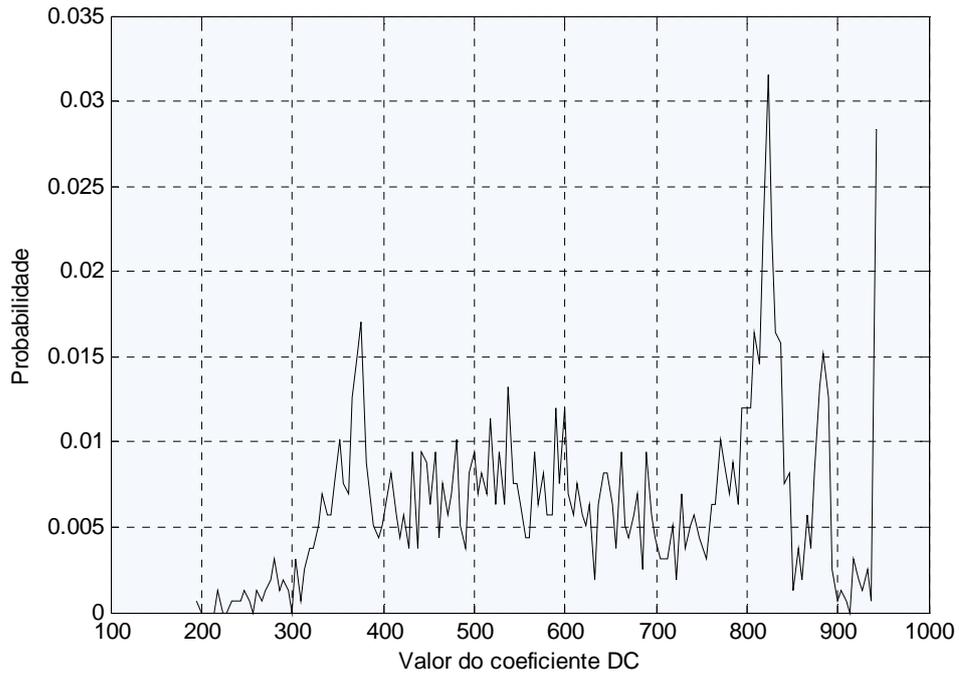


Figura 4.4: Distribuição dos coeficientes DCT para a banda DC (b_1) da sequência *Foreman*.

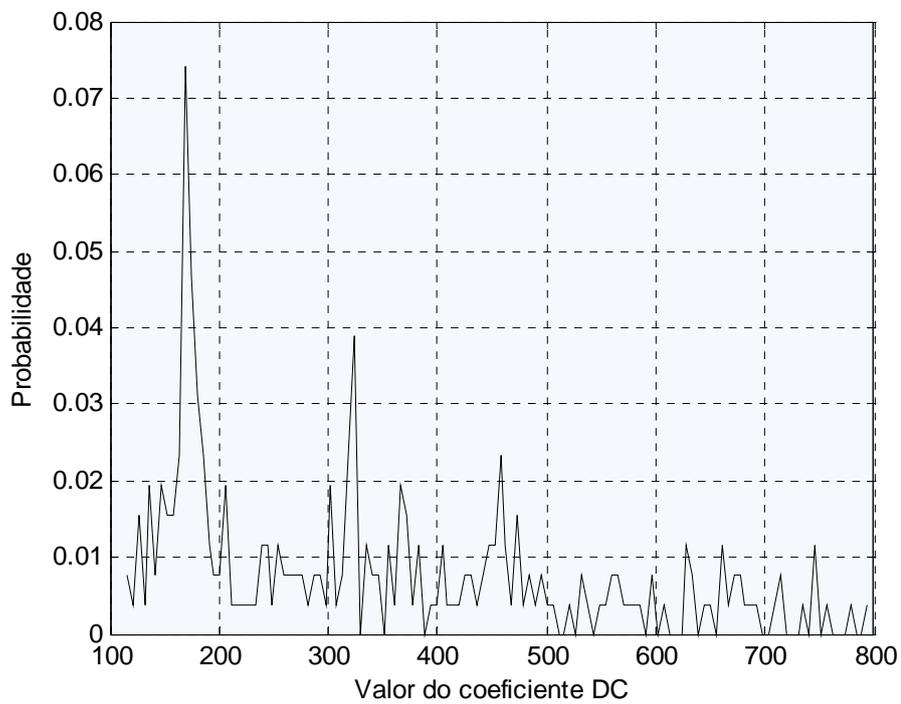


Figura 4.5: Distribuição dos coeficientes DCT para a banda DC (b_1) da sequência *News*.

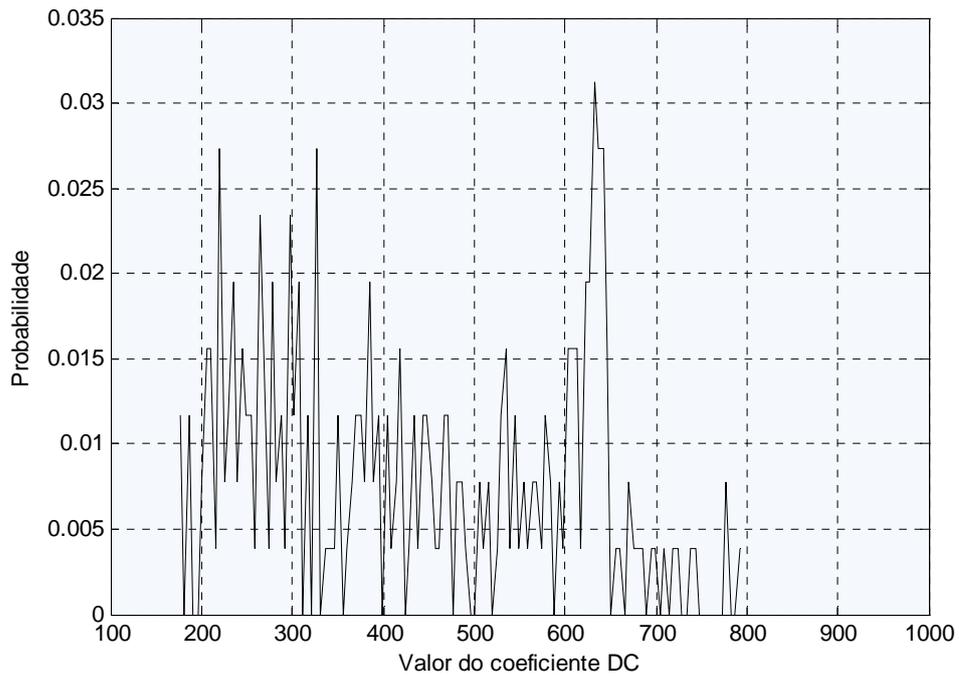


Figura 4.6: Distribuição dos coeficientes DCT para a banda DC (b_1) da sequência *Coastguard*.

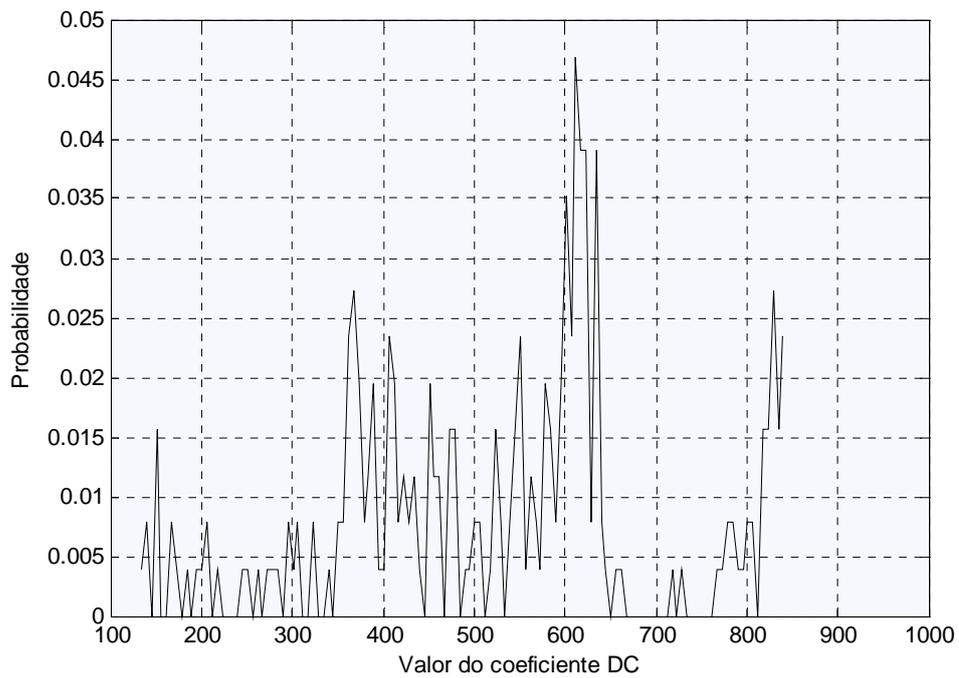


Figura 4.7: Distribuição dos coeficientes DCT para a banda DC (b_1) da sequência *Soccer*.

4.5.2.2. Quantização do Coeficiente AC

Para as 15 bandas remanescentes de coeficientes DCT, chamadas de bandas AC, a entrada do quantizador assume tanto valores positivos quanto negativos, já que mesmo para a DCT 4x4 as funções-base associadas aos coeficientes AC apresentam valor médio igual a zero [85].

As figuras 4.8 a 4.15 apresentam a distribuição dos coeficientes AC das sequências de vídeo *Foreman*, *News*, *Coastguard* e *Soccer*, QCIF, para as bandas b_2 e b_3 correspondendo às bandas AC em ordem crescente de frequência espacial e ordem decrescente do módulo dos valores dos coeficientes. Segundo essas figuras, observa-se que a distribuição dos coeficientes AC, diferentemente da distribuição DC, tende a ser aproximadamente simétrica em torno da amplitude zero. Esta característica da distribuição de coeficientes DCT ocorre não somente para as bandas b_2 e b_3 , mas para todas as bandas AC.

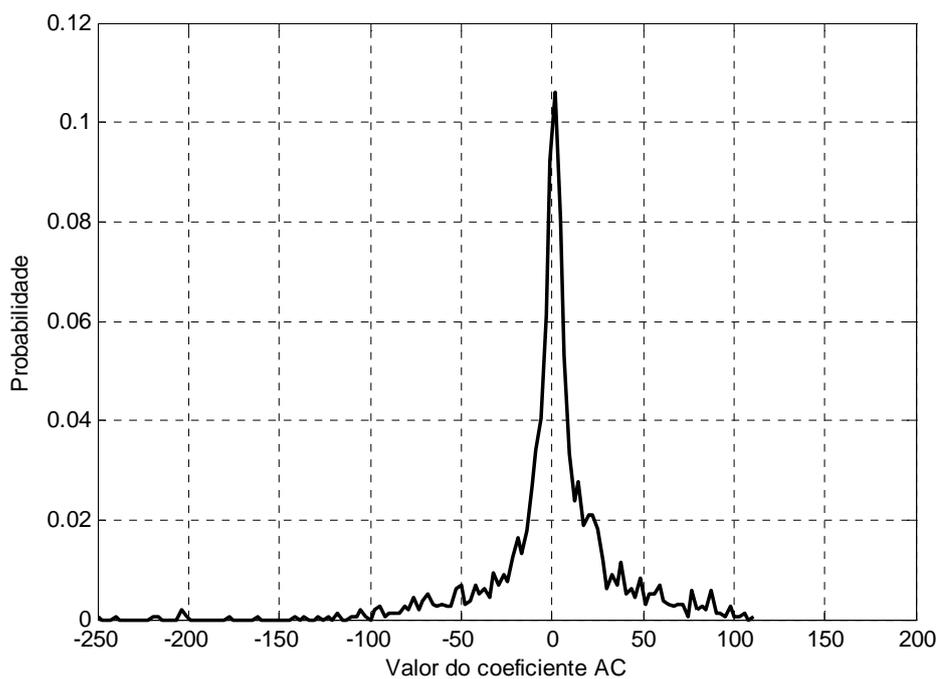


Figura 4.8: Distribuição dos coeficientes DCT para a banda AC (b_2) de mais baixa frequência espacial da sequência *Foreman*, QCIF.

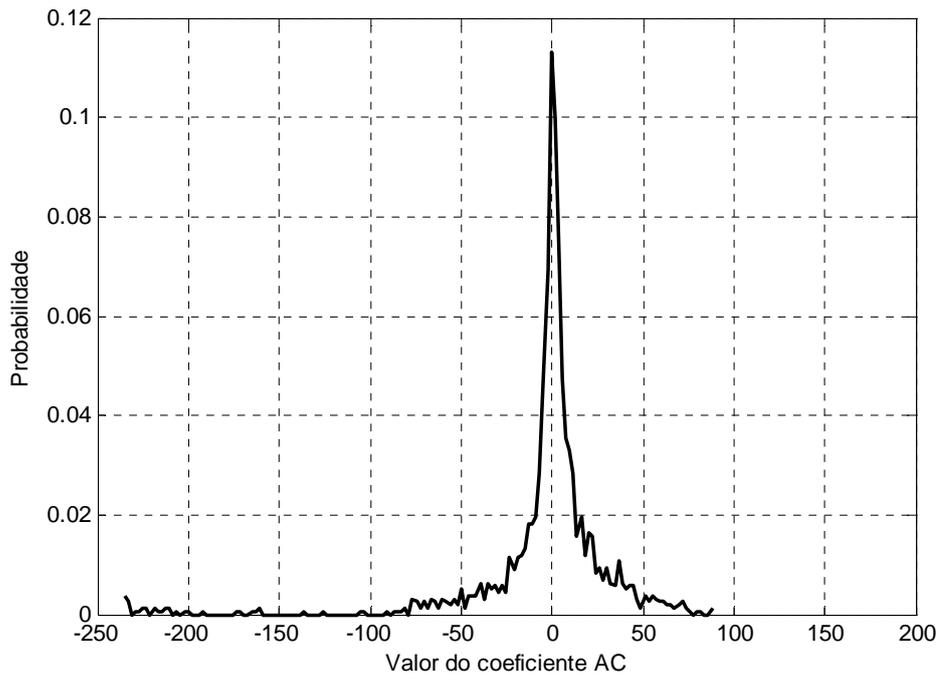


Figura 4.9: Distribuição dos coeficientes DCT para a banda AC (b_2) de mais baixa frequência espacial da sequência *News*, QCIF.

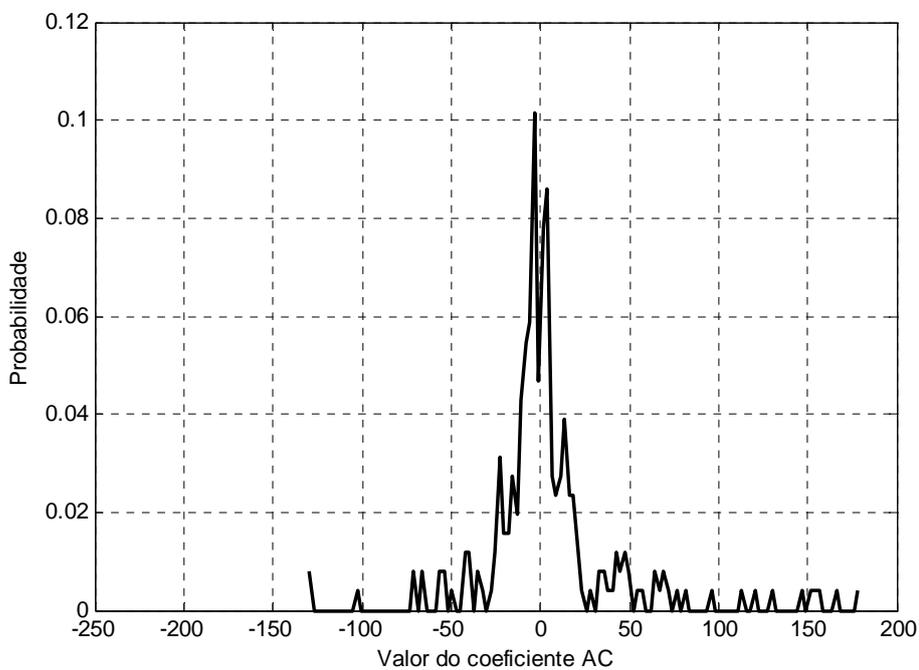


Figura 4.10: Distribuição dos coeficientes DCT para a banda AC (b_2) de mais baixa frequência espacial da sequência *Coastguard*, QCIF.

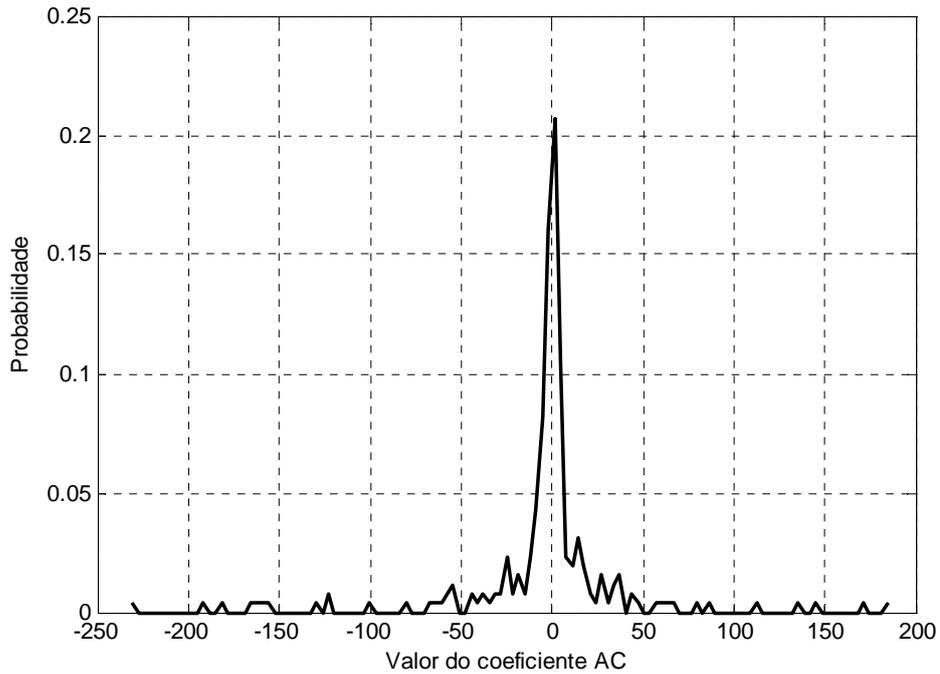


Figura 4.11: Distribuição dos coeficientes DCT para a banda AC (b_2) de mais baixa frequência espacial da sequência *Soccer*, QCIF.

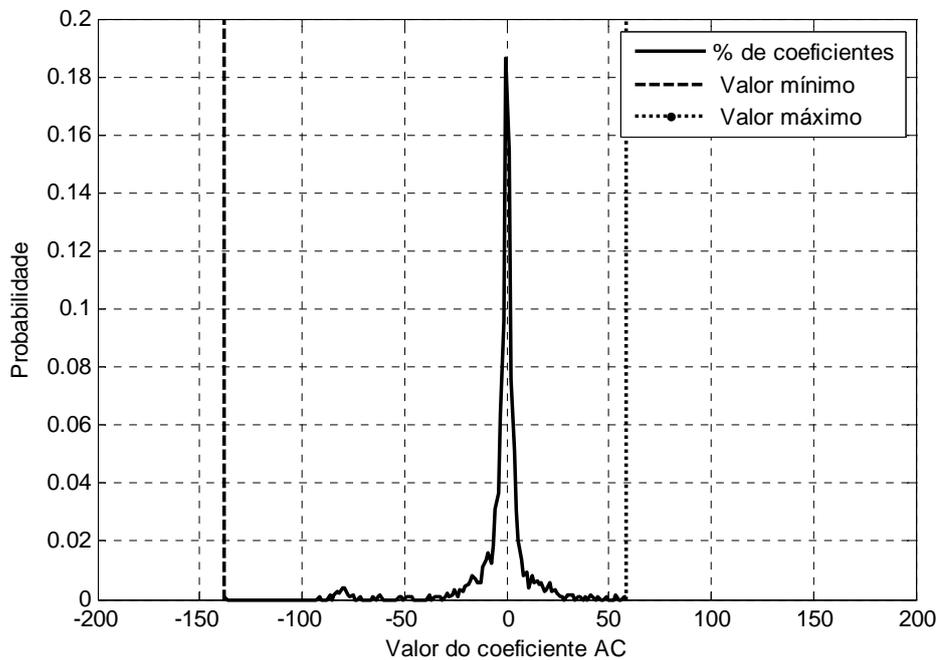


Figura 4.12: Distribuição dos coeficientes DCT para a banda AC (b_3) de mais alta frequência espacial da sequência *Foreman* QCIF.

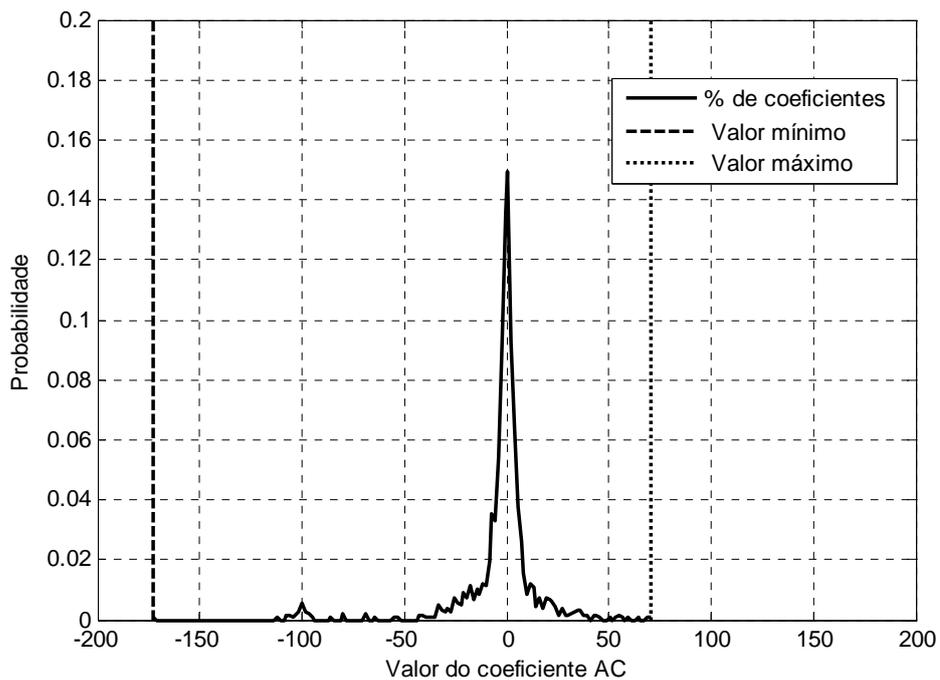


Figura 4.13: Distribuição dos coeficientes DCT para a banda AC (b_3) de mais alta frequência espacial da sequência *News*, QCIF.

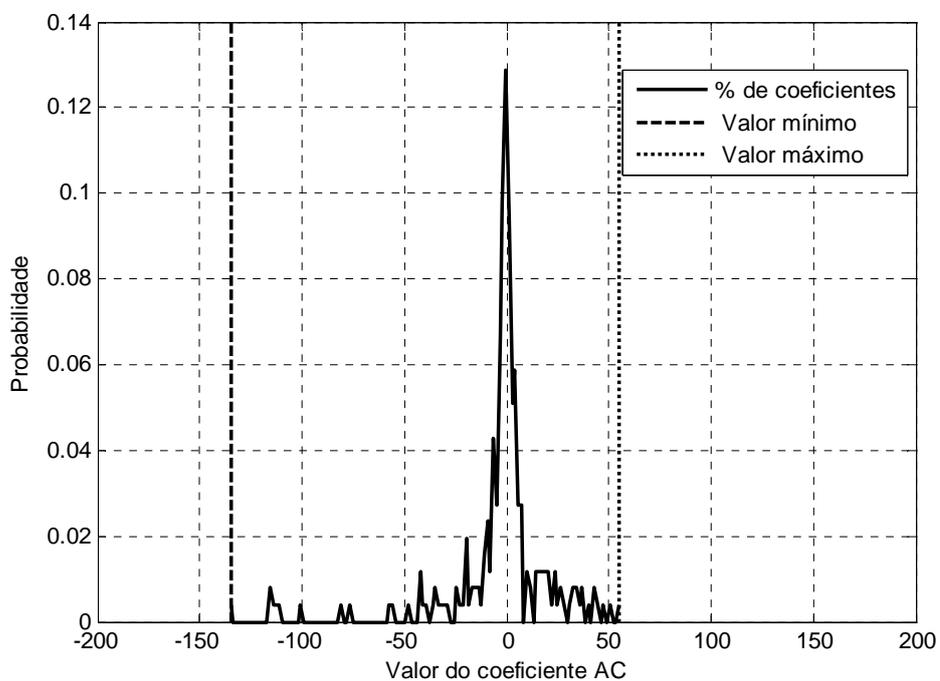


Figura 4.14: Distribuição dos coeficientes DCT para a banda AC (b_3) de mais alta frequência espacial da sequência *Coastguard*, QCIF.

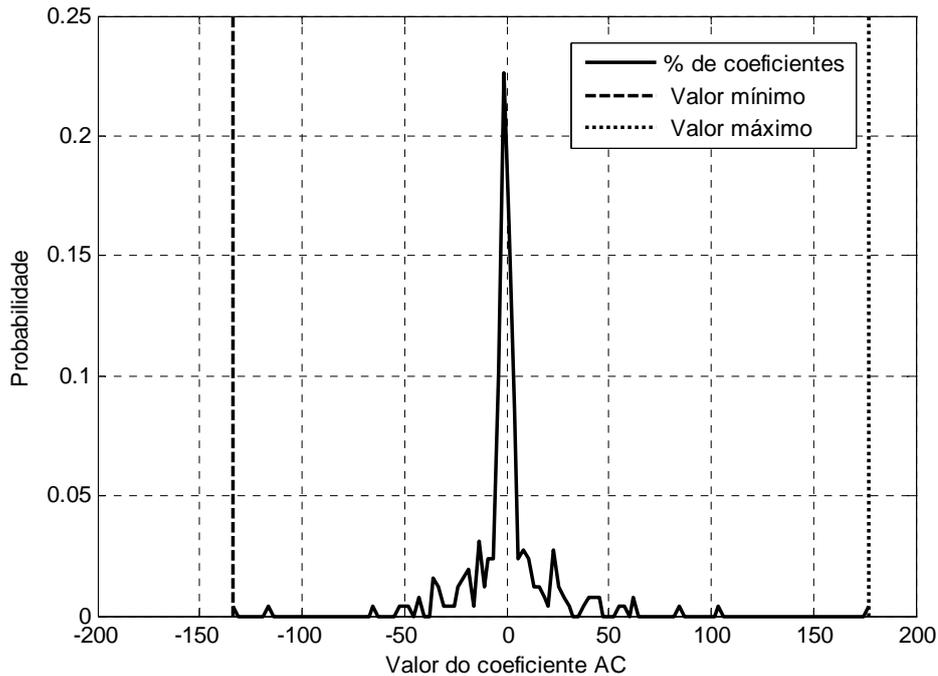


Figura 4.15: Distribuição dos coeficientes DCT para a banda AC (b_3) de mais alta frequência espacial da sequência *Soccer*, QCIF.

As bandas que possuem a mesma frequência espacial, tais como $\{b_2, b_5\}$, $\{b_3, b_6, b_9\}$, $\{b_4, b_7, b_{10}, b_{13}\}$, $\{b_8, b_{12}, b_{15}\}$, ou melhor, todas as bandas no bloco 4×4 que tem o mesmo valor da soma $i + j$, onde $i =$ índice da linha onde está o coeficiente e $j =$ índice da coluna do mesmo, apresentam uma distribuição aproximadamente semelhante de coeficientes.

Usando um quantizador similar ao apresentado na figura 4.16, no qual os índices de quantização não são simétricos em relação ao valor zero dos coeficientes, valores positivos de coeficientes DCT são mapeados para intervalos de quantização denominados com índices maiores ou iguais a zero e valores negativos de coeficientes DCT são mapeados para intervalos de quantização associados a índices negativos.

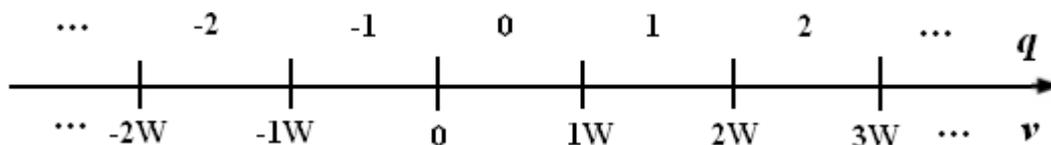


Figura 4.16: Quantizador escalar uniforme sem um intervalo de quantização simétrico em torno da amplitude zero.

Conforme mencionado na seção 4.3, Y_{2i}^{DCT} é considerada uma boa estimativa de X_{2i}^{DCT} , presente no decodificador. Entretanto, sabendo que Y_{2i}^{DCT} é uma estimativa de X_{2i}^{DCT} que não foi calculada a partir do frame original, pode haver alguns erros entre os coeficientes correspondentes de Y_{2i}^{DCT} e X_{2i}^{DCT} .

Se esses erros não forem corrigidos através de uma operação iterativa de decodificação baseada no código LDPC adaptativo proposto, efeitos irritantes de artefatos de blocos tornam-se visíveis no frame decodificado \tilde{X}_{2i} . De fato, o impacto dos erros entre os coeficientes de Y_{2i}^{DCT} e os correspondentes de X_{2i}^{DCT} pode ser mais acentuado para os coeficientes DCT em torno da amplitude zero. Nesta região, pode acontecer que um dado coeficiente Y_{2i}^{DCT} e o correspondente em X_{2i}^{DCT} tenham diferentes sinais, conforme ilustrado pela figura 4.17. Nesse cenário, o coeficiente Y_{2i}^{DCT} é mapeado para um intervalo de quantização -1 e o coeficiente correspondente em X_{2i}^{DCT} é mapeado para o intervalo de quantização 0. Porém, após a decodificação LIA, provavelmente o intervalo de quantização decodificado -1 será corrigido pela etapa de verificação de erros do código e caso haja algum efeito mínimo de artefato de bloco, este dificilmente será notável no frame decodificado.

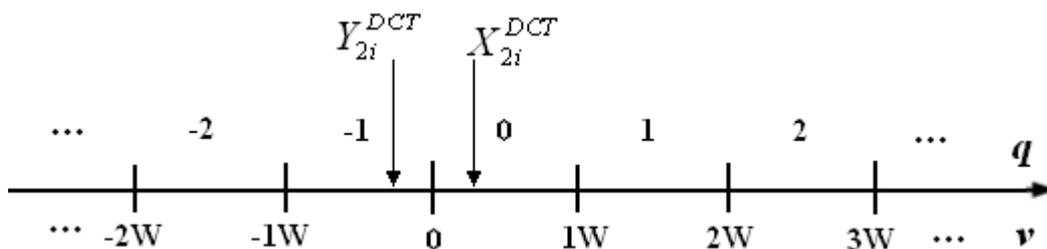


Figura 4.17: problema de quantização dos coeficientes AC usando um quantizador sem um intervalo de quantização simétrico em torno da amplitude zero.

A fim de reduzir as chances de haver pequenos artefatos na imagem reconstruída, alguns autores como Ascenso et al. [72] acharam mais adequado utilizar um quantizador com um intervalo de quantização simétrico em torno do zero, como mostra a figura 4.18. Neste caso, os valores dos coeficientes DCT em torno de zero são, agora, quantizados sob o mesmo índice de intervalo de quantização (independentemente do seu sinal), evitando erros entre os símbolos quantizados de Y_{2i}^{DCT} e os correspondentes em X_{2i}^{DCT} , e, conseqüentemente, reduzindo um pouco o efeito dos artefatos de bloco. Porém, esta configuração de

intervalos de quantização implica em um passo de quantização com maior tamanho, pois a faixa de quantização é a mesma, mas o número de níveis de quantização diminuiu, o que implica em um aumento do tamanho do passo de quantização e uma pequena perda de precisão no processo.

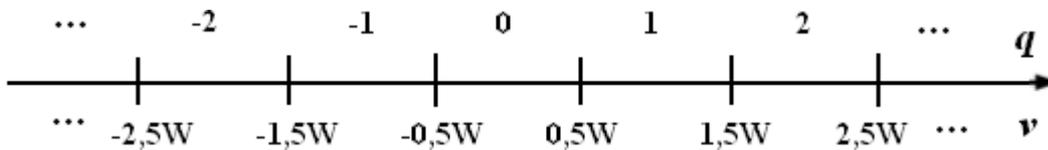


Figura 4.18: Quantizador escalar uniforme com um intervalo de quantização simétrico em torno da amplitude zero.

Para o caso ilustrado na figura 4.18, onde todos os intervalos de quantização têm a mesma largura W e com o símbolo zero abrangendo a faixa central de quantização, os limites dos intervalos de quantização de um quantizador uniforme podem ser matematicamente descritos pela eq. (4.8):

$$I_q = \begin{cases} [(q-0,5)W; (q+0,5)W] & q < 0 \text{ ou } q > 0 \\ [-0,5W; 0,5W] & q = 0 \end{cases} \quad (4.8)$$

onde I_q representa os limites do intervalo de quantização, q o índice do intervalo de quantização e W a largura do intervalo de quantização. O parâmetro W também é conhecido como o tamanho do passo de quantização.

4.5.3. Número de Níveis de Quantização

O módulo da amplitude dos coeficientes AC, dentro de um bloco de coeficientes DCT 4x4, na maioria das vezes tende a ser maior para os coeficientes próximos ao coeficiente DC e tende a diminuir para os coeficientes localizados em posições de frequências espaciais maiores [86]. Em termos de bandas de coeficientes AC, isto significa que as bandas AC, cuja soma dos índices da linha e da coluna ($i + j$) em cada bloco é baixa (mais perto de 2) agregam valores de amplitudes mais altas comparadas às bandas AC designadas com índices distantes da banda DC. De fato, dentro de um bloco de coeficientes DCT 4x4, as frequências espaciais menores englobam uma informação mais relevante sobre o

bloco do que as altas frequências, as quais geralmente correspondem a detalhes menos importantes para o sistema visual humano (HVS) [85].

Visto que o HVS é mais sensível a baixas frequências espaciais [86], os coeficientes DCT representando as frequências mais baixas são quantizados utilizando tamanhos menores de passo de quantização, isto é, com um número bem maior de intervalos de quantização (maior número de níveis), já que além dos passos de quantização menores, nas frequências mais baixas a variação ou *range* dos coeficientes ($v_{k,máx} - v_{k,min}$) é bem maior que em altas frequências. Nas frequências espaciais mais altas, os coeficientes são quantizados de forma mais grosseira (ou menos precisa), ou seja, com menos níveis de quantização, sem diminuir significativamente a qualidade visual da imagem reconstruída, visto que esses coeficientes têm pouca influência na construção da imagem. A escolha do número de níveis de quantização associado a cada banda de coeficientes DCT é uma forma importante de explorar a sensibilidade visual humana para frequências mais baixas, quando comparada a frequências mais altas.

4.5.4. Abordagem sobre a Faixa Dinâmica

Tanto o codificador quanto o decodificador possuem tabelas de quantização padronizadas, que serão descritas na seção 4.5.7. Cada uma com diferentes níveis de quantização, de acordo com a banda. Em todas as tabelas, o número de níveis de quantização é inversamente proporcional à frequência espacial associada à banda. Assim, antes de iniciar a codificação, configura-se o codificador e o decodificador para trabalharem com a mesma tabela-padrão de quantização, ou seja, sincronizando o número de níveis de quantização para cada banda.

Já que o número de níveis de quantização por banda de coeficientes DCT (no método de codificação WZ proposto) é conhecido pelo codificador e decodificador, para codificar os coeficientes de uma certa banda b_k , é necessário conhecer apenas sua faixa dinâmica, isto é, em que faixa de valores os coeficientes DCT variam, ou seja, deve-se conhecer o intervalo $[v_{k,min}, v_{k,máx}]$, e o número de níveis de quantização associado àquela banda. O tamanho do passo de quantização de cada banda, W_k , necessário para definir os limites dos intervalos

de quantização, pode ser calculado a partir destas duas quantidades, o que será mostrado nas seções 4.5.5 e 4.5.6.

O fato de que o codificador utiliza, para cada frame X_{2i} , uma faixa variável para cada banda de coeficientes DCT, ao invés de usar um valor fixo, permite ter larguras de intervalos de quantização ajustadas à faixa dinâmica de cada banda. Se forem utilizados valores fixos (pré-estabelecidos) para o tamanho do passo de quantização, ou melhor, tamanhos independentes dos valores dos coeficientes de cada banda, pode-se ter, como consequência, passos de quantização bem maiores que o necessário, caso os limites reais dos coeficientes (faixa dinâmica) sejam menores que os limites pré-fixados, o que implicaria numa menor precisão na quantização e pior qualidade da imagem reconstruída. Da mesma forma, se a faixa dinâmica real for maior que a pré-estabelecida, os coeficientes com valores próximos ao limite da faixa real seriam todos quantizados para um único valor bem abaixo do seu valor real, podendo resultar em uma distorção na imagem reconstruída.

Na prática, a faixa dinâmica pode ser um pouco menor do que a faixa fixa selecionada. Visto que o mesmo número de níveis de quantização é distribuído sobre uma faixa dinâmica mais curta, larguras menores de intervalos de quantização podem ser usadas. Quanto mais curto o tamanho do passo de quantização, menor é a distorção na imagem reconstruída pelo decodificador, conforme será explicado na seção 4.5.6. Assim, para cada frame Wyner-Ziv (X_{2i}), a faixa dinâmica de cada banda de coeficientes, determinada no estágio de quantização, é enviada ao decodificador a fim de ajudar no procedimento de reconstrução, e assumida como tendo sido recebida livre de erros (canal ideal), conforme será explicado na seção 4.5.6.

A faixa dinâmica é transmitida em um cabeçalho (*header*) com um comprimento máximo de 16 bits. Na verdade, são necessários no máximo 10 bits para representar a faixa dinâmica de cada banda DCT. Entretanto, sabendo que o tamanho do cabeçalho é tipicamente um múltiplo de 8, ele tem que ter um número inteiro de bytes. Este cabeçalho marca o início da transmissão dos bits de paridade de cada banda DCT. Assim, a transmissão da faixa dinâmica corresponde a um aumento máximo de 240 bits/frame, ou melhor, 15 bandas/frame para as quais o valor da faixa dinâmica é enviado vezes 16 bits/banda (devido a propósitos de empacotamento). Nenhum valor de faixa dinâmica é enviado para a banda b_{16} ,

pois esta possui menor frequência espacial e coeficientes de menor amplitude (insignificantes).

4.5.5. Cálculo do Passo de Quantização do Coeficiente DC

Cada banda b_k de coeficientes DCT é representada pelo vetor \underline{v}_k , o qual é dado por:

$$\underline{v}_k = \left[v_{k,1} \quad v_{k,2} \quad \dots \quad v_{k,n} \right] \quad (4.9)$$

Assim, \underline{v}_k é o vetor contendo todos os coeficientes DCT $v_{k,j}$ da banda b_k , com $j=1, 2, \dots, n$, onde n é o número de blocos N por N do frame, ou seja:

$$n = \frac{r \cdot c}{N^2} \quad (4.10)$$

sendo r e c o número de linhas e colunas do frame, respectivamente. Neste trabalho, adotou-se o padrão QCIF para o tamanho do frame, ou seja, $r = 144$ e $c = 176$, e uma transformada DCT 4x4, ou seja, $N = 4$; logo, há $n = 1584$ blocos 4x4 por frame.

O tamanho do passo de quantização de uma certa banda DCT b_k resulta do quociente entre a amplitude da faixa dinâmica da banda b_k , dada por $[v_{k,min}, v_{k,max}]$ e o número efetivo de níveis de quantização L_k , onde o valor mínimo possível da banda DC, $v_{1,min}$, é zero. O número de níveis de quantização L_k é igual a 2^{M_k} , sendo M_k o número de bits necessários para a codificação binária da banda b_k . O cálculo de do valor máximo da banda DC é dado por $v_{1,max} = máx\{v_1\}$ (neste trabalho). O método comumente utilizado para cálculo do limite superior do valor do coeficiente DC, segundo [87], é dado por:

$$N \cdot I_{máx} \quad (4.11)$$

onde N é a ordem da matriz (bloco) na qual é aplicada a DCT, e $I_{máx}$ representa a intensidade máxima atingida pelo pixel no frame X_{2i} , logo, $N.I_{máx}$ é a intensidade máxima do coeficiente DC. Assim, para um bloco de 4x4 pixels ($N = 4$) e para dados de vídeo com precisão de 8 bits para luminância (amostras com valores de 0 a 255), o limite superior da banda DC é 1020.

Assim, em [74, 87] o limite superior da banda DC é mantido fixo para todos os frames de vídeo, já que N e $I_{máx}$ na eq. (4.11) são constantes ao longo da sequência de vídeo, o que difere significativamente da realidade. Embora haja uma economia (na taxa) de 8 bits/frame (ver seção 4.5.4), no caso da banda DC isto não é vantajoso, porque o coeficiente DC é o mais significativo dentro do bloco DCT, e a diminuição da qualidade do frame reconstruído é significativa, conforme análise de experimentos apresentada na seção 4.5.8.

Além desta comprovação por meio de experimentos, observando as curvas de distribuição dos coeficientes DC das sequências apresentadas (figuras 4.4 a 4.7), nota-se que a faixa de variação dos valores dos coeficientes DC das duas sequências são similares, ou seja, situam-se aproximadamente entre 100 e 750, ou seja, valores bem distantes do valor máximo adotado pelos métodos mais modernos de quantização em DVC [54], pois para se obter, por exemplo, $v_{1,máx} = 1020$, precisaríamos ter um bloco com todos os valores de pixel iguais a 255, o que é muito difícil acontecer. Portanto, neste trabalho, o limite superior do valor do coeficiente DC será sempre calculado pela expressão $v_{1,máx} = máx\{y_1\}$. Assim, a variação do símbolo de quantização para a banda DC é dada por:

$$q_{1,j} \in [0, 2^{M_1} - 1] \quad (4.12)$$

Já para os outros coeficientes (AC), a opção de enviar somente um limite da faixa dinâmica (o valor máximo) e o outro ser igual ao negativo deste, não altera significativamente o desempenho, conforme experimentos realizados na seção 4.5.8, e por isso também será adotado pelo codec proposto. Maiores detalhes sobre o cálculo do passo de quantização AC serão dados na seção seguinte (4.5.6).

Como foi visto, o menor valor que um coeficiente DC poderia admitir em qualquer bloco é zero ($v_{1,min} = 0$), o que é assumido também por Kubasov et al. [74, 75] e por outros autores [87]. Pelo mesmo motivo relatado anteriormente, o codificador proposto disponibiliza a opção de enviar também o valor mínimo da banda DC, resultando em um pequeno acréscimo em desempenho. Desta forma, o passo de quantização da banda DC é dado por:

$$W_1 = \frac{v_{1,máx} - v_{1,min}}{2^{M_1}} \quad (4.13)$$

onde $v_{1,min}$ pode ser estabelecido como zero para fins de compressão, sem perda significativa da qualidade.

4.5.6. Cálculo do Passo de Quantização dos Coeficientes AC

Para obter o tamanho do passo de quantização (proposto neste trabalho) para as bandas AC b_k , sendo $k = 2, 3, \dots, 16$, primeiro são determinados os valores máximos $v_{k,máx}$ e mínimos $v_{k,min}$ dentro de cada banda b_k . Depois disso, o menor valor absoluto entre os dois corresponderá à metade da faixa dinâmica total da banda b_k , ou seja, a faixa dinâmica poderá ser definida como $[-v'_{k,máx}, v'_{k,máx}]$, onde o valor máximo enviado ao decodificador, $v'_{k,máx}$, é definido, neste trabalho, pela seguinte equação:

$$v'_{k,máx} = \min\{|v_{k,min}|, v_{k,máx}\} \quad (4.14)$$

O tipo de definição proposta neste trabalho (eq. (4.14)) tem a vantagem de diminuir a taxa de bits transmitidos, pois a faixa dinâmica de uma banda b_k fica reduzida a apenas um valor: $v'_{k,máx}$, e o decodificador já sabe que o limite inferior dessa faixa será $-v'_{k,máx}$. Além disso, repare que, utilizando a eq. (4.14) para determinar a faixa dinâmica, a distribuição de valores de coeficientes AC é “truncada”, resultando em uma redução do tamanho do passo de quantização e um

consequente aumento da precisão do processo. Além disso, observando as figuras 4.8 a 4.15, verifica-se que são pouquíssimos (menos de 0,1%) os coeficientes que se situam fora desta faixa “truncada”, o que também justifica a adoção do método, além dos resultados experimentais da seção 4.5.8.

Logo, sabendo-se que a amplitude da faixa dinâmica para certa banda b_k passa a ser dada por $2v'_{k,máx}$, o tamanho do passo de quantização W_k é obtido pela equação:

$$W_k = \frac{2.v'_{k,máx}}{2^{M_k}} = 2^{1-M_k} v'_{k,máx} \quad (4.15)$$

A expressão (4.15) acima difere das expressões utilizadas pelo estado da arte [74], pelo fato de o denominador da equação ser $2^{M_k} - 1$ em vez de 2^{M_k} , o que, como foi explicado na seção 4.5.2.2 e provado pelos experimentos da seção 4.5.8, não é mais eficiente do que o método adotado.

O j -ésimo índice de intervalo de quantização da banda b_k , representado por $q_{k,j}$, também conhecido como símbolo de quantização, resulta da fórmula:

$$q_{k,j} = \text{round} \left(\frac{v_{k,j}}{W_k} \right) \quad (4.16)$$

onde $v_{k,i}$ é o valor do coeficiente i da banda b_k e W_k é o tamanho do passo de quantização, anteriormente calculado, associado a esta banda. A função round significa que o argumento da mesma será arredondado para o inteiro mais próximo. Assim, qualquer j -ésimo símbolo de quantização da banda b_k AC, $q_{k,j}$, está compreendido entre os seguintes valores:

$$q_{k,j} \in \left[-\frac{2^{M_k}}{2}, \left(\frac{2^{M_k}}{2} - 1 \right) \right] = \left[-\left(2^{M_k-1} - 1 \right), \left(2^{M_k-1} - 1 \right) \right] \quad (4.17)$$

4.5.7.

Descrição do Método Proposto para Quantização

- I. Cada banda b_k de coeficientes DCT é quantizada utilizando um quantizador escalar uniforme com $L_k = 2^{M_k}$ níveis. O parâmetro M_k corresponde ao número de bits requerido para mapear cada coeficiente DCT da banda b_k para um dos 2^{M_k} níveis de quantização associados a esta banda.
- II. Visto que a qualidade do frame reconstruído e o número total de bits transmitidos por frame estão relacionados com o número de bits utilizado para codificar cada coeficiente DCT, para cada valor adotado para M_k há certo “ponto” ou “relação” taxa-distorção associado. Sendo assim, diferentes desempenhos podem ser alcançados variando-se o valor de M_k para a banda b_k . Na verdade, pode-se variar o ponto taxa-distorção do codec selecionando-se uma tabela de quantização com maior ou menor número de níveis de quantização. Essas contêm o número de níveis L_k que serão utilizados para quantizar os coeficientes de cada banda b_k .
- III. Caso seja utilizada uma tabela com maior número de níveis de quantização para cada banda, haverá uma maior precisão na representação dos coeficientes DCT e mais bits serão necessários para o mapeamento binário dos índices de quantização, o que implica em uma menor distorção da imagem (maior PSNR), porém, em virtude do aumento do número de bits necessários à codificação, haverá também um aumento do número de bits a serem transmitidos.
- IV. Para o codificador WZ-LIA, propõe-se 18 tabelas de quantização (matrizes de ordem 4), o que implica em 18 pontos taxa-distorção diferentes para a sequência de vídeo, ou seja, cada matriz da figura 4.19 está relacionada a um dado ponto taxa-distorção ou a uma certa qualidade e grau de compressão no desempenho do codificador WZ-LIA.
- V. A matriz da figura 4.19(a) representa a situação de taxa de bits mais baixa (e a maior distorção), em virtude de um menor número

de níveis de quantização (e, conseqüentemente, de bits), enquanto que a matriz 4.19(r) corresponde ao cenário da taxa de bits mais alta, proporcionando uma menor distorção ou uma melhor qualidade.

VI. É importante ressaltar que estas 18 matrizes ou tabelas de quantização foram propostas pelo autor deste trabalho, pois os autores em DVC, a priori, só utilizavam uma tabela de quantização e depois aumentaram para 8 em [68]. Para fins comparativos, essas 8 matrizes são apresentadas na figura 4.20.

VII. Comparando os dois conjuntos de matrizes, pode-se notar que o aumento do número de níveis de uma tabela para outra é muito mais suave no caso das tabelas propostas (figura 4.19). Por exemplo, observe que a mudança no número de níveis da tabela 7 (4.20g) para 8 (4.20h) é a mesma que da tabela 12 para 18 no esquema proposto (figuras 4.19(l) e 4.19(r), respectivamente).

16	4	0	0	16	8	0	0	32	8	0	0	32	8	4	0
4	0	0	0	8	0	0	0	8	0	0	0	8	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(a)	(b)	(c)	(d)												
32	16	4	0	32	16	8	0	32	16	8	4	32	16	8	4
16	4	0	0	16	8	0	0	16	8	4	0	16	8	4	4
4	0	0	0	8	0	0	0	8	4	0	0	8	4	4	0
0	0	0	0	0	0	0	0	4	0	0	0	4	4	0	0
(e)	(f)	(g)	(h)												
64	16	8	4	64	16	8	8	64	16	8	8	64	32	8	8
16	8	4	4	16	8	8	4	16	8	8	4	32	8	8	4
8	4	4	0	8	8	4	0	8	8	4	4	8	8	4	4
4	4	0	0	8	4	0	0	8	4	4	0	8	4	4	0
(i)	(j)	(k)	(l)												

64	32	16	8
32	16	8	4
16	8	4	4
8	4	4	0

(m)

128	32	16	8
32	16	8	4
16	8	4	4
8	4	4	0

(n)

128	64	16	8
64	16	8	4
16	8	4	4
8	4	4	0

(o)

128	64	32	8
64	32	8	4
32	8	4	4
8	4	4	0

(p)

128	64	32	16
64	32	16	4
32	16	4	4
16	4	4	0

(q)

128	64	32	16
64	32	16	8
32	16	8	4
16	8	4	0

(r)

Figura 4.19: 18 matrizes de quantização associadas a diferentes desempenhos taxa-distorção, onde cada elemento da mesma é o número de níveis de quantização da banda correspondente àquela posição.

16	8	0	0
8	0	0	0
0	0	0	0
0	0	0	0

(a)

32	8	0	0
8	0	0	0
0	0	0	0
0	0	0	0

(b)

32	8	4	0
8	4	0	0
4	0	0	0
0	0	0	0

(c)

32	16	8	4
16	8	4	0
8	4	0	0
4	0	0	0

(d)

32	16	8	4
16	8	4	4
8	4	4	0
4	4	0	0

(e)

64	16	8	8
16	8	8	4
8	8	4	4
8	4	4	0

(f)

64	32	16	8
32	16	8	4
16	8	4	4
8	4	4	0

(g)

128	64	32	16
64	32	16	8
32	16	8	4
16	8	4	0

(h)

Figura 4.20: 8 matrizes de quantização associadas a diferentes desempenhos taxa-distorção, onde cada elemento da mesma é o número de níveis de quantização da banda correspondente àquela posição.

VIII. Dentro de uma matriz de quantização 4×4 , o valor na posição k (onde $k = 1, 2, \dots, 16$) das figuras 4.19 e 4.20 indica o número de níveis de quantização L_k associado à banda b_k de coeficientes DCT. A relação dos índices da linha i e da coluna j com a posição k é descrita por:

$$k = 4(i - 1) + j \quad (4.18)$$

- IX. A numeração das bandas de coeficientes DCT, isto é, a ordem em que as bandas são codificadas, é apresentada na figura 4.2. As matrizes de quantização que aparecem na figura 4.19 são usadas para determinar o desempenho taxa-distorção do codec WZ proposto e são assumidas como conhecidas tanto pelo codificador quanto pelo decodificador.
- X. Nas matrizes da figura 4.19, o valor zero significa que nenhum bit será transmitido para o decodificador nas bandas correspondentes (bandas que contiverem os zeros). Neste caso, o decodificador repõe os coeficientes das bandas DCT, para as quais nenhum bit foi enviado, pelos coeficientes DCT correspondentes pertencentes às bandas da informação lateral.

4.5.8. Avaliação do Método de Quantização Proposto

Neste experimento, pretende-se avaliar e comparar a eficiência dos diversos métodos propostos neste trabalho para quantização variável em relação ao método de quantização adotado pelos padrões de codificação de vídeo distribuído, abreviado aqui como Quantização-dvc. Os métodos comparados possuem as seguintes diferenças em relação à quantização:

O cálculo do coeficiente DC para cada método é realizado segundo a tabela 4.1 abaixo. Os métodos que se baseiam no cálculo convencional do DVC iniciam sua denominação com o prefixo “dvc”.

Tabela 4.1: métodos propostos para cálculo da faixa dinâmica do coeficiente DC

Método de Quantização	Cálculo do Vmáx-DC	Cálculo do Vmín-DC
dvc	255*N	0
Proposto_QP_reduzido	255*N	0
Proposto_Imáx_QP_reduzido	máx{WZ}	0
Normal	máx{WZ}	mín{WZ}

Para o método DVC e para os outros que foram propostos aqui e que são baseados no método de quantização dvc convencional, a quantização dos coeficientes AC é dada pelo algoritmo 4.1 abaixo.

Algoritmo 4.1: cálculo do passo de quantização W_k , baseado no método dvc

%L(k) é dado pela TQ (Tabela de Quantização)

Para k=2:N²-1

Se L(k)==0

W(k)=0;

Caso contrário

Vmax1(k)=max(X_dct(k,:));

Vmin1(k)=min(X_dct(k,:));

Se Vmax1(k)>abs(Vmin1(k))

Vmin(k)=-Vmax(k);

Caso contrário

Vmax(k)=abs(Vmin1(k));

fim

W(k)=2*Vmax(k)/(L(k)-1);

fim

fim

Na tabela 4.1 acima, o método em que houver a nomenclatura “QPreduzido” significa que o cálculo do passo de quantização AC ($W(k)$ no algoritmo) é calculado como:

$$W(k) = 2 * V_{\max}(k) / L(k);$$

Assim, nesta fórmula proposta o divisor da equação é uma unidade maior, resultando em uma pequena diminuição no passo de quantização $W(k)$, o que implica em um pequeno acréscimo na precisão do quantizador, conseqüentemente, na qualidade do frame reconstruído. Porém, há a possibilidade de ocorrer pequenos erros na reconstrução dos coeficientes quantizados, conforme descrição do problema na seção 4.5.6. Por isso, buscou-se avaliar os métodos DVC utilizando-se fórmulas e algoritmos diferentes tanto para as bandas DC quanto AC.

O algoritmo 4.2 a seguir expressa a quantização dos coeficientes AC utilizada pelos métodos de quantização propostos neste trabalho.

Algoritmo 4.2: cálculo do passo de quantização W_k , baseado no método proposto

```
%L(k) é dado pela TQ
Para k=2:N²-1
  Se L(k)=0
    W(k)=0;
  Caso contrário
    Vmax(k)=max(X_dct(k,:));
    Vmin(k)=min(X_dct(k,:));
    Se abs(Vmin(k))<Vmax(k)
      Vmax(k)=abs(Vmin(k));
    senão
      Vmin(k)=-Vmax(k);
    fim
  W(k)=2*Vmax(k)/(L(k)-1);
fim
```

A diferença do algoritmo 4.1 para o 4.2 é o cálculo dos valores mínimos e máximos da banda AC a serem enviados ao decodificador, pois no algoritmo convencional utilizado em dvc o valor máximo é dado por $\max\{|\underline{v}_{k,min}|, \underline{v}_{k,max}\}$.

Desta forma, sabendo-se que, conforme explicação apresentada na seção 4.5.6, o valor mínimo é o negativo do valor máximo, a faixa de variação de valores de coeficientes DCT é expandida. Já no caso do algoritmo proposto, o valor máximo é dado por $\min\{|\underline{v}_{k,min}|, \underline{v}_{k,max}\}$, o que resulta em uma faixa de variação de valores de coeficientes DCT truncada, ou seja, diminuída, a fim de diminuir também o tamanho do passo de quantização e assim, aumentar a precisão do processo. No método “Quant-Normal” enviam-se os valores máximos e mínimos reais de cada banda para o decodificador e não somente um deles, estabelecendo-se, portanto, um limite superior para a PSNR da configuração adotada.

Observando as curvas apresentadas nas figuras 4.21 a 4.28, para as quatro sequências utilizadas neste experimento, nota-se que todos os métodos de quantização que adotaram o envio do valor máximo da banda b_l , “Imáx”, obtiveram um desempenho bem melhor que os demais, tendo que enviar somente mais um parâmetro por frame para o decodificador. Além disso, os dois métodos que utilizaram o algoritmo proposto para quantização do coeficiente AC, identificado como “QPreduzido” nos mesmos, alcançaram melhores resultados em termos de eficiência do que o método convencional, “dvc”. Logo, o método que

obteve melhor desempenho foi o que fez uso das técnicas propostas (das técnicas bem sucedidas para quantização dos coeficientes AC e do DC), o qual foi chamado de “Quantização-Imax-QPrezuido”.

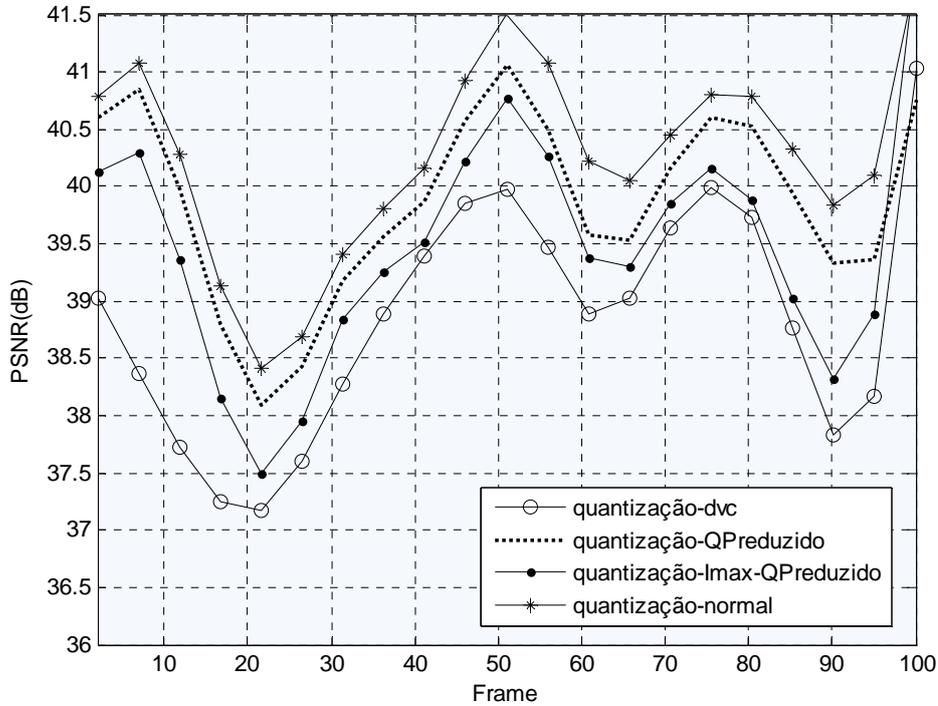


Figura 4.21: PSNR do frame Wyner-Ziv utilizando diferentes técnicas de quantização, para sequência *Foreman*, com QP = 26 (intra) e TQ = 18 (nível de qualidade).

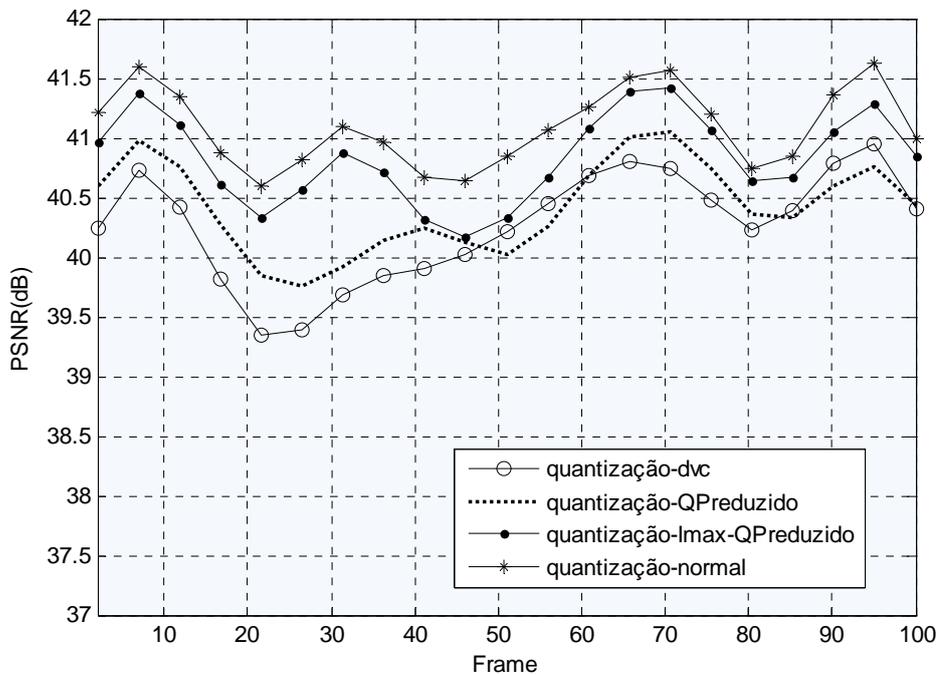


Figura 4.22: PSNR do frame Wyner-Ziv utilizando diferentes técnicas de quantização, para sequência *News*, com QP = 26 (intra) e TQ = 18 (nível de qualidade).

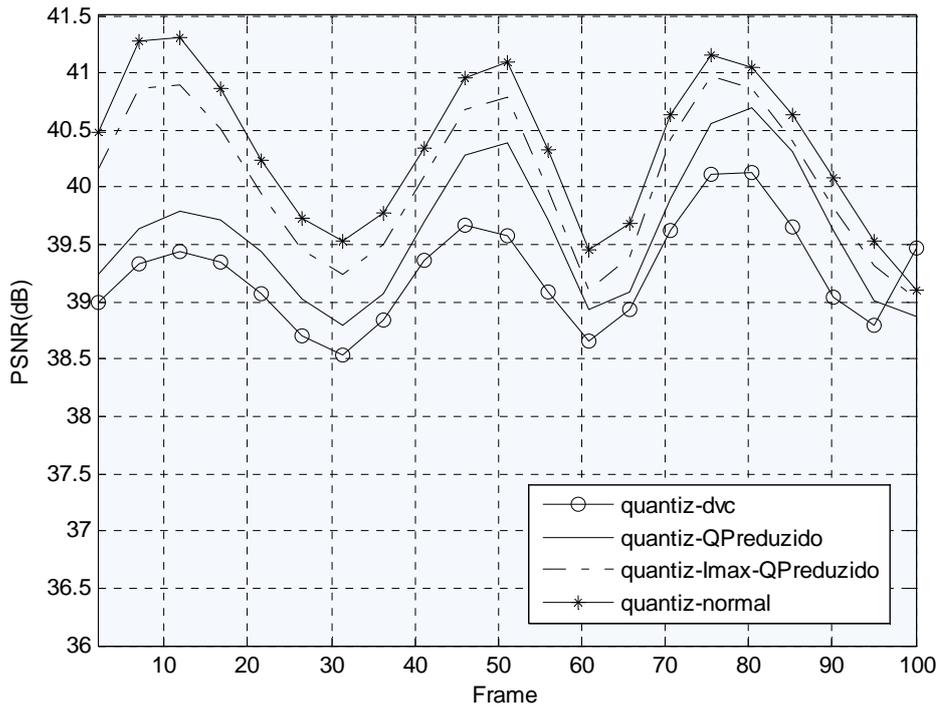


Figura 4.23: PSNR do frame Wyner-Ziv utilizando diferentes técnicas de quantização, para sequência *Coastguard*, com QP = 26 (intra) e TQ = 18 (nível de qualidade).

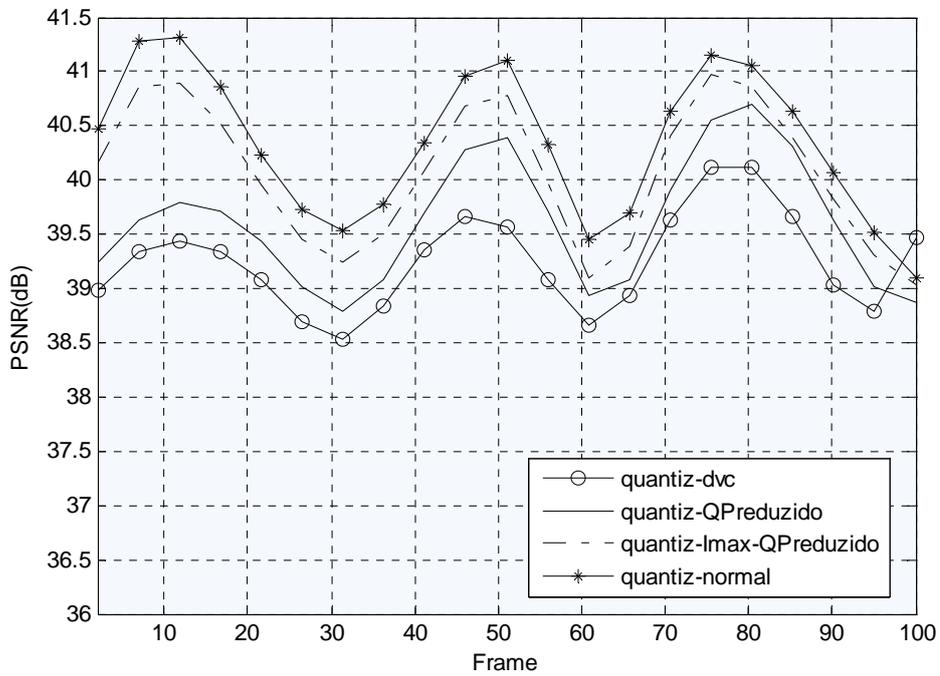


Figura 4.24: PSNR do frame Wyner-Ziv utilizando diferentes técnicas de quantização, para sequência *Soccer*, com QP = 26 (intra) e TQ = 18 (nível de qualidade).

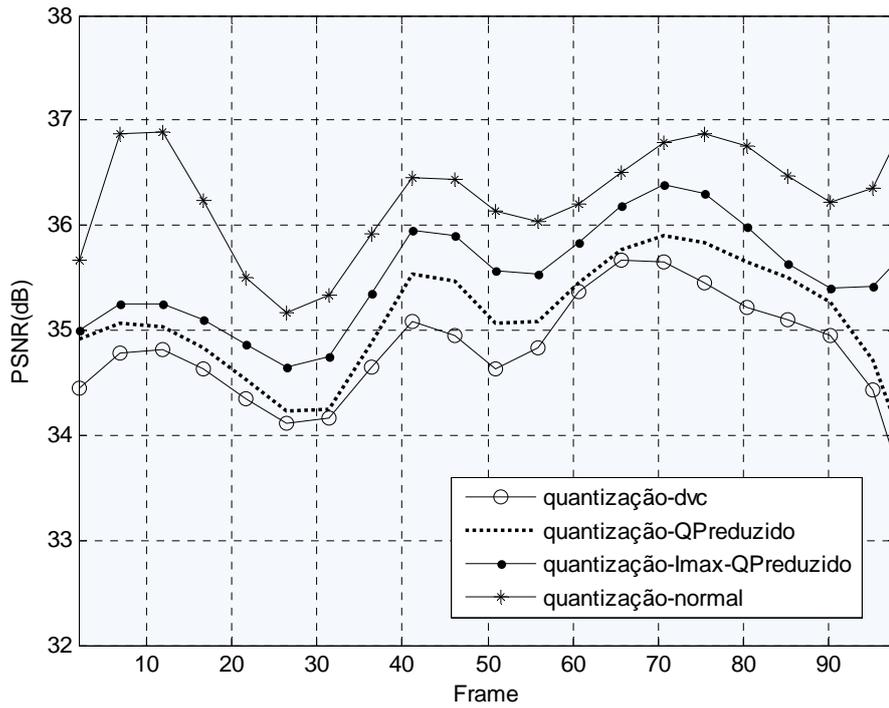


Figura 4.25: PSNR do frame Wyner-Ziv utilizando diferentes técnicas de quantização, para sequência *Foreman*, com $QP = 36$ (intra) e $TQ = 7$ (nível de qualidade).

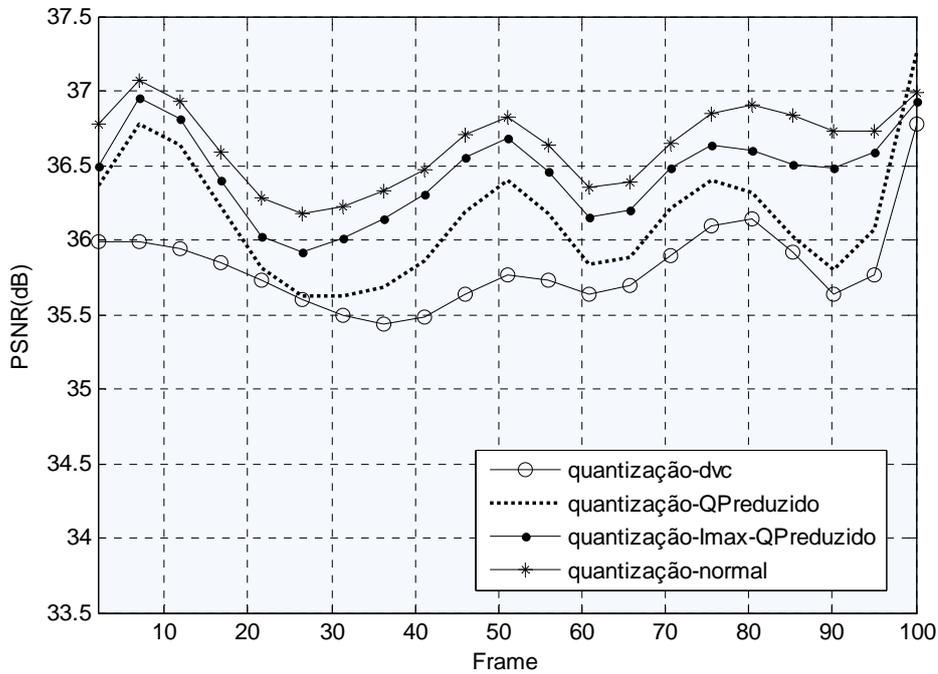


Figura 4.26: PSNR do frame Wyner-Ziv utilizando diferentes técnicas de quantização, para sequência *News*, com $QP = 36$ (intra) e $TQ = 7$ (nível de qualidade).

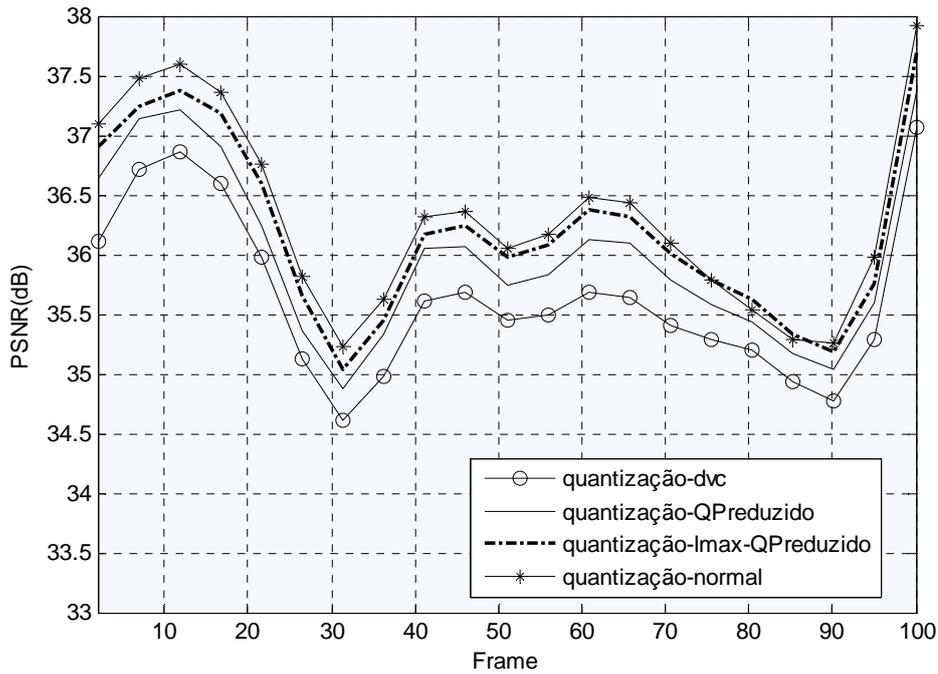


Figura 4.27: PSNR do frame Wyner-Ziv utilizando diferentes técnicas de quantização, para sequência *Coastguard*, com QP = 36 (intra) e TQ = 7 (nível de qualidade).

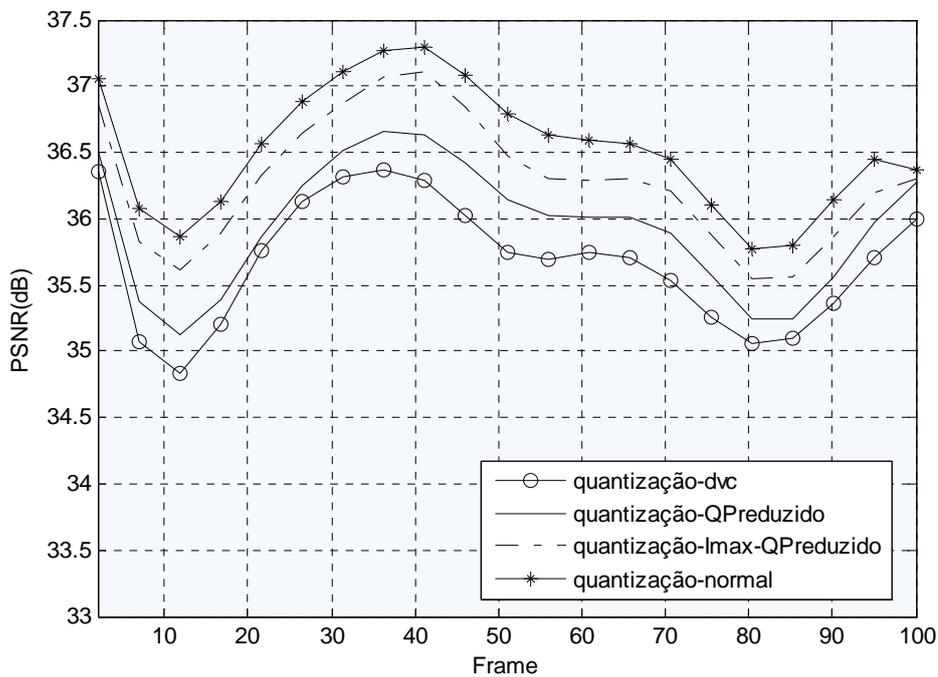


Figura 4.28: PSNR do frame Wyner-Ziv utilizando diferentes técnicas de quantização, para sequência *Soccer*, com QP = 36 (intra) e TQ = 7 (nível de qualidade).

4.5.9. Procedimento de Geração dos *Bitplanes*

Após quantizar a banda b_k , os símbolos quantizados (representados por valores inteiros) são convertidos em um *stream* binário. Em cada banda b_k , os bits de mesma significância são agrupados formando planos de bits ou *bitplanes*. Visto que os símbolos quantizados pertencentes à banda b_k são codificados com M_k bits, conseqüentemente haverá M_k *bitplanes* associados à banda b_k . O *bitplane* 1 da k -ésima banda, $\underline{P}_{k,1}$, é o conjunto (ou vetor) formado pelos bits menos significativos (LSB – *Least Significant Bits*) e o *bitplane* M_k , \underline{P}_{k,M_k} , que corresponde ao último plano de bits, é o conjunto formado pelos bits mais significativos (MSB – *Most Significant Bits*), conforme observa-se na tabela 4.2.

Tabela 4.2: composição dos *bitplanes* da k -ésima banda.

Bitplane		Q ₁	Q ₂	Q ₃	...	Q _n
1 (LSB)	$\underline{P}_{k,1}$	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$...	$d_{1,n}$
2	$\underline{P}_{k,2}$	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$...	$d_{2,n}$
⋮	⋮	⋮	⋮	⋮	...	⋮
M_k (MSB)	\underline{P}_{k,M_k}	$d_{M_k,1}$	$d_{M_k,2}$	$d_{M_k,3}$...	$d_{M_k,n}$

onde $d_{i,j} \in \{0,1\}$. Assim, o quadro total de bits da banda b_k , \underline{P}_k , contém todos os vetores *bitplanes*:

$$\underline{P}_k = \left[\underline{P}_{k,1} \quad \underline{P}_{k,2} \quad \cdots \quad \underline{P}_{k,M_k} \right]^T \quad (4.19)$$

sendo \underline{P}_{k,M_k} o último *bitplane* (M_k) da k -ésima banda. Cada *bitplane* i da banda b_k consiste em um vetor de comprimento n , cujos elementos são bits:

$$\underline{P}_{k,i} = \left[d_{i,1} \quad d_{i,2} \quad \cdots \quad d_{i,n} \right] \quad (4.20)$$

4.5.9.1. Avaliação Objetiva da Transmissão Parcial dos *Bitplanes*

O objetivo deste experimento é avaliar a perda em termos de PSNR da qualidade do frame WZ reconstruído, codificando e, conseqüentemente, transmitindo apenas parte do número total de *bitplanes* de cada banda, ou melhor,

transmitindo-se somente os bitplanes mais significativos. As figuras 4.29 e 4.30 apresentam a PSNR do frame WZ ao longo da sequência de vídeo para os casos em que: (i) transmitem-se todos os M_k bitplanes; (ii) transmitem-se todos menos um; (iii) todos menos dois; (iv) e todos menos três bitplanes. A tabela 4.3 mostra o número total absoluto e relativo de bits transmitidos para cada um dos casos descritos anteriormente.

Pelos resultados apresentados nas figuras 4.29 e 4.30, observa-se que quando se transmite um bitplane a menos não há uma perda significativa da qualidade e até quando se transmite dois bitplanes a menos, a qualidade ainda é aceitável, embora sejam transmitidos somente 76% e 52% dos bits, respectivamente, que seriam transmitidos enviando-se todos os bitplanes, segundo a tabela 4.3.

Tabela 4.3: número total de bits transmitidos por banda, conforme o número de bitplanes codificados.

Banda	Num_Níveis	Bitplanes Transmitidos por Banda			
		Mk	Mk-1	Mk-2	Mk-3
1	128	7	6	5	4
2	64	6	5	4	3
3	32	5	4	3	2
4	16	4	3	2	1
5	64	6	5	4	3
6	32	5	4	3	2
7	16	4	3	2	1
8	8	3	2	1	0
9	32	5	4	3	2
10	16	4	3	2	1
11	8	3	2	1	0
12	4	2	1	0	0
13	16	4	3	2	1
14	8	3	2	1	0
15	4	2	1	0	0
SOMA =		63	48	33	20
Bits transmitidos =		99.792	76.032	52.272	31.680
% Transmitido =		100%	76%	52%	32%

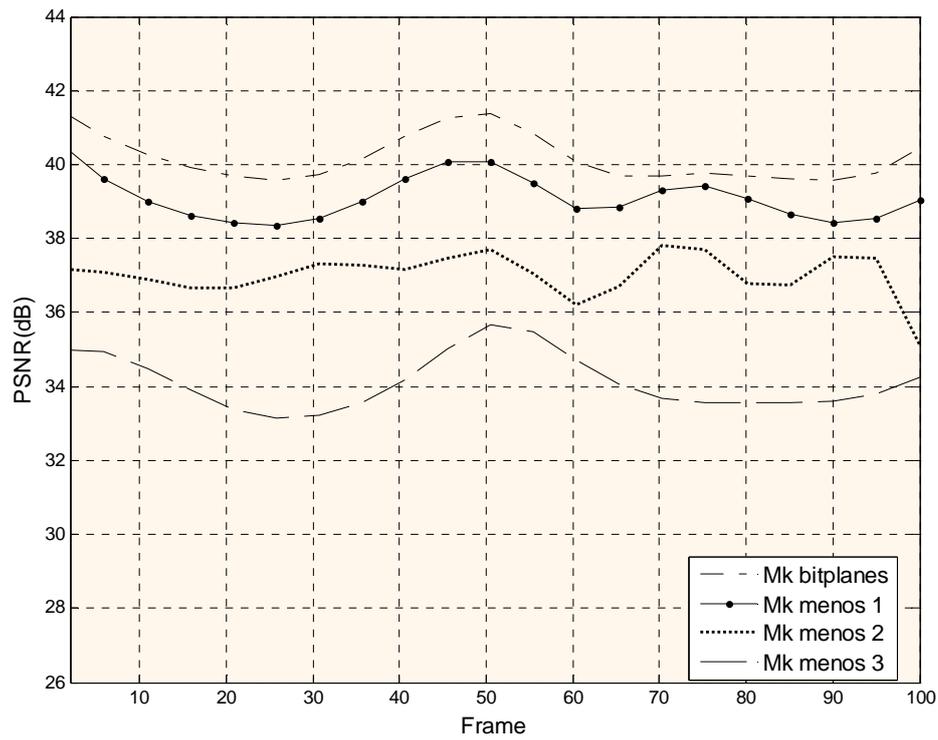


Figura 4.29: PSNR do frame Wyner-Ziv codificando-se diferentes quantidades de bitplanes, para seqüência *Foreman*, com QP = 27 (intra) e TQ = 18 (nível de qualidade).

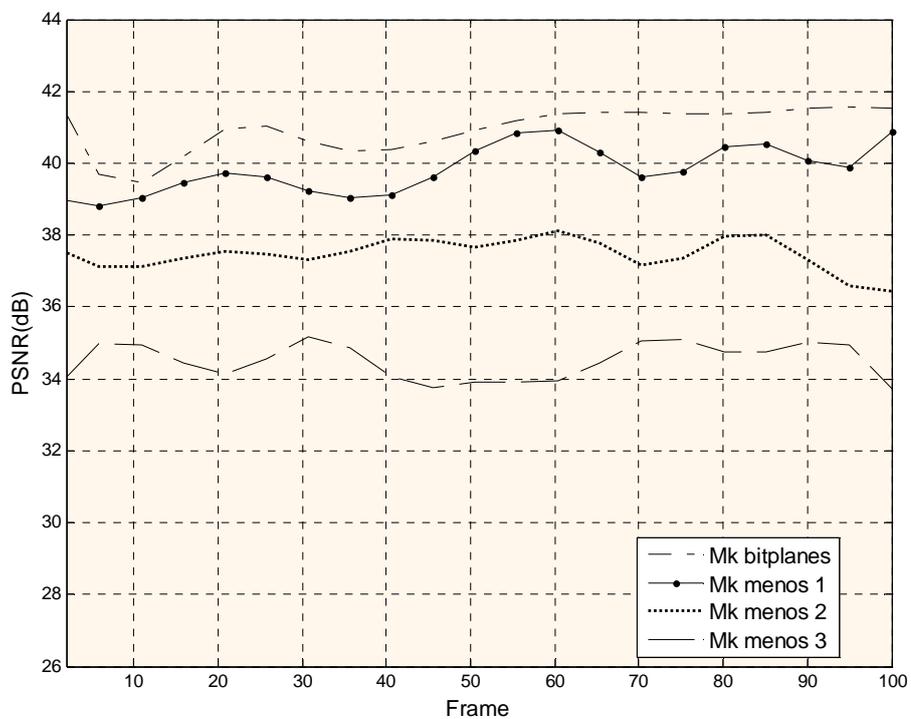


Figura 4.30: PSNR do frame Wyner-Ziv codificando-se diferentes quantidades de bitplanes, para seqüência *News*, com QP = 27 (intra) e TQ = 18 (nível de qualidade).

4.5.9.2. Avaliação Subjetiva da Transmissão Parcial dos Bitplanes

O objetivo deste experimento é avaliar a perda subjetiva da qualidade do frame WZ reconstruído, transmitindo-se apenas parte do número total de *bitplanes* de cada banda. As figuras 4.31 e 4.32 apresentam um frame WZ reconstruído, onde a figura (a) mostra o frame original, a figura (b) mostra o frame WZ reconstruído transmitindo-se todos os bitplanes, na figura (c) exclui-se o *bitplane* mais insignificante, na figura (d) excluem-se os dois bitplanes mais insignificantes e na figura (e) excluem-se os três *bitplanes* mais insignificantes. É importante ressaltar que os bitplanes não codificados serão substituídos pelos correspondentes da informação lateral. Fazendo uma avaliação subjetiva (inspeção visual) das referidas figuras, nota-se que há pouca diferença na qualidade da imagem transmitindo-se até dois bitplanes a menos, tanto para a sequência *Foreman* quanto para a *News*, o que confirma o que foi dito na seção 4.5.9.1 anterior, a respeito da transmissão parcial dos *bitplanes*.

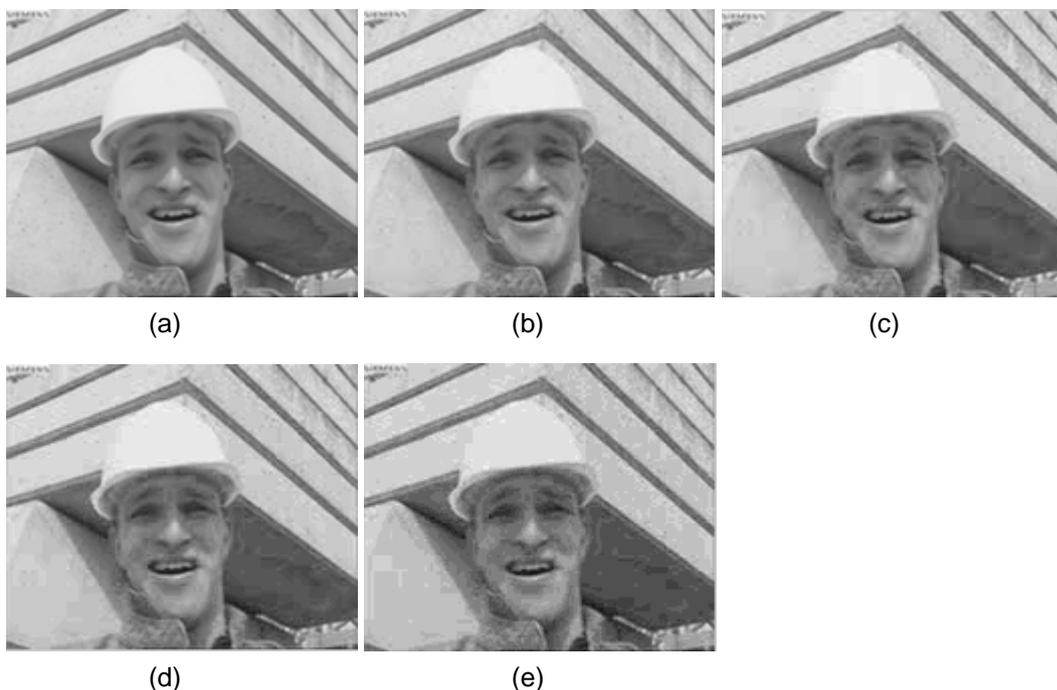


Figura 4.31: (a) frame WZ original da sequência *Foreman*; (b) frame WZ reconstruído considerando todos os bitplanes; (c) excluindo-se o bitplane menos significativo; (d) excluindo-se os dois bitplanes menos significativos; (e) excluindo-se os três bitplanes menos significativos.

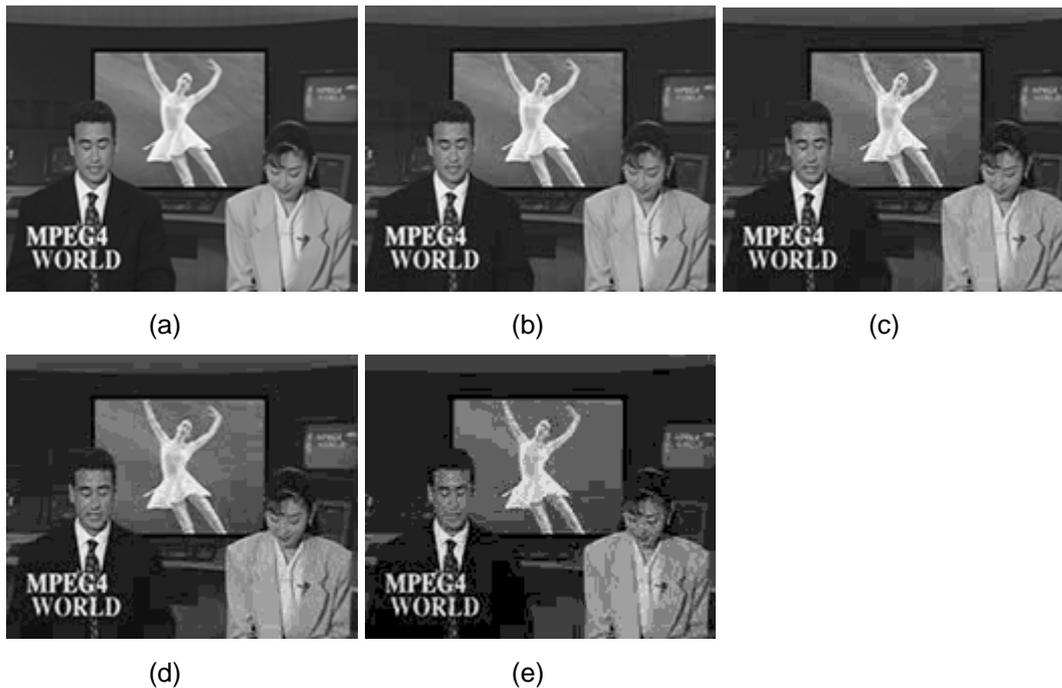


Figura 4.32: (a) frame WZ original da seqüência *News*; (b) frame WZ reconstruído considerando todos os bitplanes; (c) excluindo-se o bitplane menos significativo; (d) excluindo-se os dois bitplanes menos significativos; (e) excluindo-se os três bitplanes menos significativos.

4.5.10.

Procedimento de Geração das Síndromes

- O codificador de canal para DVC também é conhecido como codificador Slepian-Wolf, recebendo esse nome em virtude de alcançar uma maior compressão, para um mesmo nível de qualidade do frame reconstruído, por “contar” sempre com a presença de uma informação auxiliar (informação lateral) bastante relevante no decodificador, ou seja, informação essencial para correta decodificação do frame WZ. O codificador de canal proposto utiliza um tipo de código LDPC irregular e acumulado, compatível com a taxa, o qual foi denominado de LIA, para a codificação distribuída de fonte, tendo como base o código LDPC acumulado, introduzido em [71]. Os códigos LDPC acumulados (LDPCA) são os que melhor se aproximam de uma variedade de canais de comunicação, incluindo o canal virtual em DVC, quando comparados aos códigos turbo [71]. Quando se fala sobre

o “canal virtual” refere-se ao canal de correlação entre a informação lateral e o frame WZ.

- Um codificador LDPCA consiste em um código LDPC formador de síndrome concatenado com um acumulador. Para cada *bitplane*, bits de síndrome são criados utilizando o código LIA e um acumulador módulo-2 produz a síndrome acumulada.
- O procedimento de codificação LIA da banda b_k começa com a codificação do *bitplane* mais importante, que é aquele que corresponde aos bits mais significativos dos símbolos quantizados da banda b_k , ou seja, o processo se inicia com a codificação do *bitplane* M_k .
- Visto que cada banda b_k é codificada separadamente, a operação de decodificação LDPC acumulada é também realizada em nível de banda de coeficientes DCT, onde o vetor que alimenta o codificador LDPC são os *bitplanes* da banda b_k , ou seja, M_k vetores de comprimento n . O decodificador LIA começa decodificando o *bitplane* mais significativo da banda b_k . Para cada banda uma dessas bandas, o número de *bitplanes* a serem codificados depende do número de bits (M_k) necessários para mapear um valor de coeficiente DCT para um dos 2^{M_k} níveis de quantização associados à banda b_k .

4.6. Técnicas Propostas para Decodificação

4.6.1. Decodificação Iterativa LIA

- ✓ O decodificador LIA inicia seu processo decodificando banda por banda, de b_1 até b_{15} . A partir da síndrome (acumulada) recebida a_k , correspondente à banda b_k , ele decodifica primeiro o *bitplane* mais significativo, *bitplane* M_k , e depois continua o processo realizando decodificações condicionais, ou seja, decodifica o *bitplane* M_{k-1} dado o conhecimento dos bits de M_k (conjunto de bits mais significativos), depois decodifica (M_{k-2}), dado o conhecimento dos bits de M_k e (M_{k-1}) e assim por diante até

acabar de decodificar o *bitplane* 1 (menos significativo). A banda b_{16} não será transmitida, mas seus valores serão substituídos pelos correspondentes da informação lateral.

- ✓ O decodificador de canal executa um processo iterativo de decodificação, segundo a configuração do decodificador LDPC irregular acumulado, tendo como argumento as síndromes das palavras-código, a informação lateral e as estatísticas do canal de correlação.
- ✓ Terminando as iterações do decodificador de canal LIA, caso a palavra-código não seja reconhecida como parte do código, ou melhor, caso a relação entre o palavra-código decodificada \tilde{c} e a matriz de verificação de paridade H não for atendida, o que corresponde a $\tilde{c}.H^T \neq 0$, o decodificador LIA requer mais síndromes acumuladas, via canal de *feedback*, até que a probabilidade de erro do *bitplane* atual esteja abaixo de um dado limiar de probabilidade de erro. Quando isto ocorre, a operação de decodificação LIA do *bitplane* atual é considerada bem sucedida.
- ✓ Depois de decodificar com sucesso o *bitplane* mais significativo da banda b_k , o decodificador procede de forma análoga para os (M_k-1) *bitplanes* remanescentes associadas àquela banda.
- ✓ Uma vez que todos os *bitplanes* da banda b_k foram decodificados com êxito, o decodificador inicia a decodificação da banda b_{k+1} . Este procedimento é repetido até que todos os *bitplanes* de todas as bandas de coeficientes DCT, para as quais os bits Wyner-Ziv (bits de paridade) foram transmitidos, sejam decodificados.

4.6.2. Buffer e Transmissão Parcial das Síndromes

O codificador de canal Wyner-Ziv, chamado aqui de LIA, armazena as síndromes acumuladas em um *buffer* e inicialmente transmite poucas delas. Este grupo inicial de síndromes corresponde à taxa teórica mínima $R_{\text{mín}}$, que será discutida na próxima seção. Se o decodificador Wyner-Ziv falha na realização de

uma correta decodificação, mais síndromes acumuladas são solicitadas ao *buffer* do codificador através de um canal de *feedback*. Caso a operação de decodificação seja considerada bem sucedida, o *buffer* libera novas síndromes acumuladas, ou melhor, síndromes pertencentes a outros bitplanes. Além disso, para auxiliar o decodificador na detecção de erros residuais, Rebollo et al. [68] sugere que pode ser transmitido também um código CRC (*Cyclic Redundancy Check*) de 8 bits para cada bitplane codificado, porém, ele não foi incorporado ao codec deste trabalho, pois sua adição fornece pouquíssimo ganho (quase desprezível) em termos de eficiência, além de ter que gastar mais 8 bits para cada bitplane (verificação em [21–25]), o que significa transmitir 504 bits a mais por frame para o caso de utilizarmos a tabela de quantização 18 (figura 4.19(r)).

4.6.3. Estimação de Taxa Mínima

A fim de reduzir o número de pedidos a ser feito pelo decodificador ou reduzir a quantidade de síndromes acumuladas enviadas por causa da incerteza sobre o sucesso da decodificação, o decodificador pode estimar um número mínimo de síndromes acumuladas a serem enviadas por bitplane e por banda, pois dentre as 15 bandas, há as mais significativas (com maiores valores de seus coeficientes), e dentre os M_k bitplanes de certa banda b_k , há também uma ordem de importância. Assim, para os bitplanes mais significativos envia-se um percentual maior de síndromes acumuladas e para os bitplanes menos significativos um percentual menor.

O aumento do número de pedidos de envio de mais síndromes acarreta um maior tempo de decodificação, pois para cada pedido, um novo conjunto de iterações do decodificador LIA e uma nova verificação de erros deverá ser executada. Por isso, torna-se necessária a estimação de uma taxa mínima $R_{k,min}$, a fim de que não haja muitas retransmissões e, ao mesmo tempo, a taxa não seja alta o bastante de modo que o número de síndromes acumuladas não se iguale ao número de síndromes.

O método adotado pelo codec proposto para estimar a taxa mínima é baseado no limite taxa-distorção de Wyner-Ziv [2] para duas fontes gaussianas correlatadas. Este limite define que a taxa mínima na qual uma fonte X_{2i} pode ser

transmitida, considerando-se uma dada distorção $D_k = E[d(v_k, \hat{v}_k)]$ entre os valores reais da banda b_k e os reconstruídos, deve ser igual a:

$$R_{k,min}(D_k) = \frac{1}{2} \cdot \log_2 \left(\frac{\sigma_k^2}{D_k} \right) \quad (4.21)$$

a qual é chamada de função taxa-distorção. Já a função distorção-taxa é dada por:

$$D_{k,Máx}(R_k) = \frac{2^{-2R_{k,Min}}}{\sigma_k^2} \quad (4.22)$$

onde σ_k^2 é a variância do ruído de correlação entre duas fontes, em nível de banda (conforme cálculos descritos na seção 4.6.5.2), dado que a segunda fonte (Y_{2i} , a informação lateral) é perfeitamente conhecida no decodificador. O cálculo de $D_{k,Máx}$ é descrito com maiores detalhes em [74, 75]. Em [74], a única menção sobre o controle de taxa foi o fato de um bitplane não poder ter uma probabilidade de erro maior do que um certo valor, neste caso 10^{-3} , o que era determinado de forma ideal do lado do decodificador. Este valor também foi adotado em [68].

Uma taxa separada para cada bitplane pode ser obtida estimando-se a diminuição da distorção pelo acréscimo de cada bitplane, em relação aos bitplanes decodificados previamente. σ_k^2 é um parâmetro de modelagem do canal de correlação, o qual será discutido na seção 4.6.5.2. Ele é estimado no lado do decodificador e é enviado de volta ao codificador via canal de retorno. Um método alternativo, que foi adotado tanto neste trabalho como em [10], é baseado na entropia condicional $H(X_{2i}/Y_{2i})$ entre o dado a ser codificado X_{2i} e a informação lateral Y_{2i} . Esta entropia condicional definida pela eq. (4.23), por sua vez, depende da probabilidade cruzada apresentada na eq. (4.24).

$$H(x_{i,j} | y_j) = -p_{CR} \log_2 p_{CR} - (1 - p_{CR}) \log_2 (1 - p_{CR}) \quad (4.23)$$

$$p_{CR} \equiv \Pr(x_{k,i,j} \neq \hat{x}_{k,i,j}(y_{k,j})) \quad (4.24)$$

onde $\hat{x}_{k,i,j}(y_{k,j})$ é a estimativa gerada pelo decodificador sobre o j -ésimo bit do i -ésimo *bitplane* da k -ésima banda de X_{2i} , conhecendo-se o valor da informação lateral $y_{k,j}$. Enfim, a probabilidade cruzada é estimada utilizando o modelo do canal virtual, a qual torna-se conhecida para o codificador através do canal de *feedback*. A partir deste ponto será omitido, nesta seção, o índice k nos coeficientes x e y , subentendendo-se que os cálculos envolvendo os mesmos estão relacionados a k -ésima banda.

O modelo de correlação entre as bandas DCT de X_{2i} e as correspondentes da informação lateral Y_{2i} é assumido como laplaciano segundo Ascenso et al. [69] (explicações mais detalhadas na seção 4.6.5.2), sendo este tipo de modelagem considerada conhecida pelo codificador. Assim, a quantidade p_{CR} é estimada a partir de um modelo de correlação laplaciano da seguinte maneira: o cruzamento de valores ocorre no i -ésimo *bitplane* para um coeficiente DCT em particular x_i , quando o j -ésimo bit $x_{i,j}$ difere do bit observado $\hat{x}_{i,j}$, o qual é estimado no decodificador utilizando o j -ésimo coeficiente DCT da informação lateral da k -ésima banda, $y_{k,j}$, e os bits pertencentes aos *bitplanes* já decodificados $(\hat{x}_{i+1,j}, \hat{x}_{i+2,j}, \dots, \hat{x}_{Mk-1,j}, \hat{x}_{Mk-2,j})$, isto é, quando:

$$x_{i,j} \neq \hat{x}_{i,j} \equiv \arg \max_{b=0,1} \Pr(b | y_{k,j}, \hat{x}_{i+1,j}, \hat{x}_{i+2,j}, \dots, \hat{x}_{Mk-1,j}, \hat{x}_{Mk-2,j}) \quad (4.25)$$

onde $\Pr(b | y_{k,j}, \hat{x}_{i+1,j}, \hat{x}_{i+2,j}, \dots, \hat{x}_{Mk-1,j}, \hat{x}_{Mk-2,j})$ é probabilidade *a posteriori* do evento $x_{i,j} = b$. O procedimento para calcular o valor de $\Pr(x_{2,j} = 1 | y_j, \hat{x}_{1,j})$ é ilustrado na figura 4.33.

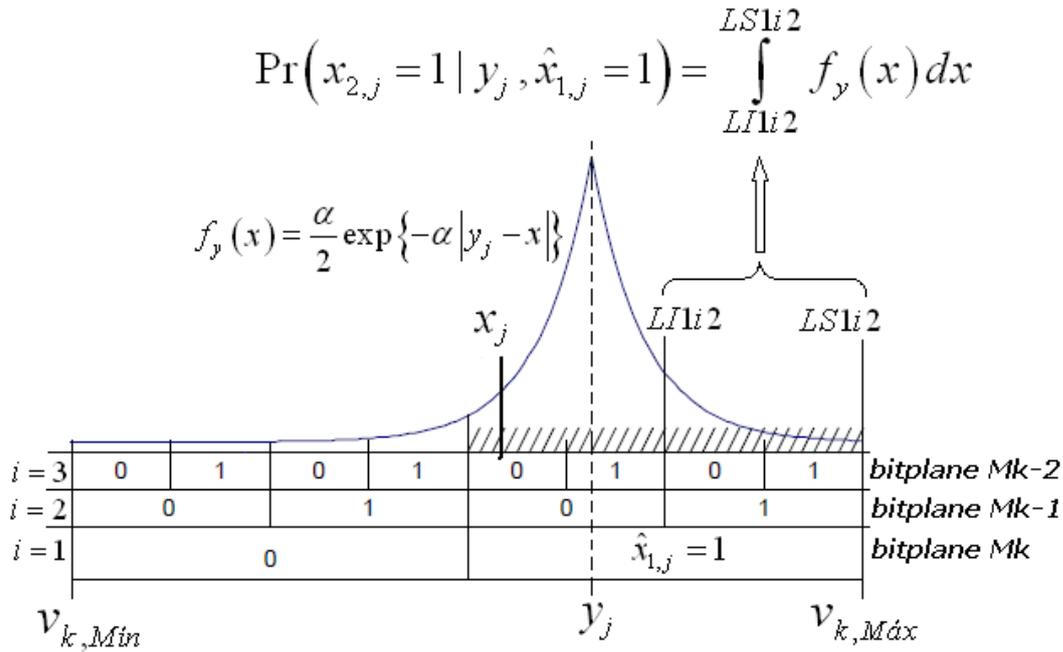


Figura 4.33: Probabilidade condicional do bit $x_{2,j}$ dada a informação lateral y_j e que o bit anterior, $x_{1,j}$, foi estimado com valor igual 1.

Na figura 4.33 acima, y_j é o j -ésimo coeficiente da banda b_k da informação lateral e representa o centro da distribuição laplaciana $f_y(x)$. $Ll12$ e $Ls12$ representam os limites inferior e superior, respectivamente, da integral da distribuição, para $x_{2,j} = 1$. A probabilidade cruzada para o i -ésimo *bitplane* é obtida através do valor esperado da probabilidade cruzada $\Pr(x_{i,j} \neq \hat{x}_{i,j})$, apresentada na eq. (4.26), dada a distribuição laplaciana centrada no j -ésimo coeficiente da banda b_k , y_j , o qual é calculado pela média dos coeficientes DCT de Y_{2i} ao longo de toda a banda b_k . Assim, a probabilidade cruzada é calculada da seguinte forma:

$$p_{CR} = \frac{1}{n} \sum_{x \in v_k} \left[\int_{-\infty}^{+\infty} \Pr(x_{i,j} \neq \hat{x}_{i,j}) \frac{\alpha_k}{2} e^{-\alpha_k |y_j - x|} dx \right] \quad (4.26)$$

onde n é o comprimento da banda b_k e α_k é o valor do parâmetro laplaciano da banda b_k .

4.6.3.1. Medida de Confiança

Aqui se descreve um método utilizado para estimar a taxa de erro na saída do decodificador LIA. A técnica é baseada em uma medida de confiança sobre os bits estimados e é calculada através da razão de verossimilhança $\Lambda_{i,j}$, segundo a eq. (4.27), relacionada ao j -ésimo bit do *bitplane* i :

$$\Lambda_{i,j} = \log_{10} \frac{P(\hat{d}_{i,j} = 1 | y_j)}{P(\hat{d}_{i,j} = 0 | y_j)} \quad (4.27)$$

onde $\hat{d}_{i,j}$ é o bit observado j do *bitplane* i e y_j é o valor da informação lateral correspondente. O valor $|\Lambda_{i,j}|$ é usado como medida de confiança para cada bit: o bit i é considerado não confiável (incerto) se $|\Lambda_{i,j}|$ estiver abaixo de um certo limiar, expresso por $\log_{10} T$, onde $50 < T < 100$. Assim, a estimativa é considerada confiável se:

$$\text{Estimativa confiável: } \begin{cases} P1/P0 \geq T & \text{se } P1 \geq P0 \\ P0/P1 \geq T & \text{se } P1 < P0 \end{cases} \quad (4.28)$$

onde $P1/P0 = P(\hat{d}_{i,j} = 1 | y_j) / P(\hat{d}_{i,j} = 0 | y_j)$. Por exemplo, estabelecendo que $T=99$ implica que somente os bits que atendem à restrição:

$$\max(P(\hat{d}_{i,j} = 1 | y_j), P(\hat{d}_{i,j} = 0 | y_j)) \geq 0,99$$

serão considerados como decodificados com sucesso. Para obter o número total de decodificações bem sucedidas para o *bitplane*, propõe-se aqui um cálculo de *NCC* (número de casos confiáveis), o qual é determinado pelo cálculo do número de bits com $|\Lambda_{i,j}| < \log_{10} T$, dividido pelo comprimento do *bitplane*:

$$NCC = \frac{\sum \{|\Lambda_{i,j}| < \log_{10} T\}}{n} \quad (4.29)$$

onde n é o comprimento do bitplane .Se a contagem do número de casos em que a restrição de confiança não foi atendida for maior que 10^{-3} (ou $NCC < 1-10^{-3}$), o decodificador LIA pede mais bits ao *buffer*, caso contrário, prossegue-se com o próximo conjunto de síndromes acumuladas. As probabilidades acima são fornecidas como resultado da decodificação LIA. O algoritmo 4.3 (abaixo) foi utilizado neste trabalho para realizar este processo.

Algoritmo 4.3: verificação de confiança em relação aos bits decodificados

```

Conte=0;
Para k=1:N²-1
  Para i=1:Mk
    Para j=1:n
       $\Lambda_{i,j} = \log_{10} \left[ \frac{P(\hat{d}_{i,j} = 1 | y_j)}{P(\hat{d}_{i,j} = 0 | y_j)} \right];$ 
      Se  $|\Lambda_{i,j}| < \log_{10} T$  //se a estimativa for confiável
        conte = conte + 1;
      fim
    fim
  NCC = conte/n; //percentual de estimativas confiáveis
  Se  $NCC \geq 1-10^{-3}$ 
    "Decodificação bem sucedida"
  caso contrário:
    "Decodificação mal sucedida"
    "Requisitar mais bits"
  fim
fim
fim

```

4.6.4. Modelagem do Canal de Correlação

Para o decodificador fazer uso da informação lateral Y_{2i} , ele precisa ter um conhecimento confiável ou razoavelmente preciso do modelo que caracteriza as relações estatísticas entre o frame original e sua estimativa. A dependência estatística entre esses dois frames corresponde a um canal virtual (ver figura 4.1) com um padrão de erro caracterizado por certa distribuição estatística ou modelo, já que a informação lateral pode ser vista como uma versão corrompida ou distorcida da informação original.

Se o modelo descrever com precisão a relação entre X_{2i} e Y_{2i} , ter-se-á uma grande eficiência na codificação. Entretanto, se o modelo falhar ou se houver uma

grande falta de equivalência entre a verdadeira correlação (entre X_{2i} e Y_{2i}) e a correlação estimada pelo canal virtual, será observada uma queda na eficiência da decodificação. No contexto da figura 4.1, uma falha no modelo de correlação implica em uma informação auxiliar menos precisa na entrada do decodificador do canal, o que refletirá em uma maior dificuldade na correção de erros e, conseqüentemente, a necessidade de uma maior requisição de bits para uma correta decodificação.

O codec empregado neste trabalho faz uso de uma distribuição laplaciana para modelar o ruído de correlação, isto é, para modelar a distribuição do resíduo entre as bandas DCT correspondentes aos frames X_{2i} e Y_{2i} . O parâmetro da distribuição laplaciana é estimado em tempo real no decodificador e leva em consideração a variabilidade temporal (variação frame a frame) e espacial (ao longo do frame) das estatísticas do ruído de correlação $f(X_{2i} - Y_{2i})$.

Esta técnica evita uma prática comum na literatura [68], que é calcular os parâmetros da distribuição do ruído de correlação fazendo uso de um estágio de treinamento ou preparação que não é feito em tempo real, ou melhor, não é feito enquanto a sequência é decodificada, mas o cálculo dos parâmetros utilizados na modelagem do canal de correlação é realizado antes de iniciar a codificação da sequência. Este processo *offline* não é realista, pois ele requer ou que o codificador recrie a informação lateral (aumentando a complexidade do codificador) ou que os dados originais estejam disponíveis no decodificador, ou seja, sem distorção, o que necessitaria de uma compressão sem perdas dos *Key frames*, tornando-se inviável na prática.

As técnicas empregadas aqui neste trabalho, baseadas em [69], propõem uma estimação do parâmetro α da distribuição laplaciana em função da banda (um parâmetro α por banda) e em função dos coeficientes DCT (um parâmetro α por coeficiente DCT). O método de estimação do parâmetro α utiliza um frame residual, no caso, a diferença entre \tilde{X}_{2i-1} e \tilde{X}_{2i+1} , como uma medida de confiança da operação de interpolação do frame, conforme mostra a seção 4.6.7. Então, o modelo de distribuição laplaciana é utilizado para converter os coeficientes DCT da informação lateral em um vetor de entrada com valores suavizados (mais precisos) para o decodificador LDPC, conforme se observou na figura 4.17. Assim, de acordo com o que foi apresentado na seção anterior (4.6.4), a

probabilidade condicional $P(X_{2i}|Y_{2i})$ obtida para cada coeficiente DCT é convertida em probabilidades condicionais de bits considerando-se o conhecimento dos bitplanes anteriormente decodificados e do valor da informação lateral.

4.6.4.1.

Estimação do Ruído de Correlação em nível de Sequência (*offline*)

Conforme mencionado na seção 4.3.12, nos trabalhos anteriores em que os autores procuraram otimizar a decodificação pelo uso do canal de correlação [60, 68, 69, 74], esses utilizaram uma distribuição laplaciana, conforme mostra a eq. (4.30), para modelar a correlação estatística entre o frame original e a informação lateral.

$$f(X_{2i} - Y_{2i}) = \frac{\alpha}{2} \exp\{-\alpha |X_{2i} - Y_{2i}|\} \quad (4.30)$$

A distribuição laplaciana é utilizada para converter a informação lateral (valores de pixel) em uma informação com entradas suaves (números reais), a qual é necessária à decodificação LDPC adaptativa. O parâmetro da distribuição laplaciana α pode ser calculado por:

$$\alpha = \frac{2}{\sigma^2} \quad (4.31)$$

No caso da estimação em nível de sequência, o parâmetro α é calculado *offline*, no codificador, com base na sequência inteira de vídeo e enviado para o decodificador antes de iniciar o procedimento de codificação Wyner-Ziv. Ele é mantido constante para decodificação de todos os frames WZ. Na eq. (4.31), σ^2 é a variância do resíduo entre os frames X_{2i} e Y_{2i} . Entretanto, este processo para calcular o α não é muito eficiente, além de não realista, pois não explora a variabilidade do modelo de correlação ao longo do tempo (variações entre frames) e ao longo do espaço (entre regiões de um frame).

4.6.5. Modelos de Ruído de Correlação Estimados *Online*

O processo de estimação *offline* do ruído de correlação (no codificador) não é realístico porque requer que o codificador recrie a informação lateral. Visto que são utilizados algoritmos de estimação e compensação do movimento para gerar a informação lateral, esta tarefa não pode ser realizada em um codificador de baixa complexidade, o que é um dos principais objetivos da codificação distribuída.

Nesse contexto, a abordagem prática é execução de uma estimação dinâmica do parâmetro da distribuição laplaciana: em nível de frame ou bloco, como os autores em [60, 68, 69, 74], ou em nível de banda ou coeficiente, como é proposto neste trabalho. Esta estimação deve ser feita no decodificador, onde mais recursos computacionais estão disponíveis, de acordo com o paradigma DVC. Além disso, o parâmetro da distribuição laplaciana não deve ser transmitido do codificador para o decodificador, o que, tipicamente, aconteceria em condições propensas a erros.

A proposta desta seção reside no estudo e implementação de um novo método de estimação do parâmetro α , que funcione eficientemente no decodificador. Isso representa um passo importante na solução da codificação de vídeo Wyner-Ziv, uma vez que não é mais necessário recriar a informação lateral no lado do codificador. Na literatura, encontram-se dois níveis de estimação *online*: de frame e de bloco. Neste trabalho, além de serem implementados estes dois métodos de estimação, em nível de frame e de bloco, também são propostas mais duas estimativas, uma em nível de banda e outra em nível de coeficientes DCT, todas seguindo a arquitetura descrita na seção 4.1, mas com modificações na configuração da modelagem do canal de correlação.

4.6.5.1. Estimação do Ruído de Correlação em nível de Frame

A fim de explorar a variabilidade no tempo do modelo de ruído de correlação, um método possível é realizar o cálculo da variância do resíduo entre os frames \tilde{X}_{2i-1} e \tilde{X}_{2i+1} , usar a eq. (4.31) para calcular o parâmetro laplaciano α . Em seguida, o valor do parâmetro da distribuição laplaciana é utilizado no processo de decodificação do frame X_{2i} .

Para estimar o parâmetro da distribuição laplaciana no decodificador, é necessário definir uma métrica que expresse a relação entre a informação original e a lateral, já que o frame original não está disponível no decodificador. Assim, a técnica de estimação em nível de frame, segundo [74], pode ser descrita pelos seguintes passos:

I. Geração do Frame Residual: a priori, calcula-se o frame residual R_{2i} a partir das versões dos frames reconstruídos \tilde{X}_{2i-1} e \tilde{X}_{2i+1} , compensados em movimento, conforme a eq. (4.32):

$$R_{2i}(l, c) = \left| X_B \left(l + \frac{\Delta l_B}{2}, c + \frac{\Delta c_B}{2} \right) - X_F \left(l + \frac{\Delta l_F}{2}, c + \frac{\Delta c_F}{2} \right) \right| \quad (4.32)$$

Os termos $X_B(l + \Delta l_B/2, c + \Delta c_B/2)$ e $X_F(l + \Delta l_F/2, c + \Delta c_F/2)$ representam os frames *key* reconstruídos e compensados em movimento, a partir de uma estimação reversa e direta, respectivamente, ou melhor, X_B é o frame \tilde{X}_{2i+1} compensado em movimento em relação a \tilde{X}_{2i-1} , e X_F é o frame \tilde{X}_{2i-1} compensado em movimento em relação a \tilde{X}_{2i+1} . Como o frame WZ situa-se entre os dois frames, a compensação é feita considerando o vetor de movimento para o frame WZ, MV_{WZ} , igual ao vetor estimado MV dividido por 2, o que será mais bem explicado na seção 4.6.7. Neste contexto, $(\Delta l_B, \Delta c_B)$ e $(\Delta l_F, \Delta c_F)$ na eq. (4.28) representam os vetores de movimento estimados para os frames X_B e X_F , respectivamente. As coordenadas (l, c) correspondem à localização da linha e coluna do pixel no frame R_{2i} .

II. Cálculo da variância do Frame Residual: a variância do frame residual é obtida pela fórmula:

$$\begin{aligned} \sigma_R^2 &= E \left\{ \left[R_{2i}(l, c) - E[R_{2i}(l, c)] \right]^2 \right\} = \\ \sigma_R^2 &= E \left\{ [R_{2i}(l, c)]^2 \right\} - \left\{ E[R_{2i}(l, c)] \right\}^2 \end{aligned} \quad (4.33)$$

III. Estimação do parâmetro α : a variância σ_R^2 é uma medida do grau de confiança do processo de criação da informação lateral, a qual indica quão eficaz é o efeito da interpolação. Idealmente, o valor de σ_R^2 deveria ser igual à variância do resíduo entre os frames Wyner-Ziv e a informação lateral. Portanto, utiliza-se a medida de variância definida na eq. (4.34) como uma forma de representar a variância entre a informação lateral e a informação original, mas desta vez o cálculo é realizado em nível de frame. A estimativa do parâmetro α para cada frame Wyner-Ziv é obtida através da eq. (4.31), substituindo-se σ^2 (nível de sequência) pelo σ_R^2 (nível de frame), resultando na eq. (4.34):

$$\hat{\alpha}_{2i} = \frac{2}{\sigma_R^2} \quad (4.34)$$

4.6.5.2.

Estimção Proposta para o Ruído de Correlação: em Nível de Banda

Adaptando-se o parâmetro da distribuição laplaciana (entre os frames WZ e a informação lateral) em nível de banda, pode-se melhorar o desempenho taxa-distorção do codec, quando comparado à abordagem da estimção em nível de sequência e frame, pois há uma maior especificação (ou precisão) no cálculo do α , já que há 4^2 bandas num frame (quando se utiliza uma DCT 4x4) e um parâmetro α associado a cada uma.

Assim, propõe-se uma técnica de estimção do parâmetro α em nível de banda, realizada no decodificador, com o objetivo de melhorar o desempenho taxa-distorção, quando comparado ao da estimção em nível de frame. A abordagem proposta aqui pode ser descrita pelos seguintes passos:

- I. Geração do Frame Residual:** inicialmente, calcula-se o frame residual R_{2i} a partir das versões dos frames reconstruídos \tilde{X}_{2i-1} e \tilde{X}_{2i+1} , compensados em movimento, conforme a expressão descrita na eq. (4.32).
- II. Cálculo da Variância da k -ésima Banda do Frame Residual:** a variância da k -ésima banda, $\sigma_{R,k}^2$, pode ser obtida a partir da eq. (4.35),

considerando, no entanto, a operação de “valor esperado” sobre as n amostras de cada banda b_k , onde n é o número de blocos N por N do frame (definido na eq. 4.9), assim:

$$\sigma_{R,k}^2 = E \left\{ v_{R,k}^2 \right\} - \left\{ E \left[v_{R,k} \right] \right\}^2 \quad (4.35)$$

III. Estimação do Parâmetro α : o cálculo proposto para obter a estimativa do parâmetro α da k -ésima banda do frame X_{2i} é dado por:

$$\hat{\alpha}_k^2 = \frac{2}{\sigma_{R,k}^2} \cdot \frac{E \left[v_{R,k} \right]}{E \left[R_{2i}(l, c) \right]} \quad (4.36)$$

Em (4.36), $E \left[v_{R,k} \right]$ é o “valor esperado” da k -ésima banda de n coeficientes do frame residual e $E \left[R_{2i}(l, c) \right]$ é o “valor esperado” do frame R .

4.6.5.3. Estimação do Ruído de Correlação em Nível de Bloco

Dentro do frame Y_{2i} existem regiões onde o processo de interpolação foi bem sucedido (alta correlação entre os blocos de X_{2i} e os correspondentes de Y_{2i}) e regiões onde a interpolação foi mal sucedida (baixa correlação), logo, é necessária uma adaptação do modelo de ruído de correlação para o nível de blocos, ou melhor, para um nível de estimação com uma precisão maior que a dos níveis anteriores.

O algoritmo de interpolação de frames falha com certa frequência em regiões onde ocorreu uma grande quantidade de movimento ou em regiões inicialmente descobertas pela compensação do movimento, conforme seção 4.6.7, porém, a interpolação é bastante eficiente onde a quantidade de movimento é baixa, como, por exemplo, em planos de fundo etc. Visto que áreas diferentes do frame Y_{2i} estão associadas a diferentes quantidades de erros de interpolação, é esperado que o modelo do ruído de correlação entre os frames X_{2i} e Y_{2i} varie dentro do frame.

Com o objetivo de explorar a variabilidade espacial do modelo de ruído de correlação, o parâmetro da distribuição laplaciana do j -ésimo bloco, α_j , é calculado para cada bloco de $h \times h$ amostras, onde h é a ordem do bloco utilizado no estágio de estimação e interpolação do módulo de geração da informação lateral, a fim de detectar com maior precisão os erros de interpolação. O procedimento adotado para estimação do α em nível de bloco pode ser descrito pelos seguintes passos:

I. Geração do Frame Residual: inicialmente, calcula-se o frame residual R_{2i} a partir das versões dos frames reconstruídos \tilde{X}_{2i-1} e \tilde{X}_{2i+1} , compensados em movimento, conforme a expressão descrita na eq. (4.32).

II. Cálculo da Variância do j -ésimo Bloco do Frame Residual: a variância $\sigma_{R,j}^2$ do j -ésimo bloco $h \times h$ pode ser obtida através da equação:

$$\sigma_{R,j}^2 = E \left\{ \left[B_{R,j}(l, c) \right]^2 \right\} - \left\{ E \left[B_{R,j}(l, c) \right] \right\}^2 \quad (4.37)$$

III. Estimação do Parâmetro α : a estimativa do parâmetro α para o k -ésimo bloco do frame R é dada por:

$$\hat{\alpha}_{R,j}^2 = \frac{2}{\sigma_{R,j}^2} \cdot \frac{E \left[B_{R,j}(l, c) \right]}{E \left[R_{2i}(l, c) \right]} \quad (4.38)$$

Na eq. (4.38), $E \left[R_j(l, c) \right]$ é a operação de “valor esperado” sobre o j -ésimo bloco do frame residual R e $E \left[R_{2i}(l, c) \right]$ é o “valor esperado” do frame R .

4.6.5.4. Estimação Proposta para o Ruído de Correlação: em Nível de Coeficiente

Ainda que se tenha buscado otimizar o processo de estimação com o uso do método de estimação em nível de bloco, descrito na seção anterior, consegue-se obter um aumento na eficiência do processo propondo-se uma estimação em nível

de coeficientes. Assim, nesta seção é proposta uma técnica também *online* para estimar o parâmetro da distribuição laplaciana em nível de coeficientes DCT. Esta técnica tem em vista alcançar a melhor modelagem do canal virtual dentre os métodos apresentados, quando comparada aos níveis menos precisos (sequência, frame, banda e bloco), embora demande um número bem maior de cálculos de parâmetros laplacianos. A abordagem da estimação do α em nível de coeficientes DCT proposta aqui é descrita pelos seguintes passos:

- I. **Geração do Frame Residual:** inicialmente, calcula-se o frame residual R_{2i} a partir das versões dos frames reconstruídos \tilde{X}_{2i-1} e \tilde{X}_{2i+1} , compensados em movimento, conforme a expressão descrita na eq. (4.32).
- II. **Estimação do parâmetro α :** a estimativa do parâmetro α associado a cada coeficiente j de cada banda b_k , é dada por:

$$\hat{\alpha}_{(k,j)}^2 = \frac{2}{R_{2i}(k,j)} \cdot \frac{1}{E[R_{2i}(k)]} \quad (4.39)$$

onde $R_{2i}(k, j)$ é o valor do elemento situado na posição j da k -ésima banda e $R_{2i}(k)$ é o vetor-banda.

4.6.6. Avaliação da Modelagem do canal de Correlação

Neste experimento, tem-se como objetivo avaliar a eficiência de cada método utilizado para estimar o parâmetro laplaciano no processo de modelagem do canal virtual. Sendo assim, as figuras 4.34 a 4.39 exibem os resultados experimentais dos diversos níveis de estimação, utilizando-se 100 frames (50 *key frames* e 50 *WZ frames*) das sequências *Foreman*, *News*, *Coastguard* e *Soccer*, QCIF, onde são apresentados os valores da PSNR (ao longo da sequência), para diferentes níveis de granularidade, ou melhor, para diferentes níveis de estimação de parâmetros do canal de correlação. É importante ressaltar que todas as estimações realizadas neste trabalho foram *online*, ou seja, no decodificador (em nível de frame, banda, bloco e coeficiente). Nesta simulação, considerou-se 15

bandas de frequência e 396 blocos de dimensão 8x8 (144x176/64) que foram utilizados no processo de estimação de movimento do módulo de geração da informação lateral.

Observa-se nas figuras 4.34 a 4.39 que os valores de PSNR dos métodos de estimação em nível de bloco, banda e coeficiente são bastantes próximos um do outro, sendo os três bem superiores ao método de estimação em nível de frame. Porém, observe a tabela 4.4 abaixo contendo a PSNR média dos frames e o número de parâmetros laplacianos que precisam ser calculados nos três métodos com melhores resultados:

Tabela 4.4: PSNR média, número de parâmetros laplacianos que necessitam ser calculados e número de médio de operações para cada nível de granularidade.

Nível	PSNR médio (dB)				Número de parâmetros Laplacianos	Tempo para estimar os parâmetros Laplacianos
	Foreman	News	Coastguard	Soccer		
Banda	37,08	37,07	37,00	36,61	15	5,5 ms
Bloco	37,12	36,98	37,01	36,59	1.584	6,2 ms
Coeficiente	37,22	37,18	37,11	36,72	25.344	8,5 ms

Na tabela 4.4 acima, nota-se que a PSNR média dos três níveis é bastante similar, conforme havia sido comentado acima, mas o número de parâmetros laplacianos que precisam ser calculados para o caso do método de estimação em nível de banda, que é o método proposto, é muito menor que o número de parâmetros do método de estimação em nível de bloco, convencionalmente utilizados em DVC, e que o do outro método proposto, que é a estimação em nível de coeficientes, a qual obteve a maior PSNR média.

Além disso, as figuras 4.38 e 4.39 mostram claramente a grande diferença de PSNR dos frames reconstruídos ao longo da sequência de vídeo, com e sem a utilização da modelagem do canal de correlação, onde se conclui que o módulo de modelagem do canal de correlação é essencial para um auxílio eficiente da informação lateral na correta decodificação do frame Wyner-Ziv, ou seja, embora seja possível realizar a decodificação do frame WZ sem a modelagem do canal de correlação, segundo as conclusões acima, a diferença da qualidade do frame reconstruído é bastante significativa.

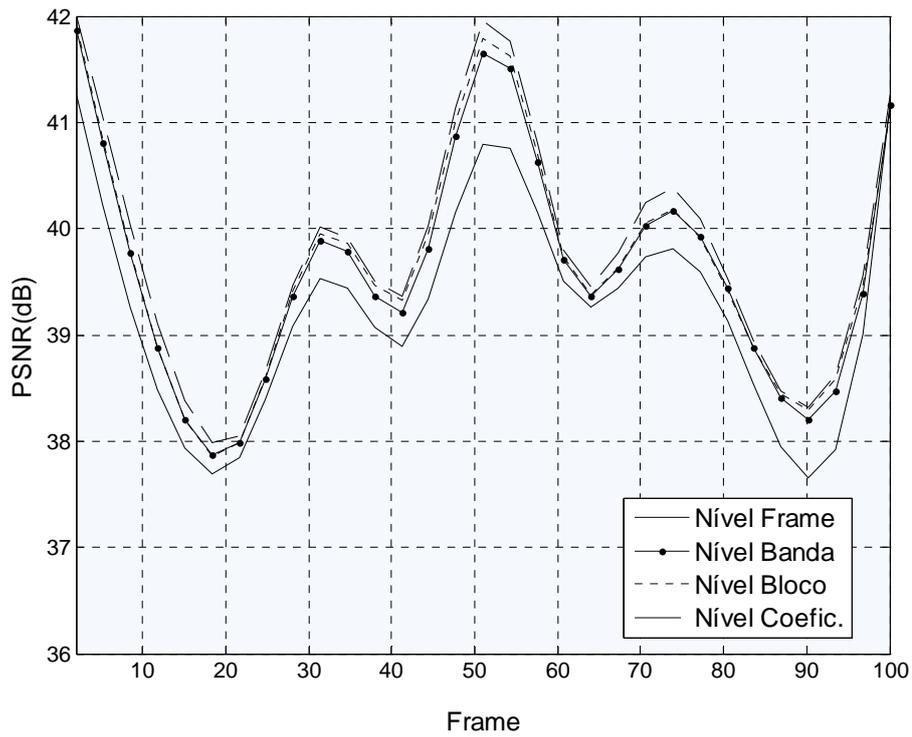


Figura 4.34: PSNR para diferentes níveis de estimação do parâmetro laplaciano, para a sequência *Foreman*, QCIF, com QP = 32 (intra) e TQ = 9 (WZ).

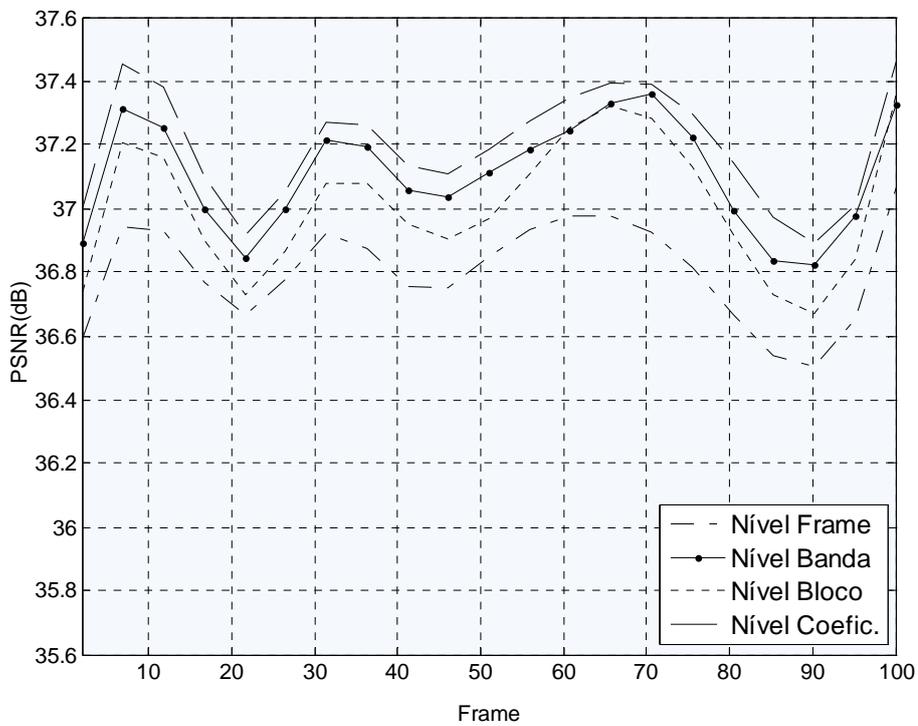


Figura 4.35: PSNR para diferentes níveis de estimação do parâmetro laplaciano, para a sequência *News*, QCIF, com QP = 32 (intra) e TQ = 9 (WZ).

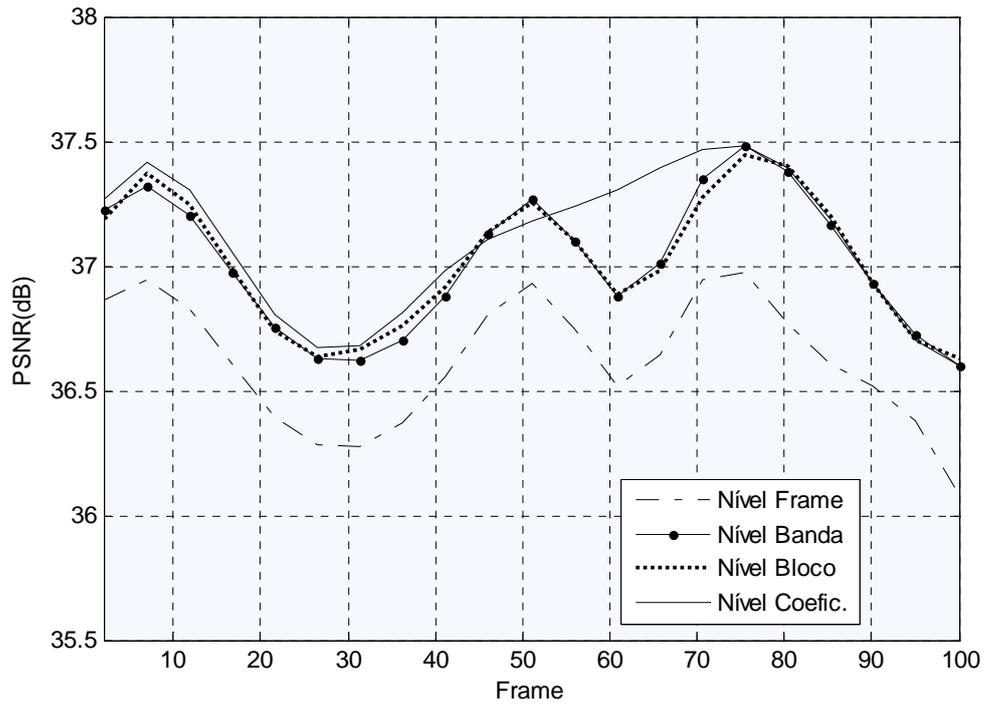


Figura 4.36: PSNR para diferentes níveis de estimação do parâmetro laplaciano, para a sequência *Coastguard*, QCIF, com QP = 32 (intra) e TQ = 9 (WZ).

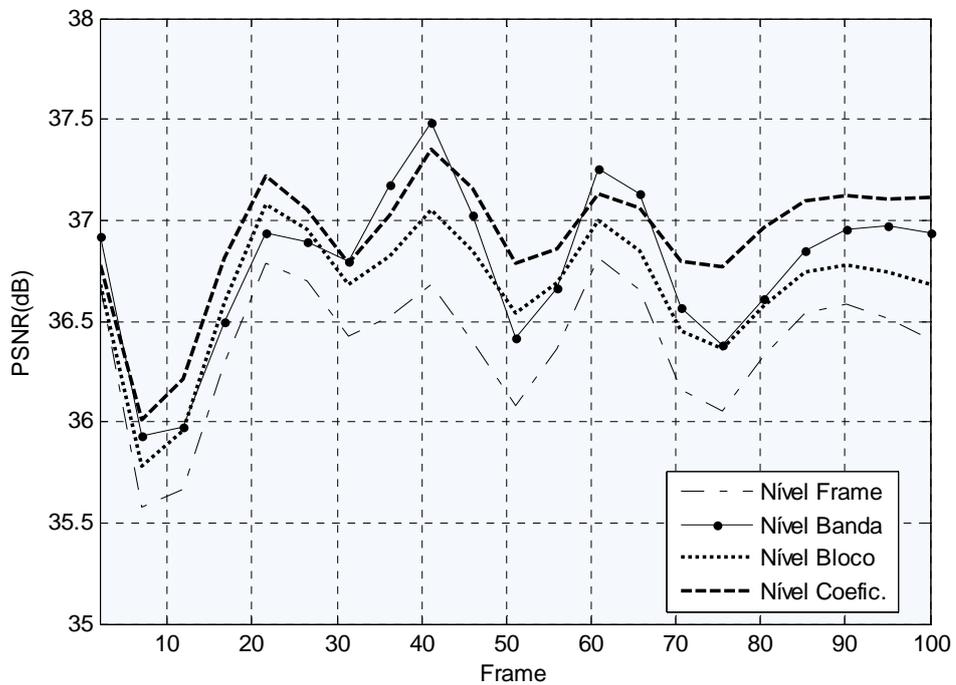


Figura 4.37: PSNR para diferentes níveis de estimação do parâmetro laplaciano, para a sequência *Soccer*, QCIF, com QP = 32 (intra) e TQ = 9 (WZ).

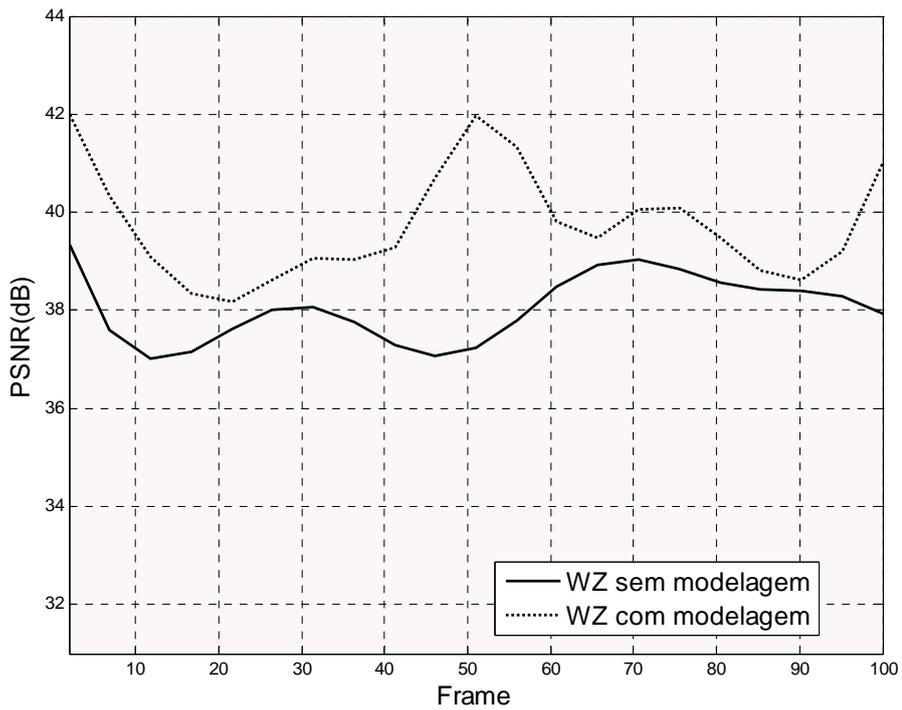


Figura 4.38: PSNR para decodificação com e sem a utilização da modelagem do canal de correlação, para a sequência *Foreman*, QCIF, com QP = 32 (intra) e TQ = 9 (WZ).

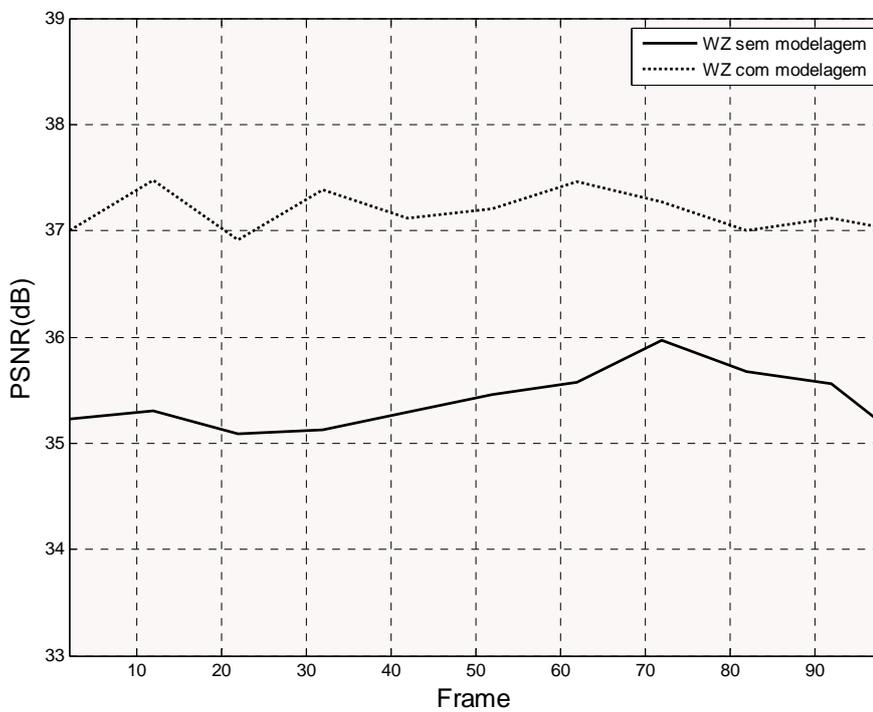


Figura 4.39: PSNR para decodificação com e sem a utilização da modelagem do canal de correlação, para a sequência *News*, QCIF, com QP = 32 (intra) e TQ = 9 (WZ).

Outras observações importantes a respeito do experimento: para estimação dinâmica no decodificador do parâmetro da distribuição laplaciana, o desempenho alcançado é proporcional ao nível de granularidade, ou melhor, quanto mais preciso este for, mais eficiente será o resultado da estimação (ou seja, a estimativa). Ganhos ainda maiores podem ser esperados se a informação lateral exibe variações temporais e/ou espaciais mais abruptas. O nível mais preciso de granularidade é o que demanda maior complexidade computacional, no entanto, isto não deve ser considerado uma questão crítica, já que o aumento da complexidade ocorre no lado do decodificador, onde a complexidade não é um grande problema, segundo o paradigma da codificação DVC. Entretanto, é importante notar que os métodos propostos aqui para estimação *online* têm a grande vantagem de serem realizados no decodificador, proporcionando um cenário de codificação Wyner-Ziv mais realista.

4.6.7. Geração da Informação Lateral

Foi mencionado anteriormente que o decodificador Wyner-Ziv requer que a informação lateral esteja disponível no decodificador para reconstruir X_{2i} , mas a forma como é gerada a informação lateral no decodificador varia de acordo com o autor. Neste trabalho, a arquitetura proposta é ilustrada pela figura 4.41. Os frames ímpares da sequência de vídeo, X_{2i-1} e X_{2i+1} (chamados de frames *key*) estão disponíveis no decodificador com uma baixa taxa de compressão, ou seja, uma versão com baixa distorção: \tilde{X}_{2i-1} e \tilde{X}_{2i+1} ; partindo deste princípio, existem diversas técnicas que podem ser empregadas no decodificador Wyner-Ziv para gerar a informação lateral Y_{2i} . Por exemplo, o método adotado por Brites [70], embora utilize ferramentas bastante eficientes na geração da informação lateral, não é prático, pois utilizam-se os frames *key* originais X_{2i-1} e X_{2i+1} no processo de interpolação ao invés dos reconstruídos (\tilde{X}_{2i-1} e \tilde{X}_{2i+1}), o que leva a concluir que não foi utilizado o processo de quantização na codificação *intraframe*, o que resulta em uma baixíssima compressão dos frames *key*.

As técnicas utilizadas na geração da informação lateral no decodificador influenciam significativamente no desempenho do codec DVC, segundo a análise

taxa-distorção, da mesma forma que ferramentas eficientes de estimação e compensação de movimento têm gerado grandes avanços na compressão e no desempenho do codificador convencional de vídeo.

Visto que o objetivo em DVC é encontrar uma estimativa do frame Wyner-Ziv atual, chamada aqui de Y_{2i} , é necessário um conjunto diferente de ferramentas para realizar tal tarefa. Neste trabalho, propõe-se um novo método para geração da informação lateral, o qual inclui uma estimação bidirecional de movimento, compensação, dois métodos de interpolação e uma extrapolação, a fim de aumentar a eficiência da geração da informação lateral e, conseqüentemente, do codec DVC proposto.

As técnicas mais simples de interpolação de frame que podem ser utilizadas são: i) fazer Y_{2i} igual ao frame adjacente anterior a X_{2i} , ilustrado na figura 4.40 por \tilde{X}_{2i-1} , na qual despreza-se a variação temporal; ou ii) desempenhar uma interpolação entre os frames *key* \tilde{X}_{2i-1} e \tilde{X}_{2i+1} . Entretanto, se alguma dessas técnicas for empregada para gerar a informação lateral em uma seqüência de vídeo onde a similaridade entre dois frames adjacentes não é tão alta, Y_{2i} será uma estimativa grosseira de X_{2i} . Neste caso, é fácil perceber que o decodificador precisará pedir mais bits de paridade (mais síndromes) para o codificador a fim de decodificar os bitplanes, quando comparado ao caso onde Y_{2i} é uma estimativa bem próxima do frame X_{2i} . Assim, a taxa de bits Wyner-Ziv aumentará para uma mesma PSNR obtida, diminuindo-se, então, a eficiência do codec. A taxa de bits Wyner-Ziv refere-se ao número de bits WZ (bits de paridade) necessário para decodificar os bitplanes e reconstruir o *stream* de símbolos quantizados q_{2i} . Subjetivamente, essas técnicas mais simples de interpolação de frame introduzirão artefatos no frame reconstruído \tilde{X}_{2i} , especialmente para baixas taxas de bits, ou seja, para maiores taxas de compressão, o que pode ser observado nos resultados dos experimentos da seção 4.7.

Técnicas mais refinadas e complexas, por exemplo, as baseadas em estimação do movimento da seqüência de vídeo, são, portanto, essenciais na construção de uma informação lateral de alta qualidade, quando a similaridade entre os frames adjacentes não é muito alta, ou melhor, quando há uma grande quantidade de movimento de um frame X_j para o frame X_{j+1} . Com técnicas mais sofisticadas do que uma simples interpolação de frames, é possível obter nessas

situações um frame de informação lateral mais semelhante ao frame WZ e assim, minimizar a taxa de bits de síndrome necessária à decodificação, para uma mesma qualidade do frame reconstruído \tilde{X}_{2i} . A precisão do módulo de geração da informação lateral (ver figura 4.41) é, portanto, uma característica chave para o desempenho do codificador WZ-LIA proposto.

As técnicas de estimação de movimento, utilizadas na codificação tradicional de vídeo, tentam escolher a melhor predição para o frame atual. Em outras palavras, para um dado bloco no frame atual, as técnicas de estimação de movimento buscam encontrar o melhor casamento de blocos no frame de referência, independentemente do movimento verdadeiro do bloco na cena. Já para a interpolação de frames Key , estas técnicas não são tão bem adequadas, visto que neste caso o frame atual não é conhecido e por isso, é necessário estimar qual seria o movimento mais provável dos blocos dentre as possíveis hipóteses, conforme mostra a figura 4.40, para corretamente interpolar o frame ausente (frame atual), geralmente pela compensação do movimento entre os frames adjacentes.

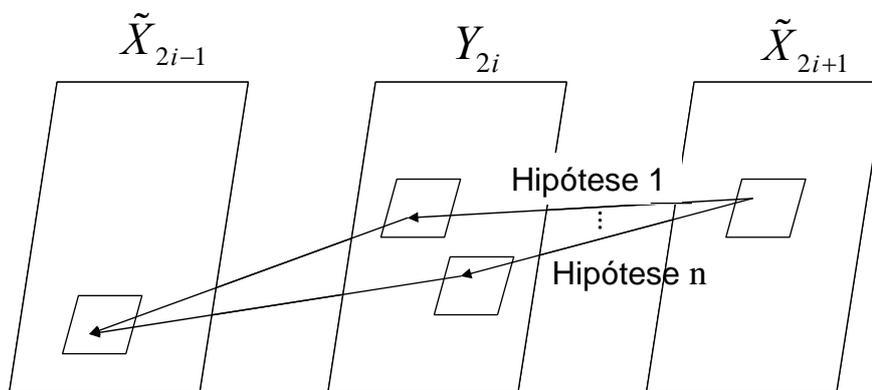


Figura 4.40: hipóteses de estimação de movimento do frame atual.

Assim, a figura 4.41 mostra a arquitetura proposta para o método de geração da informação lateral. Além da etapa de estimação “direta” do movimento, a qual é utilizada pelos codecs DVC mais avançados (*estado da arte*), outras técnicas foram adicionadas neste trabalho a fim de obter soluções mais eficientes de interpolação de frames compensados em movimento.

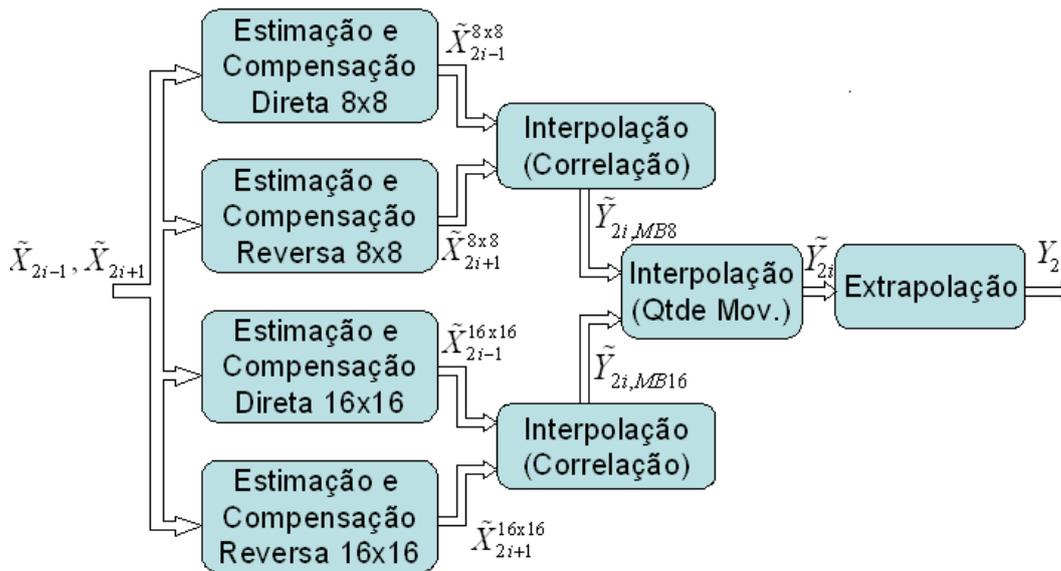


Figura 4.41: Processo de geração da informação lateral Proposto

Conforme ilustra a figura 4.41, o procedimento proposto neste trabalho para geração da informação lateral contém as seguintes etapas:

1. Estimação direta do movimento baseada em blocos 8x8 e compensação do movimento, gerando o frame compensado em movimento \tilde{X}_{2i-1}^{8x8} ;
2. Estimação reversa do movimento baseada em blocos 8x8 e compensação do movimento, gerando o frame compensado em movimento \tilde{X}_{2i+1}^{8x8} ;
3. Estimação direta do movimento baseada em blocos 16x16 e compensação do movimento, gerando o frame compensado em movimento \tilde{X}_{2i-1}^{16x16} ;
4. Estimação reversa do movimento baseada em blocos 16x16 e compensação do movimento, gerando o frame compensado em movimento \tilde{X}_{2i+1}^{16x16} ;
5. Interpolação baseada na medida de correlação entre os frames *key* e o frame WZ. Esta técnica é aplicada aos frames resultantes da compensação da estimativa bidirecional baseada em blocos 8x8 e baseada em blocos 16x16, resultando nos frames $\tilde{Y}_{2i,MB8}$ e $\tilde{Y}_{2i,MB16}$, respectivamente.

6. Interpolação ponderada baseada na quantidade de movimento estimada para a sequência de vídeo. \tilde{Y}_{2i}
7. Extrapolação em nível de pixels, produzindo a informação lateral Y_{2i} .

Em seguida, na seção 4.6.8, todos os módulos contidos nesse processo serão descritos em detalhes.

4.6.8. Método Proposto para Geração da Informação Lateral

4.6.8.1. Estimação Direta de Movimento

- ❖ A princípio, executa-se a estimação direta do movimento entre os frames *key* (adjacentes) anterior \tilde{X}_{2i-1} e posterior \tilde{X}_{2i+1} . Para realizar tal tarefa com maior eficiência, foram analisados vários algoritmos de casamento de bloco a fim de se adotar aqueles que mais se enquadram, em termos de eficiência, à estimação bidirecional de movimento, tanto para blocos de tamanho 8x8 quanto para 16x16. Os resultados dos experimentos com esses algoritmos são apresentados na seção 4.6.10.
- ❖ Um algoritmo de estimação de movimento é caracterizado, basicamente, pelo tamanho da janela de busca, faixa de busca e tamanho do passo de busca, conforme ilustra a figura 4.20. O tamanho da janela é a dimensão do bloco quadrado utilizado como unidade básica para executar a estimação do movimento e é fixa para o frame inteiro. Para a estimação direta (sentido convencional), por exemplo, a faixa de busca define a dimensão da área de \tilde{X}_{2i-1} , na qual busca-se o bloco mais similar ao bloco atual no frame \tilde{X}_{2i+1} . O tamanho do passo é o deslocamento mínimo da janela de busca (bloco), em pixels, no frame *key* anterior \tilde{X}_{2i-1} , onde se busca pelo vetor de movimento. Através da variação do passo de busca, temos a

capacidade de variar a complexidade computacional do método de estimação de movimento, pois aumentando ou diminuindo o tamanho do passo de busca, consegue-se prover uma aproximação mais grosseira ou mais precisa do movimento verdadeiro. A métrica adotada para o casamento de blocos na etapa de estimação de movimento foi a MAD (*Mean Absolute Difference* – Média do valor Absoluto das Diferenças), a qual é a medida de casamento de blocos mais utilizada nesse processo e bastante eficiente [50, 94].

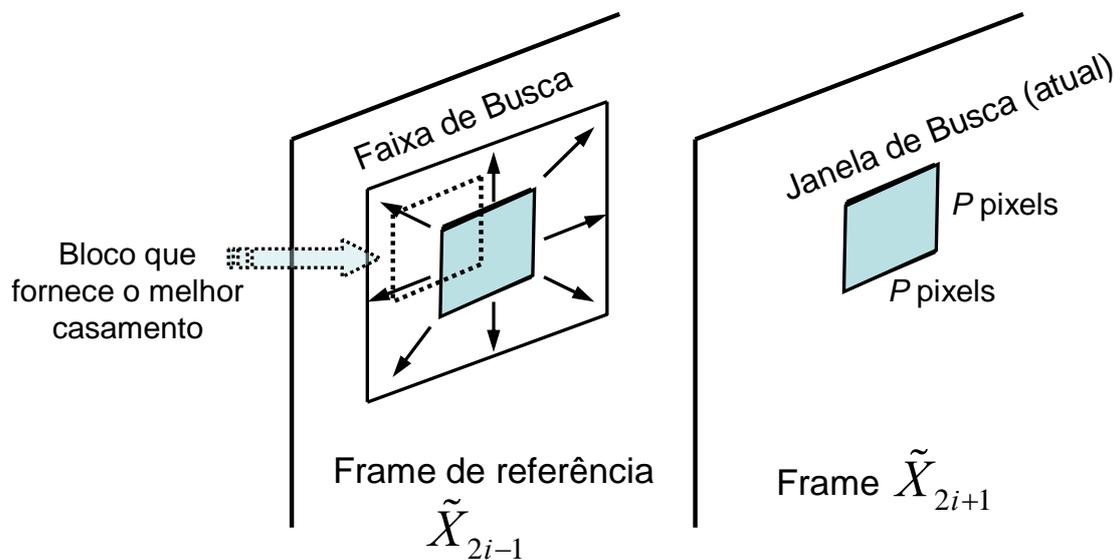
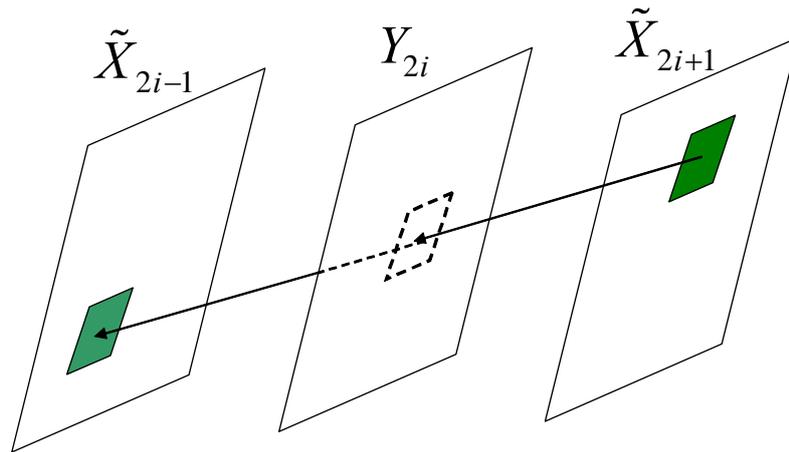


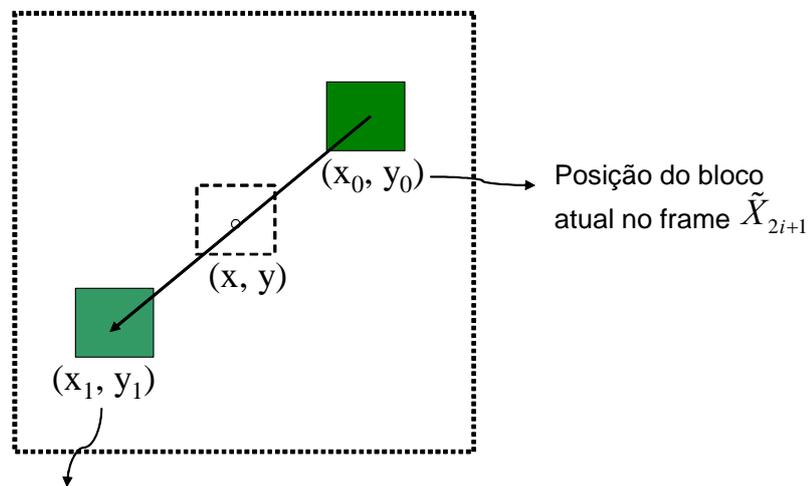
Figura 4.42: Componentes básicos do processo de estimação direta de movimento

- ❖ Na estimação direta do movimento, utiliza-se o frame \tilde{X}_{2i-1} para ser o frame de referência do frame \tilde{X}_{2i+1} e realiza-se a estimação do movimento, como é ilustrado na figura 4.43, obtendo-se os vetores de movimento para todos os blocos do frame \tilde{X}_{2i+1} .



(a)

Plano de projeção dos blocos no frame



Posição do bloco mais semelhante
(ao bloco atual) no frame \tilde{X}_{2i-1}

Posição do bloco
atual no frame \tilde{X}_{2i+1}

(b)

Figura 4.43: (a) estimaco direta do movimento passando pelo frame interpolado Y_{2i} . (b) projeo dos blocos utilizados na estimaco do movimento.

- ❖ Para fins de cculo, parte-se da hiptese que, em mdia, os trs pontos apresentados na figura 4.43(b) esto ligados por uma linha reta, assim, a movimentaco de um determinado bloco do frame \tilde{X}_{2i-1} at o frame \tilde{X}_{2i+1} no esto sujeita s variaoes observadas na figura 4.40. Portanto, o cculo realizado na construo do frame interpolado $\tilde{X}_{2i+1}^{P \times P}$ ($P \times P$  a dimenso do bloco usado na estimaco), em cada ponto (x, y) , a partir do frame resultante da compensaco do

movimento entre \tilde{X}_{2i-1} e \tilde{X}_{2i+1} é descrito abaixo. Para facilitar, o frame resultante da compensação de movimento será chamado aqui de Fc .

Fazendo $dx = x_1 - x_0$ e $dy = y_1 - y_0$, tem-se que:

$$x_1 = x_0 + dx, \quad y_1 = y_0 + dy \quad (4.40)$$

Assim, o frame interpolado na posição (x_0, y_0) , $Fc(x_0, y_0)$, pode ser expresso por:

$$Fc(x_0, y_0) = \tilde{X}_{2i-1}(x_1, y_1) = \tilde{X}_{2i-1}(x_0 + dx, y_0 + dy) \quad (4.41)$$

Como o ponto (x, y) está localizado no ponto médio da semi-reta que vai do ponto (x_0, y_0) até (x_1, y_1) , tem-se que:

$$x_1 = x + dx/2, \quad y_1 = y + dy/2 \quad (4.42)$$

Com base na eq. (4.42), podemos definir o frame interpolado como:

$$\tilde{X}_{2i+1}^{PxP}(x, y) = \tilde{X}_{2i-1}(x_1, y_1) = \tilde{X}_{2i-1}(x + dx/2, y + dy/2) \quad (4.43)$$

Substituindo a eq. (4.41) em (4.43), chega-se ao resultado:

$$\tilde{X}_{2i+1}^{PxP}(x, y) = Fc(x_0, y_0) = Fc(x - dx/2, y - dy/2) \quad (4.44)$$

logo, usando o fato de que $x = x_0 + dx/2$ e $y = y_0 + dy/2$ e substituindo x e y na eq. (4.44), chega-se em:

$$\tilde{X}_{2i+1}^{PxP}(x_0 + dx/2, y_0 + dy/2) = Fc(x_0, y_0) \quad (4.45)$$

- ❖ Esta eq. (4.45) mostra como obter o frame interpolado \tilde{X}_{2i+1}^{PxP} a partir do frame resultante do processo normal de compensação de movimento, Fc . Note que o frame compensado Fc na codificação convencional de vídeo representaria uma predição do frame \tilde{X}_{2i+1} .

- ❖ Ao contrário do processo de compensação do movimento, no qual há um único ponto (x_l, y_l) no frame de referência associado ao ponto (x_0, y_0) no frame atual, no caso do frame interpolado pode haver vários pontos associados a um único ponto (x, y) , ou seja, pode haver mais de uma estimativa com o vetor de movimento passando pelo centro de um mesmo bloco j . Esta situação é ilustrada pela figura 4.44.

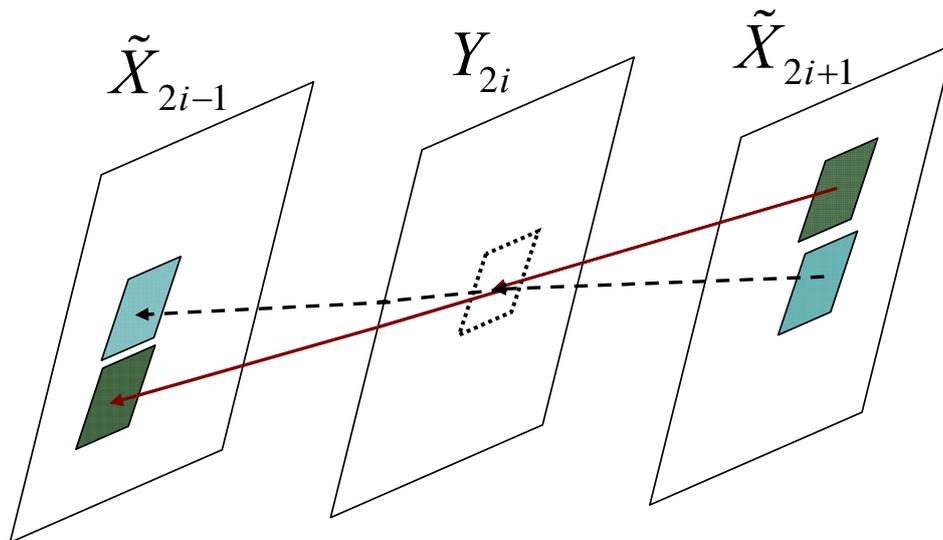


Figura 4.44: interpolação a partir do processo de compensação do movimento, apresentando várias estimativas associadas a um único ponto do frame interpolado.

- ❖ Neste trabalho, foi proposto o algoritmo (4.2) abaixo para gerar o frame interpolado \tilde{X}_{2i+1}^{PxP} a partir do frame compensado em movimento. A diferença principal entre o algoritmo proposto (4.2) e os métodos existentes comumente utilizados está no cálculo do frame interpolado, o qual é dado por:

$$\tilde{X}_{2i+1}^{PxP}(x_p + (dx/2), y_p + (dy/2)) = (1 - 1/Sb) \tilde{X}_{2i+1}^{PxP}(x_p + (dx/2), y_p + (dy/2)) + (1/Sb) \tilde{X}_{2i-1}(x_p, y_p)$$

onde Sb representa o fator de ponderação, o qual está relacionado à quantidade de sobreposições de blocos interpolados e que é melhor compreendido no contexto do algoritmo (4.4).

Algoritmo 4.4: cálculo do frame interpolado

// Dados os vetores de movimentos (MV) encontrados na etapa da estimação // de movimento:

$MV = [MV_1 \ MV_2 \ \dots \ MV_n]$; //n é o número de macroblocos P x P no //frame.

$MV_j = (dx_j, dy_j)$; // MV para cada bloco j

Sb = uns(row, col) //Sb = "sobreposição" = matriz de uns com // dimensão igual a do frame: row.col

num_blocos = row.col/P²;

Para¹ j = 1:num_blocos //j = bloco atual

Para² p = 1:P² //p = índice do pixel no bloco j

l = x_p + dx_j; //l = índice da linha

c = y_p + dy_j; //c = índice da coluna

Se³ ((x_p+dx/2) ≥ 1 & (x_p+dx/2) ≤ row & (y_p+dy/2) ≥ 1 & (y_p+dy/2) ≤ col) //se estes pontos não pertencem às //bordas

Se⁴ $\tilde{X}_{2i+1}^{PxP}(x_p + (dx/2), y_p + (dy/2)) = 0$

$\tilde{X}_{2i+1}^{PxP}(x_p + (dx/2), y_p + (dy/2)) = \tilde{X}_{2i-1}(x_p, y_p)$;

Caso contrário:

Sb(x_p, c) = Sb(x_p, c) + 1; //conta mais uma sobreposição

$\tilde{X}_{2i+1}^{PxP}(x_p + (dx/2), y_p + (dy/2)) = (1/Sb) \cdot \tilde{X}_{2i-1}(x_p, y_p) +$

$(1-1/Sb) \cdot \tilde{X}_{2i+1}^{PxP}(x_p + (dx/2), y_p + (dy/2))$;

//cálculo de média ponderada dos pixels sobrepostos

Fim⁴

Fim³

Fim²

Fim¹

4.6.8.2.**Estimação Reversa do Movimento**

- ❖ A estimação reversa é realizada seguindo o mesmo procedimento utilizado na estimação direta do movimento, porém, o sentido de todos os vetores de movimento e também as direções dos mesmos são diferentes, pois o fato de um determinado bloco B no frame \tilde{X}_{2i-1} ser o melhor casamento do bloco A pertencente ao frame \tilde{X}_{2i+1} , não necessariamente implica que o bloco A seja o melhor casamento de B (estimação reversa), podendo, por exemplo, ser um bloco C qualquer, pertencente a \tilde{X}_{2i+1} , conforme ilustra a figura 4.45. Assim, deve-se realizar uma nova estimação dos vetores de movimento e uma correspondente compensação para cada bloco.

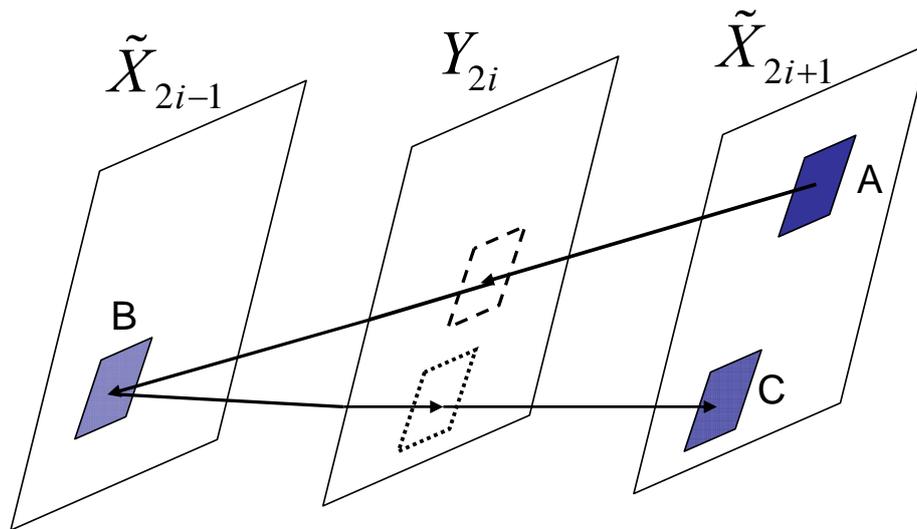


Figura 4.45: Sentido e direção dos vetores de movimento na estimação reversa.

- ❖ No cenário de codificação da figura 4.41, os pixels interpolados, isto é, os pixels de Y_{2i} , não são conhecidos, em oposição ao que acontece no modo de codificação dos frames B . Logo, uma diferente técnica de estimação é utilizada na arquitetura do codificador WZ-LIA proposto. Conforme mostra a figura 4.43, a técnica de estimação de movimento adotada seleciona uma trajetória reta entre os frames *key* posterior e anterior, X_{2i+1} e X_{2i-1} , respectivamente, passando pelo centro dos blocos interpolados.

4.6.9.

Avaliação dos Métodos de Estimação Bidirecional de Movimento

Neste experimento, pretende-se avaliar a eficiência dos métodos de estimação mais promissores aplicados à estimação bidirecional utilizada em DVC, em relação ao método que proporciona a maior PSNR, que é a técnica da “Busca Exaustiva”, apesar de esta técnica ser a que demanda maior carga computacional (maior tempo de processamento). As técnicas de estimação de movimento a serem avaliadas são: *Exhaustive Search* (ES), *Three Step Search* (TSS), *New Three Step Search* (NTSS), *Simple and Efficient TSS* (SES), *Four Step Search* (aqui chamado de QSS), *Diamond Search* (DS) e *Adaptive Rood Pattern Search* (ARPS), as quais são descritas em [94]. Todas as técnicas são disponibilizadas no codec proposto para sua utilização no módulo de geração da informação lateral. Assim, para dois

QPs diferentes (QP=40 e QP=26) e dois tamanhos de bloco de estimação diferentes (macrobloco 16x16 e 8x8), as figuras 4.46 a 4.61 apresentam a PSNR da informação lateral, ao longo da sequência de vídeo, correspondente a cada frame WZ.

Observando-se as figuras 4.46 a 4.53, as quais utilizaram o bloco 8x8 para estimação de movimento, nota-se que o método que resulta em uma maior qualidade (PSNR) da informação lateral, depois do método de busca exaustiva (ES), é o *Diamond Search* (DS) e depois o *Adaptive Rood Pattern Search* (ARPS). Além disso, o número de operações destes dois métodos é quase o mesmo, logo, o método DS será adotado como preferencial para estimação com blocos 8x8. Enquanto o método ES realiza, em média, aproximadamente 205 buscas por macrobloco 8x8, analisando as figuras 4.52 e 4.53, nota-se que o DS e o ARPS levam menos de um sexto desse valor, mostrando que, além de eficientes, são muito mais rápidos. O mesmo pode ser observado em relação à estimação 16x16, em que o método ES realiza, em média, mais de 200 buscas, por meio das figuras 4.60 e 4.61, observa-se que o DS e o ARPS levam menos de um sexto desse valor.

Entretanto, analisando as figuras 4.54 a 4.61, percebe-se que o método que resulta em uma maior qualidade da informação lateral é o ARPS e depois o DS, por isso, será adotado o método ARPS para estimação bidirecional de movimento com macroblocos 16x16 e o método DS para estimação com blocos 8x8 no processo de geração da informação lateral do codec proposto neste trabalho.

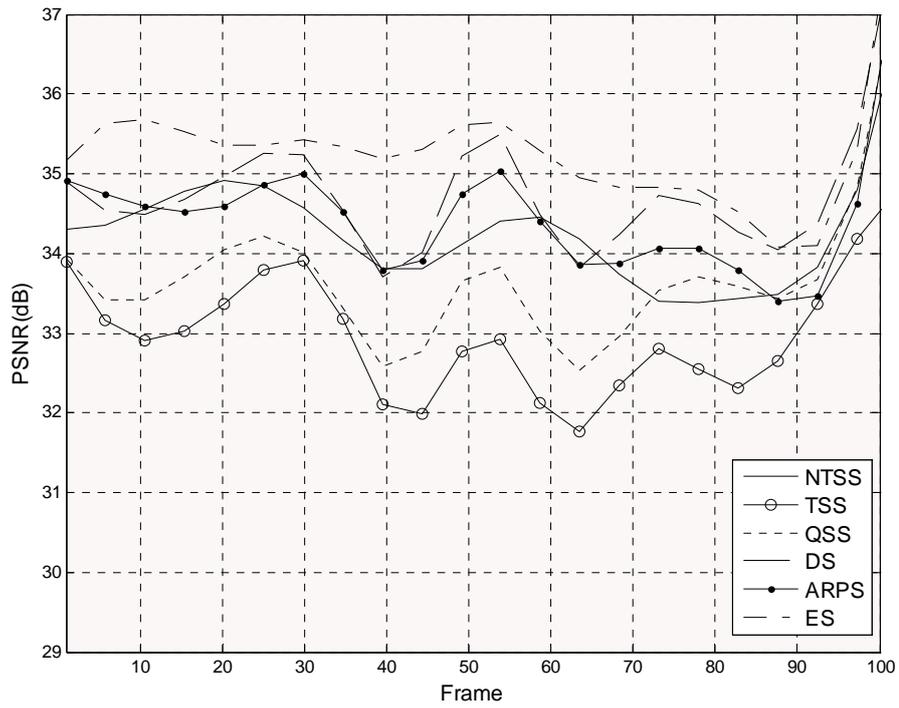


Figura 4.46: PSNR da informação lateral ao longo da sequência para diferentes métodos de estimação de movimento, sequência *Foreman*, com QP = 40 (intra), macrobloco 8x8.

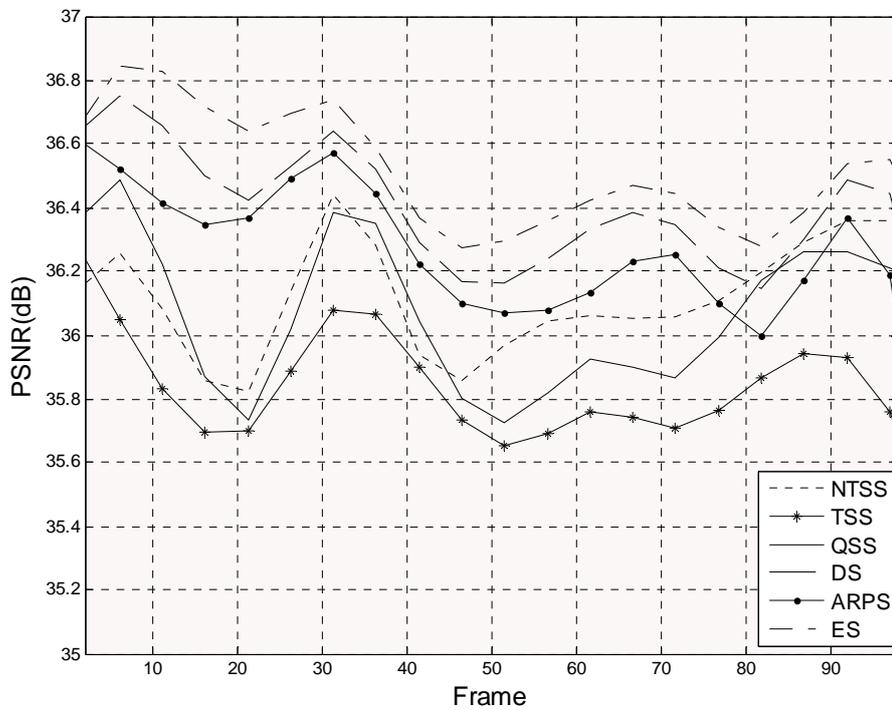


Figura 4.47: PSNR da informação lateral ao longo da sequência para diferentes métodos de estimação de movimento, sequência *News*, com QP = 40 (intra), macrobloco 8x8.

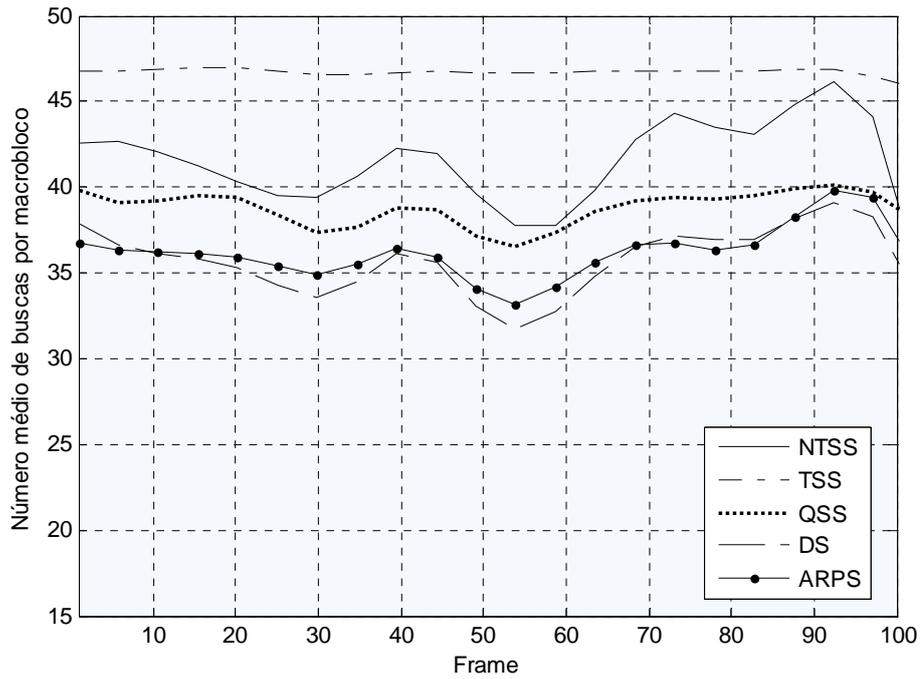


Figura 4.48: Número médio de buscas por macrobloco, para diferentes métodos, para a sequência *Foreman*, com QP = 40 (intra), macrobloco 8x8.

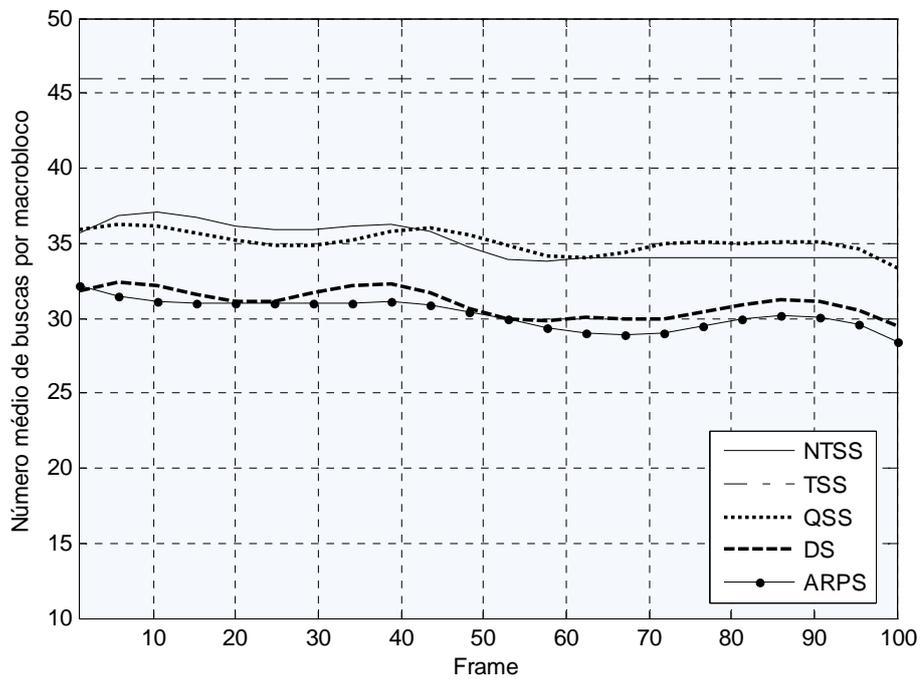


Figura 4.49: Número médio de buscas por macrobloco, para diferentes métodos, para a sequência *News*, com QP = 40 (intra), macrobloco 8x8.

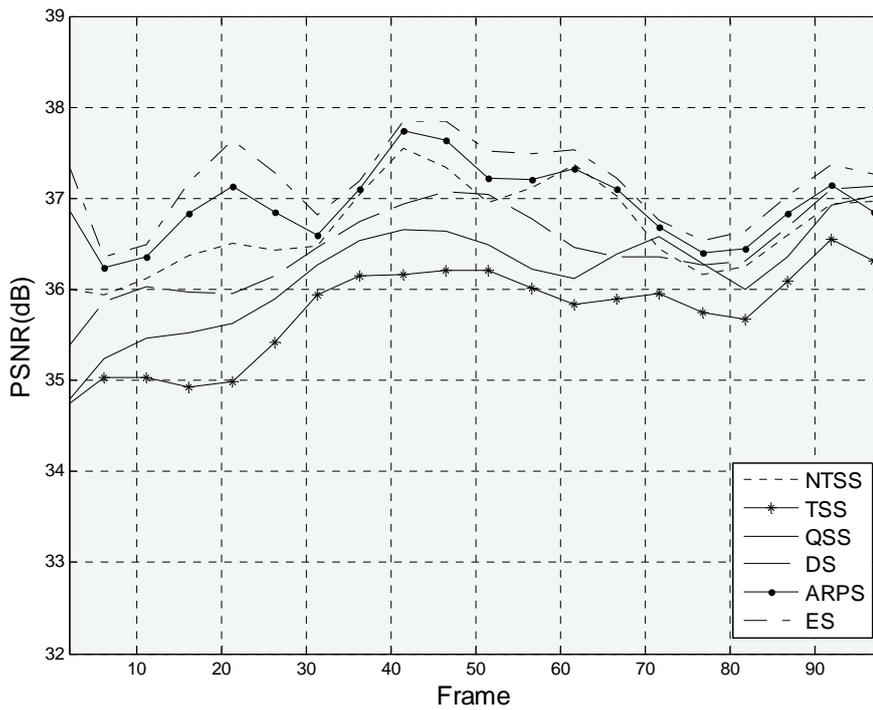


Figura 4.50: PSNR da informação lateral ao longo da sequência para diferentes métodos de estimação de movimento, sequência *Foreman*, com QP = 26 (intra), macrobloco 8x8.

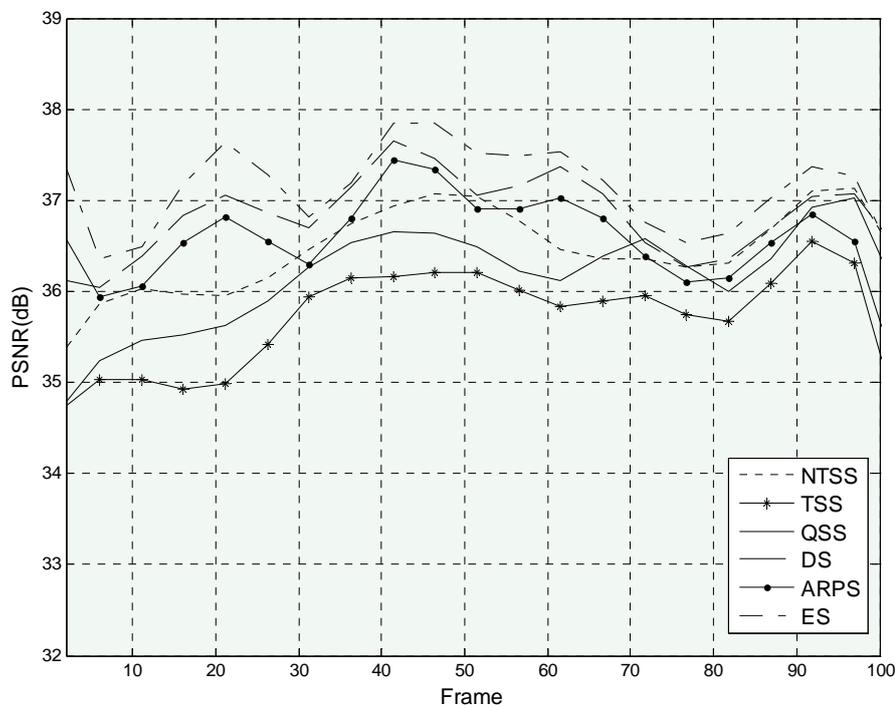


Figura 4.51: PSNR da informação lateral ao longo da sequência para diferentes métodos de estimação de movimento, sequência *News*, com QP = 26 (intra), macrobloco 8x8.

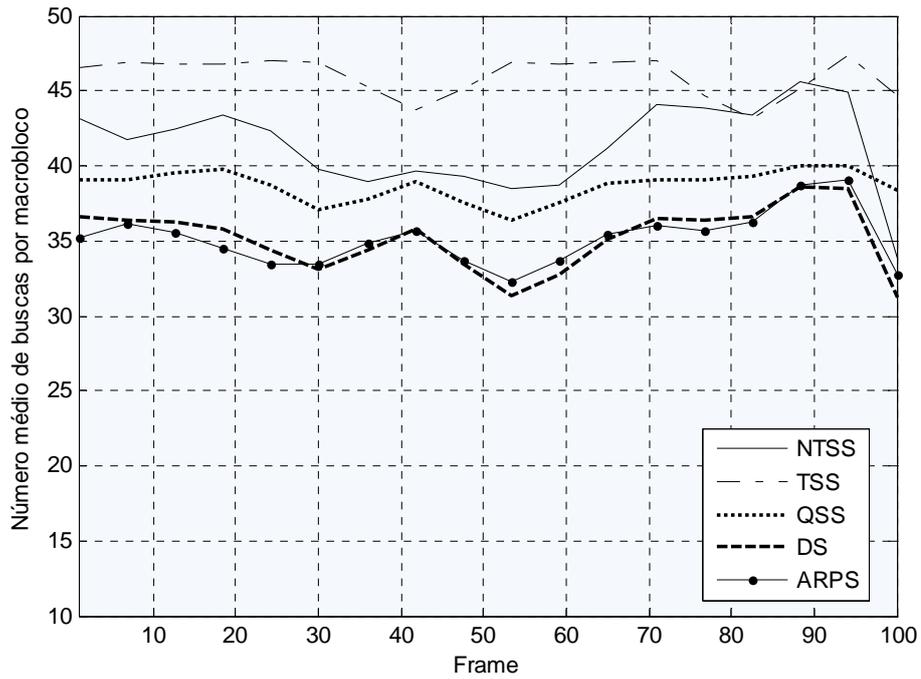


Figura 4.52: Número médio de buscas por macrobloco, para diferentes métodos, para a sequência *Foreman*, com QP = 26 (intra), macrobloco 8x8.

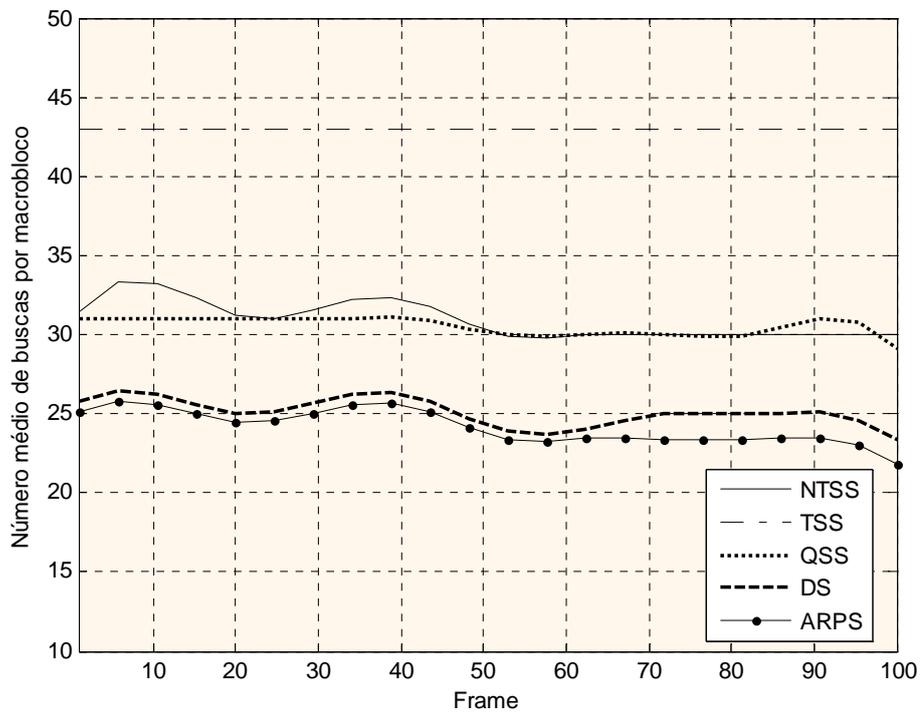


Figura 4.53: Número médio de buscas por macrobloco, para diferentes métodos, para a sequência *News*, com QP = 26 (intra), macrobloco 8x8.

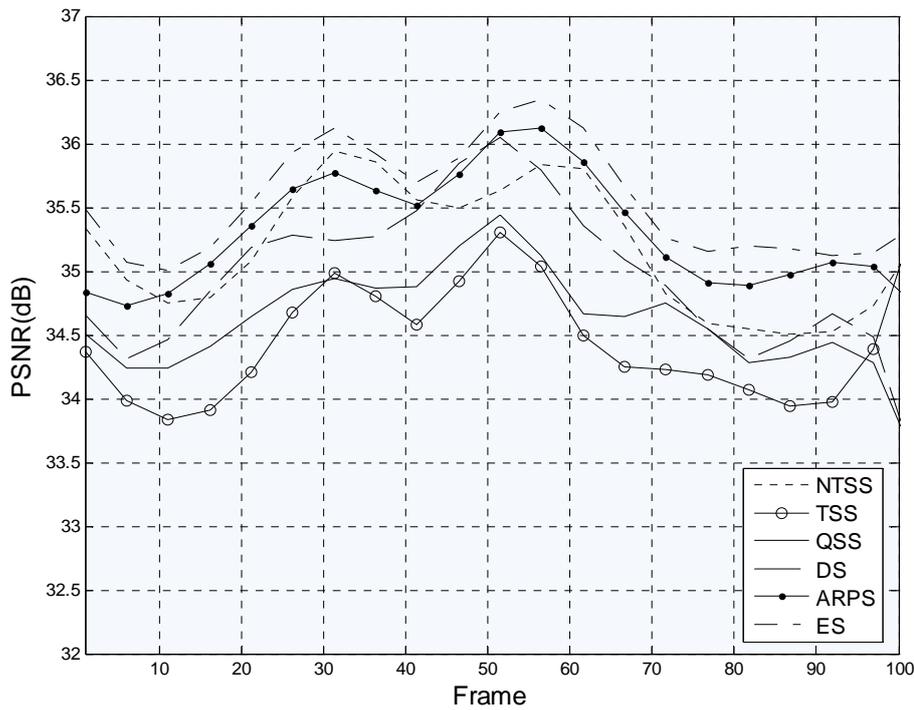


Figura 4.54: PSNR da informação lateral para diferentes métodos de estimação de movimento, sequência *Foreman*, com QP = 40 (intra), macrobloco 16x16.

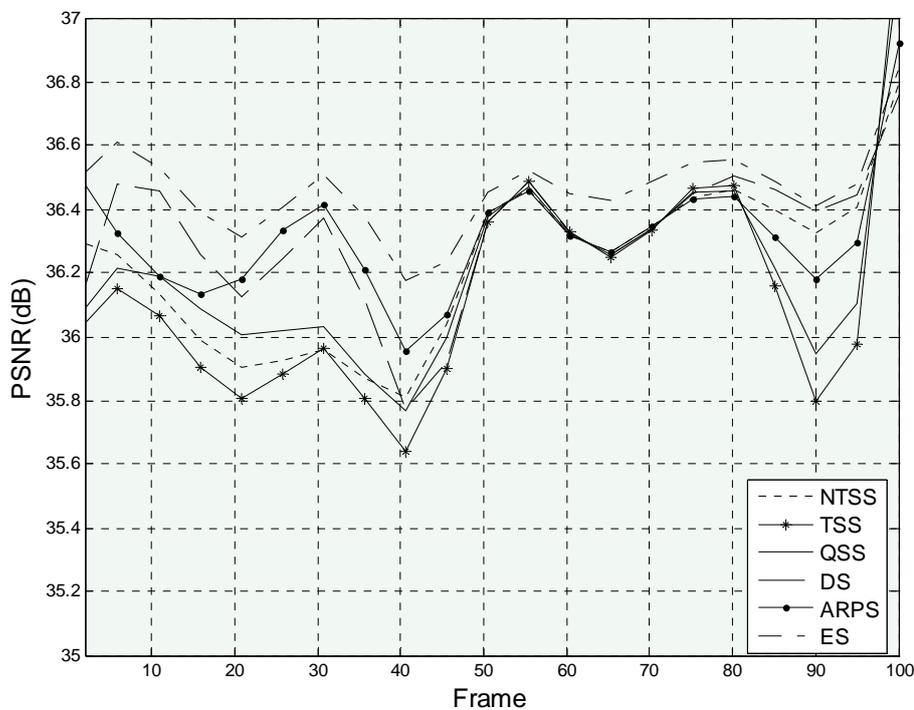


Figura 4.55: PSNR da informação lateral ao longo da sequência para diferentes métodos de estimação de movimento, sequência *News*, com QP = 40 (intra), macrobloco 16x16.

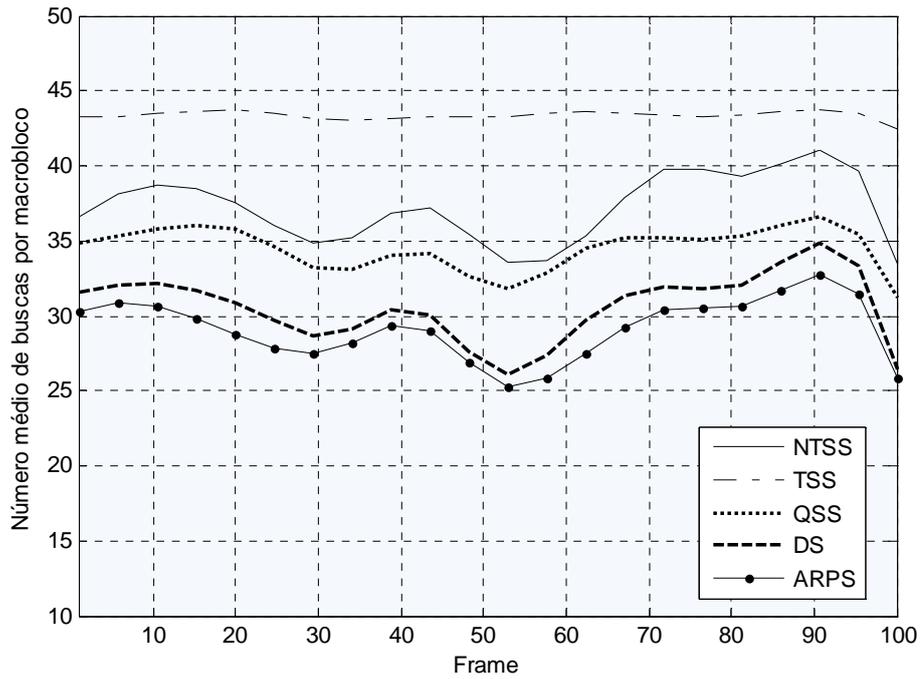


Figura 4.56: Número médio de buscas por macrobloco, para diferentes métodos, para a sequência *Foreman*, com QP = 40 (intra), macrobloco 16x16.

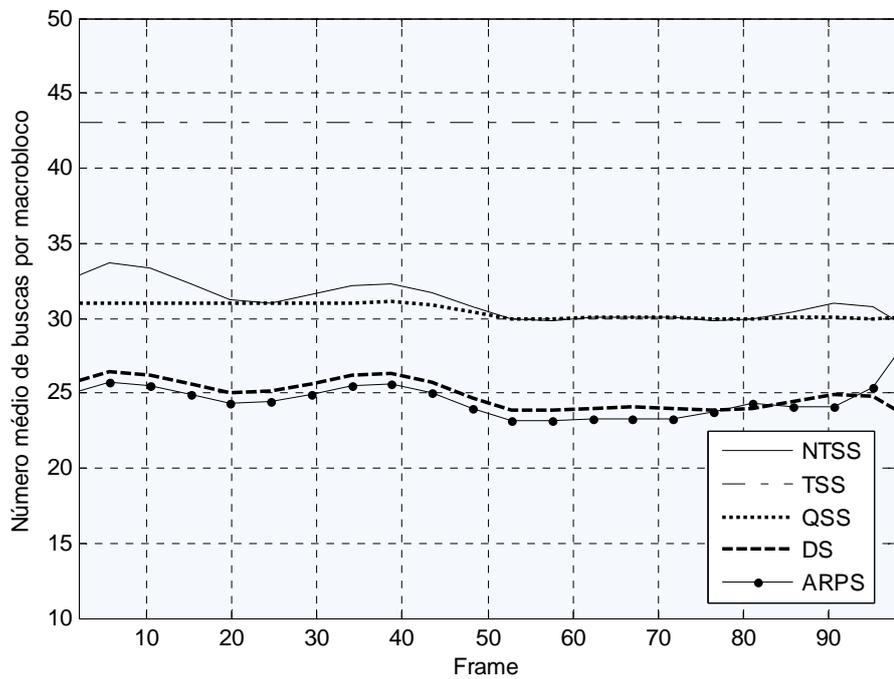


Figura 4.57: Número médio de buscas por macrobloco, para diferentes métodos, para a sequência *News*, com QP = 40 (intra), macrobloco 16x16.

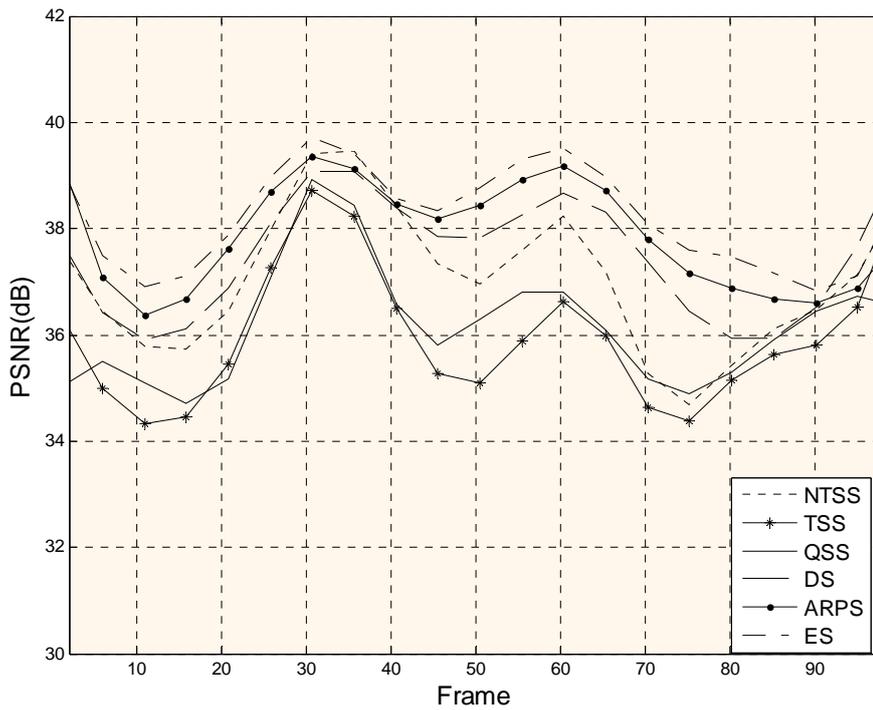


Figura 4.58: PSNR da informação lateral para diferentes métodos de estimação de movimento, sequência *Foreman*, com QP = 26 (intra), macrobloco 16x16.

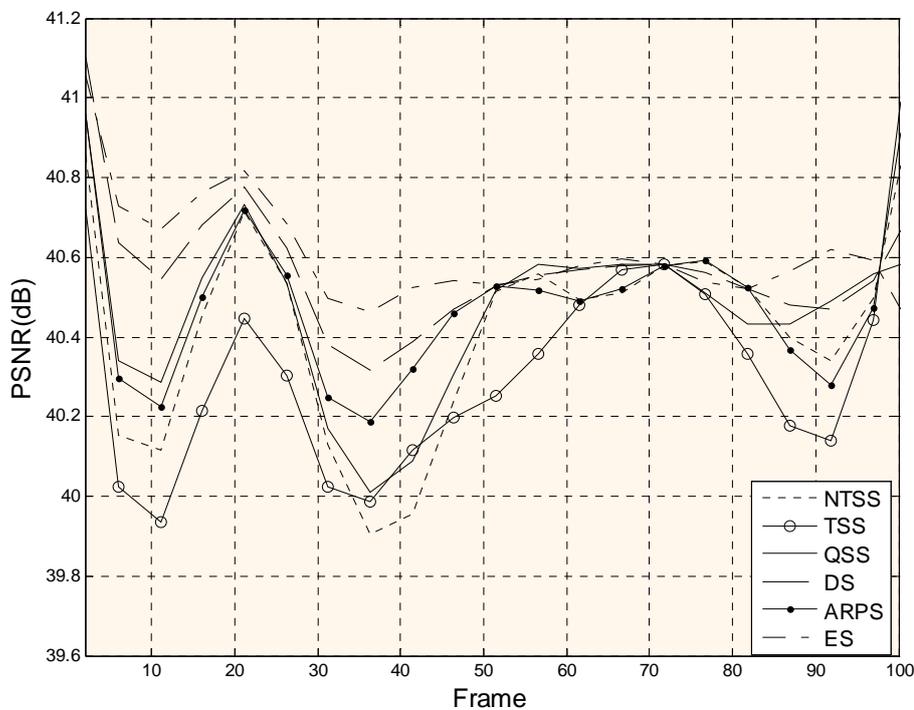


Figura 4.59: PSNR da informação lateral ao longo da sequência para diferentes métodos de estimação de movimento, sequência *News*, com QP = 26 (intra), macrobloco 16x16.

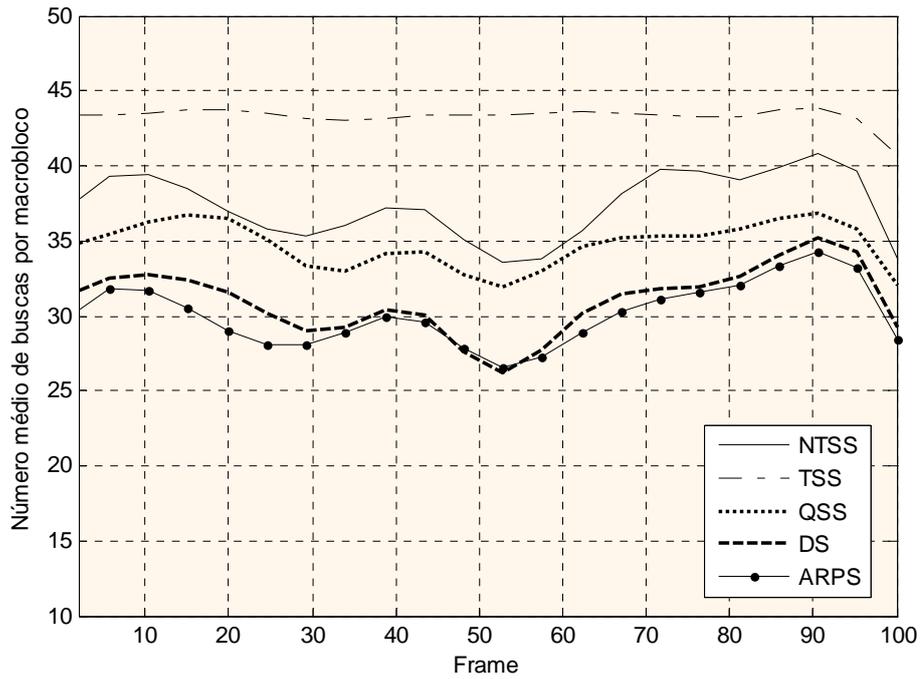


Figura 4.60: Número médio de buscas por macrobloco, para diferentes métodos, para a sequência *Foreman*, com QP = 26 (intra), macrobloco 16x16.

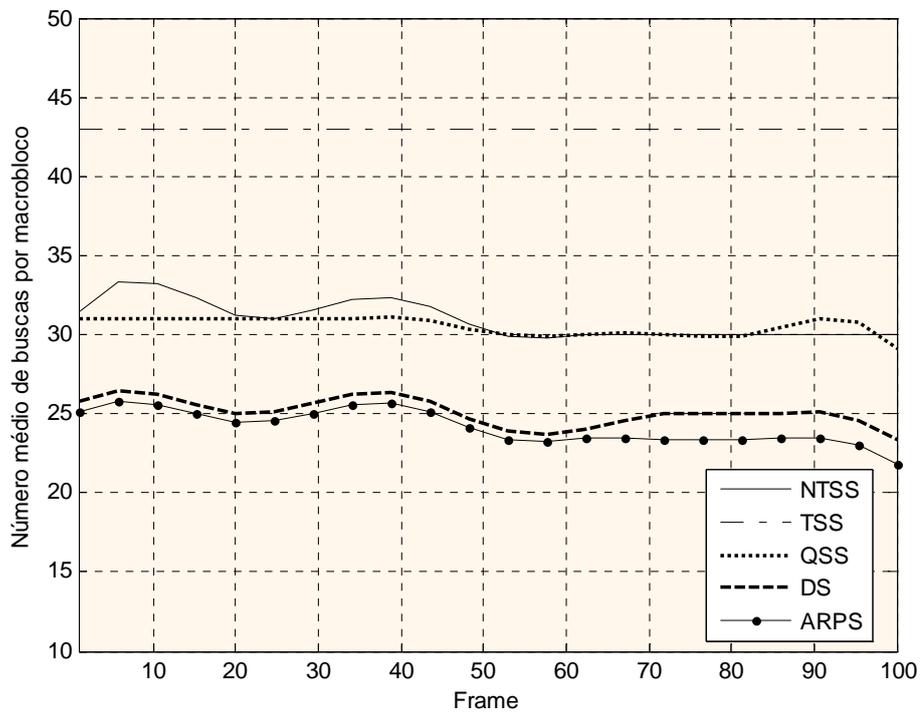


Figura 4.61: Número médio de buscas por macrobloco, para diferentes métodos, para a sequência *News*, com QP = 26 (intra), macrobloco 16x16.

4.6.9.1. Problemas da Compensação de Frames Interpolados

No caso onde se necessita estimar o movimento mais provável de um determinado bloco pertencente a um frame “desconhecido”, o algoritmo de estimação pode falhar na captura de todos os aspectos do campo de movimento, pois quando é realizada uma interpolação de frame, áreas sobrepostas e áreas descobertas aparecerão no frame interpolado (estimado). Isto porque os vetores de movimento obtidos não necessariamente interceptam o frame interpolado no centro de cada bloco não-sobreposto. A figura 4.24 [70] ilustra esse problema, mostrando também que os pixels entre dois blocos vizinhos no frame interpolado não foram preenchidos com qualquer textura (pixels de valor zero).

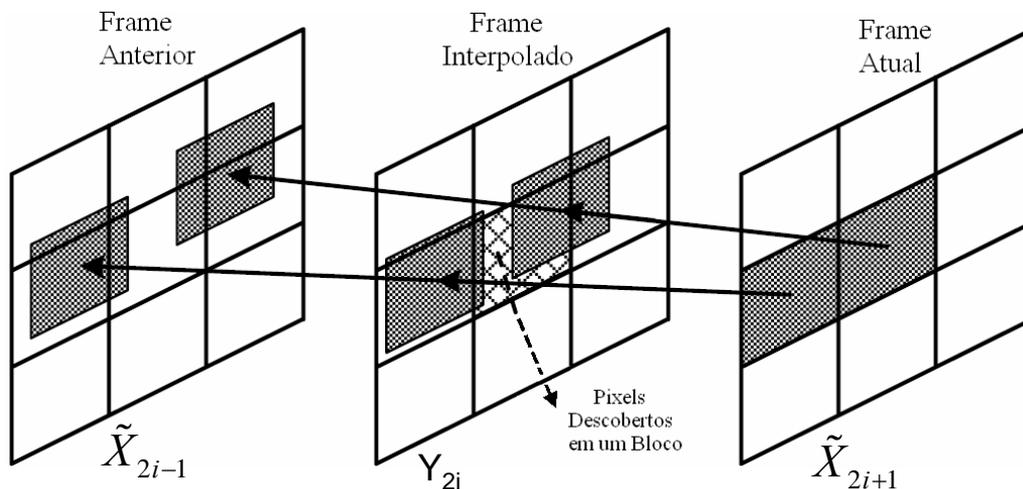


Figura 4.62: Pixels descobertos no frame interpolado devido ao fato da estimação de movimento não partir do centro dos blocos do frame interpolado [70].

- ❖ O frame resultante da estimação e compensação do movimento tem, em geral, um aspecto semelhante ao ilustrado pela figura 4.63, onde se observa que há muitos pixels descobertos no frame e partes de blocos sobrepostas. Um exemplo prático de um frame gerado segundo o processo de estimação mencionado acima é apresentado no experimento da seção 4.6.10.

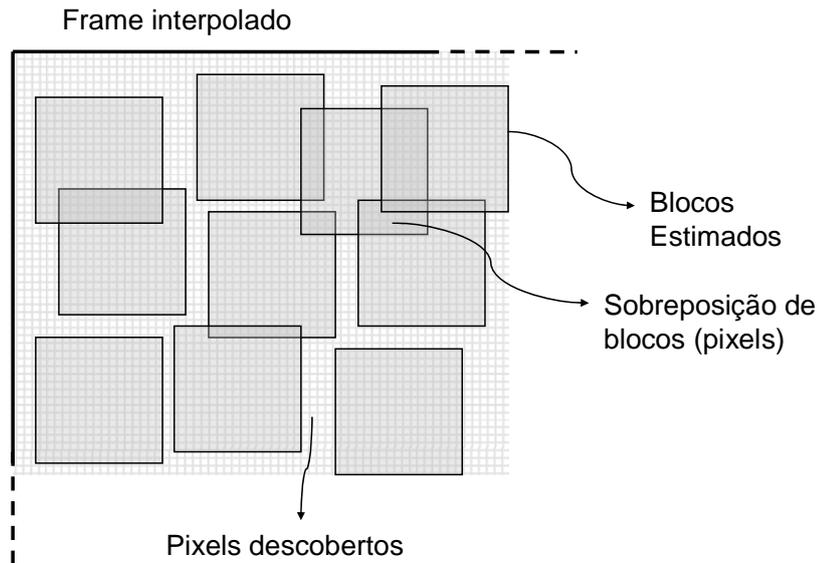


Figura 4.63: Exemplo de disposição de blocos resultantes da estimação e compensação do movimento no frame interpolado.

A fim de minimizar o número de pixels descobertos, realiza-se uma segunda estimação bidirecional do movimento, desta vez, no entanto, baseada em blocos 16x16 ao invés de blocos 8x8. Utilizando-se blocos de tamanho 16x16 em vez de 8x8, embora gere resultados aparentemente menos precisos de estimação de movimento, consegue-se um maior preenchimento do frame interpolado, isto é, haverá menos pixels descobertos. O resultado comparativo entre os frames interpolados, utilizando-se blocos 8x8 e 16x16 na estimação de movimento é ilustrado pelas figuras 4.64(a) e 4.64(b), respectivamente.

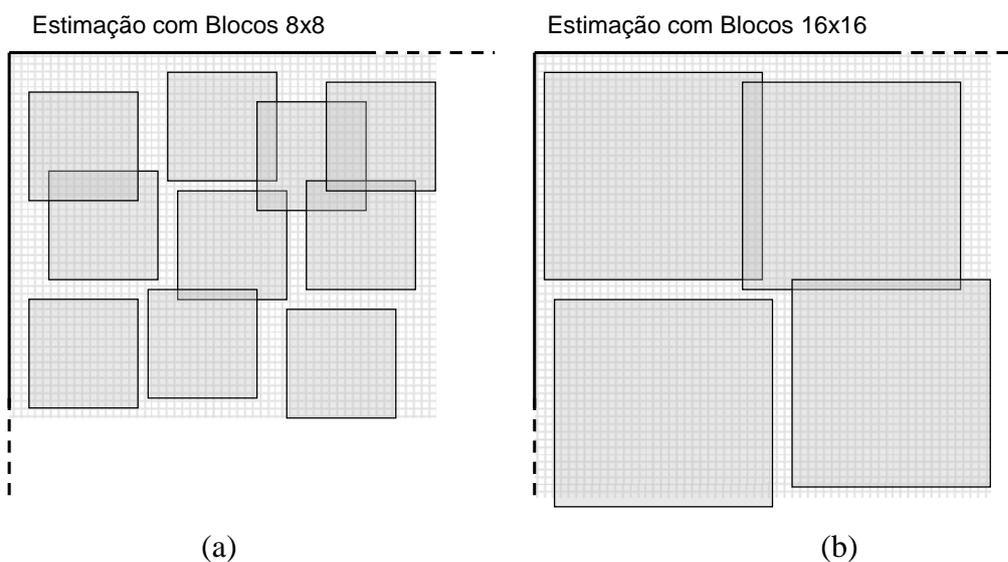


Figura 4.64: Disposição dos blocos resultantes da compensação de movimento, no frame interpolado: (a) utilizando blocos 8x8; (b) utilizando blocos 16x16.

4.6.9.2. Interpolação com base na Correlação entre Frames

Os dois módulos de interpolação de frames foram propostos pelo autor deste trabalho com o objetivo de aumentar a qualidade do frame resultante da etapa de estimação e compensação do movimento, através do cálculo de valores mais precisos para os pixels do frame interpolado e através de um maior preenchimento das áreas descobertas. O primeiro módulo de interpolação de frames é baseado em uma medida de similaridade entre os frames *key* e o frame WZ, daí a utilização da expressão correlação entre frames.

A medida de correlação escolhida pelo autor deste trabalho para medir a similaridade entre os frames *key* e o frame WZ foi a MAD (Média do valor Absoluto das Diferenças), a qual é a medida de casamento de blocos mais utilizada conforme foi dito anteriormente e, por ser de baixa exigência computacional [50, 51, 94], é bem adequada para ser utilizada no codificador, onde o frame WZ original está disponível. Na MAD, os pixels correspondentes em cada frame são comparados e os módulos de suas diferenças são somados e o resultado é dividido pelo número de pixels do frame. Assim, calcula-se a MAD entre o frame X_{2i-1} e o frame X_{2i} , e o resultado é armazenado na variável M_1 ; depois se calcula M_2 , que é a MAD entre o frame X_{2i+1} e o frame X_{2i} . A formulação matemática para cálculo de M_1 e M_2 é descrita pelas equações abaixo:

$$M_1 = \frac{1}{r \cdot c} \sum_{m=1}^r \sum_{n=1}^c |X_{2i-1}(m, n) - X_{2i}(m, n)| \quad (4.46)$$

$$M_2 = \frac{1}{r \cdot c} \sum_{m=1}^r \sum_{n=1}^c |X_{2i+1}(m, n) - X_{2i}(m, n)| \quad (4.47)$$

onde $r \cdot c$ é a dimensão do frame e $X_{2i}(m, n)$ é o valor do pixel na m -ésima linha e n -ésima coluna de X_{2i} . Uma menor MAD entre dois frames implica em uma maior similaridade entre suas amostras. Perceba que este cálculo deve ser realizado no codificador, porque além do codificador dispor de valores originais dos frames da sequência de vídeo, o decodificador não possui o frame WZ original. Outro ponto importante é que por causa de sua baixa complexidade, o

cálculo da MAD pode ser realizado no codificador sem “quebrar” o paradigma da baixa complexidade do codificador DVC.

Depois de M_1 e M_2 serem calculadas, calcula-se o valor relativo de M_1 em relação M_2 , denominado m_1 , conforme a eq. (4.48):

$$m_1 = M_1 / (M_1 + M_2) \quad (4.48)$$

Assim, este valor percentual m_1 é enviado ao decodificador a fim de ajudar no processo de interpolação ponderada, pois somente é necessário enviar um valor relativo da MAD, porque o segundo valor é calculado a partir do primeiro. Partindo desses princípios, a fórmula de interpolação de frames com base na medida de similaridade MAD é descrita abaixo:

$$Y_{2i,MB8} = m_1.F_{DIR,MB8}^{Comp} + (1 - m_1).F_{REV,MB8}^{Comp} \quad (4.49)$$

$$Y_{2i,MB16} = m_1.F_{DIR,MB16}^{Comp} + (1 - m_1).F_{REV,MB16}^{Comp} \quad (4.50)$$

onde $F_{DIR,MB8}^{Comp}$ e $F_{REV,MB8}^{Comp}$ são os frames compensados derivados da estimação direta e reversa, respectivamente, usando blocos 8x8. Além desse cálculo, no algoritmo proposto, caso um dos dois frames interpolados possua um pixel de valor zero, o pixel correspondente no frame interpolado assumirá um valor ponderado do pixel não zero pertencente ao outro frame interpolado.

4.6.9.3. Interpolação baseada na Estimativa de movimento

O algoritmo (4.3) de interpolação de frames com base na estimativa subjetiva de quantidade de movimento da sequência para cálculo da segunda etapa de interpolação, o qual resulta no frame \tilde{Y}_{2i} , é descrito abaixo. Neste algoritmo, a variável qm representa um coeficiente de quantidade de movimento da sequência, a qual assume valores de 1 a 5 para sequências com baixa e alta quantidade média de movimentos, respectivamente. Este coeficiente é baseado na avaliação subjetiva a priori da quantidade de movimento da sequência a ser codificada

Algoritmo 4.5: interpolação baseada na quantidade de movimento

```

Para1 r=1:num_linhas
  Para2 c=1:num_colunas
    Se3 ( $Y_{2i,MB8}(r,c)=0$  &  $Y_{2i,MB16}(r,c)\neq 0$ )
       $\tilde{Y}_{2i}(r,c)=Y_{2i,MB16}(r,c)$ ;
    caso contrário, se ( $Y_{2i,MB8}(r,c)\neq 0$  &  $Y_{2i,MB16}(r,c)=0$ )
       $\tilde{Y}_{2i}(r,c)=Y_{2i,MB8}(r,c)$ ;
    senão
       $\tilde{Y}_{2i}(r,c)=\text{round}((Y_{2i,MB8}(r,c)*(1-0,05qm)$ 
         $+Y_{2i,MB16}(r,c))/(2-0,05qm))$ ;
  fim3
fim2
fim1

```

4.6.9.4. Extrapolação

O algoritmo 4.6 descreve o procedimento de extrapolação de pixels, com o objetivo de cobrir áreas não preenchidas no frame interpolado (estimado). O método posposto neste trabalho para preencher os pixels descobertos é o seguinte: busca-se expandir cada bloco interpolado até preencher as áreas descobertas ao seu redor; caso uma parte da expansão de um bloco se sobreponha a outro bloco (extrapolado ou não), é realizada uma média ponderada para calcular o valor do pixel resultante nesta área. A figura 4.65 ilustra este procedimento.

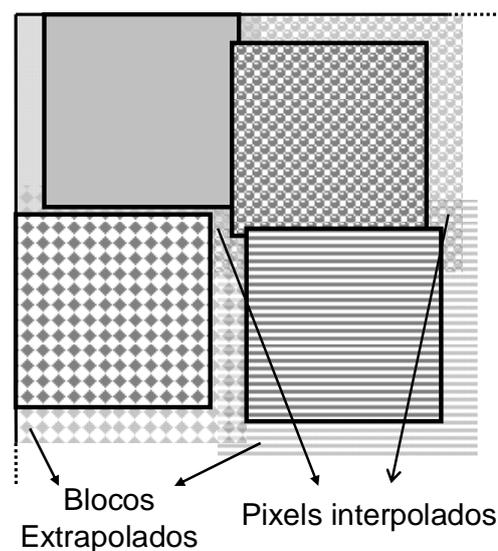


Figura 4.65: Método de extrapolação proposto neste trabalho.

Algoritmo 4.6: extrapolação de pixels

```

f =  $\tilde{Y}_{2i}$ ;
Para1 r=1:num_linhas
  Para2 c=1:num_colunas
    Se3 f(r,c) == 0
      %===== PIXELS NAS QUINAS =====
      Se4 (r==1 && c==1) %/////canto superior esquerdo
        f(r,c)=ceil((f(r,c+1)+f(r+1,c)+f(r+1,c+1))/3);
      Caso contrário, se (r==1 && c==176) %canto superior direito
        f(r,c)=ceil((f(r,c-1)+f(r+1,c-1)+f(r+1,c))/3);
      Caso contrário, se (r==144 && c==1) %canto inferior esquerdo
        f(r,c)=ceil((f(r-1,c)+f(r-1,c+1)+f(r,c+1))/3);
      Caso contrário, se (r==144 && c==176) %canto inferior direito
        f(r,c)=ceil((f(r-1,c-1)+f(r-1,c)+f(r,c-1))/3);
      %===== PIXELS DAS BORDAS =====
      Caso contrário, se (r==1 && c>1 && c<176) %borda superior
        f(r,c)=ceil((f(r,c-1)+f(r,c+1)+f(r+1,c-1)+f(r+1,c)+
          f(r+1,c+1))/5);
      Caso contrário, se (r==144 && c>1 && c<176) %borda inferior
        f(r,c)=ceil((f(r-1,c-1)+f(r-1,c)+f(r-1,c+1)+f(r,c-1)+
          f(r,c+1))/5);
      Caso contrário, se (c==1 && r>1 && r<144) %borda esquerda
        f(r,c)=ceil((f(r-1,c)+f(r-1,c+1)+f(r,c+1)+f(r+1,c)+
          f(r+1,c+1))/5);
      Caso contrário, se (c==176 && r>1 && r<144) %borda direita
        f(r,c)=ceil((f(r-1,c-1)+f(r-1,c)+f(r,c-1)+f(r+1,c-1)+
          f(r+1,c))/5);
      %=== CASO GERAL: PIXEL FORA DAS LATERAIS =====
      Senão %pixel fora das laterais
        f(r,c)=ceil((f(r-1,c-1)+f(r-1,c)+f(r-1,c+1)+f(r,c-1)+
          f(r,c+1)+f(r+1,c-1)+f(r+1,c)+f(r+1,c+1))/8);
      fim4, fim3, fim2, fim1
     $Y_{2i} = f$ ;
  
```

4.6.10.**Avaliação Subjetiva do Módulo Proposto para Geração da Informação Lateral**

Neste experimento, pretende-se avaliar a eficiência do processo de geração da informação lateral de forma geral. Assim, como primeiro experimento, buscase comparar subjetivamente (inspeção visual do autor deste trabalho) a qualidade da informação lateral criada a partir do método proposto (descrito na seção 4.6.8) em relação ao frame original Wyner-Ziv, correspondente, comparando-a também com uma informação lateral gerada com base apenas em métodos de interpolação de frames adjacentes. Assim, a figura 4.66(a) mostra uma estimativa de um frame WZ gerada utilizando-se somente uma simples interpolação dos frames adjacentes; na figura 4.66(b) a estimativa foi gerada usando apenas uma interpolação ponderada (ver eq. (5.1) abaixo) dos frames adjacentes; a figura 4.66(c) mostra a estimativa obtida pelo método proposto para geração da

informação lateral; e na figura 4.66(d) tem-se o frame WZ original. Observando-se as imagens, nota-se que há grandes distorções na imagem WZ reconstruída utilizando-se apenas uma interpolação no processo de geração da informação lateral, principalmente na região da face do *Foreman* (região de maior movimento). Embora a figura 4.66(b), onde se aplicou uma interpolação ponderada, apresente uma distorção um pouco menor, sua qualidade ainda é bem inferior à da figura 4.66(c), que foi gerada a partir do método proposto. Na figuras 5.2 referentes à sequência *News*, além de serem retomados os mesmos comentários acima, percebe-se que na figura 5.2(a) aparecem quatro braços na bailarina, o que sinaliza grandes erros de estimação, o qual já é atenuado pelo uso da interpolação ponderada proposta (figura 5.2(b)) e completamente retirado pela aplicação do método proposto completo (figura 5.2(c)). Na figuras 4.68 referente à sequência *Coastguard*, a interpolação ponderada consegue atenuar, mas não eliminar elementos indesejáveis na imagem, tais como as várias pernas aparecendo para cada jogador, o que torna o método bem ineficiente. E na figura 4.69, relativa à sequência *Soccer*, nota-se também uma grande distorção na imagens geradas a partir dos métodos mais simples em comparação com o proposto.



Figura 4.66: Frame WZ reconstruído utilizando a seguinte técnica de geração da informação lateral: (a) Interpolação simples; (b) Interpolação ponderada (proposta); (c) Método proposto (b) Frame WZ original da sequência *Foreman*.

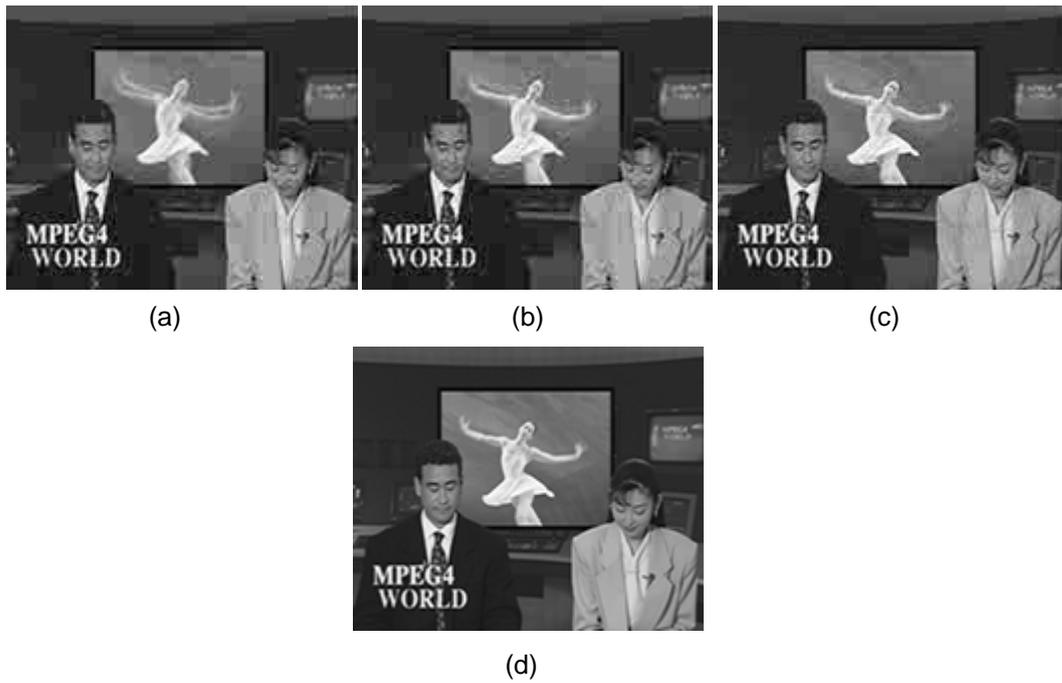


Figura 4.67: Frame WZ 138 reconstruído utilizando a seguinte técnica de geração da informação lateral: (a) interpolação simples; (b) Interpolação ponderada (proposta); (c) Método proposto (b) Frame WZ original da sequência *News*.

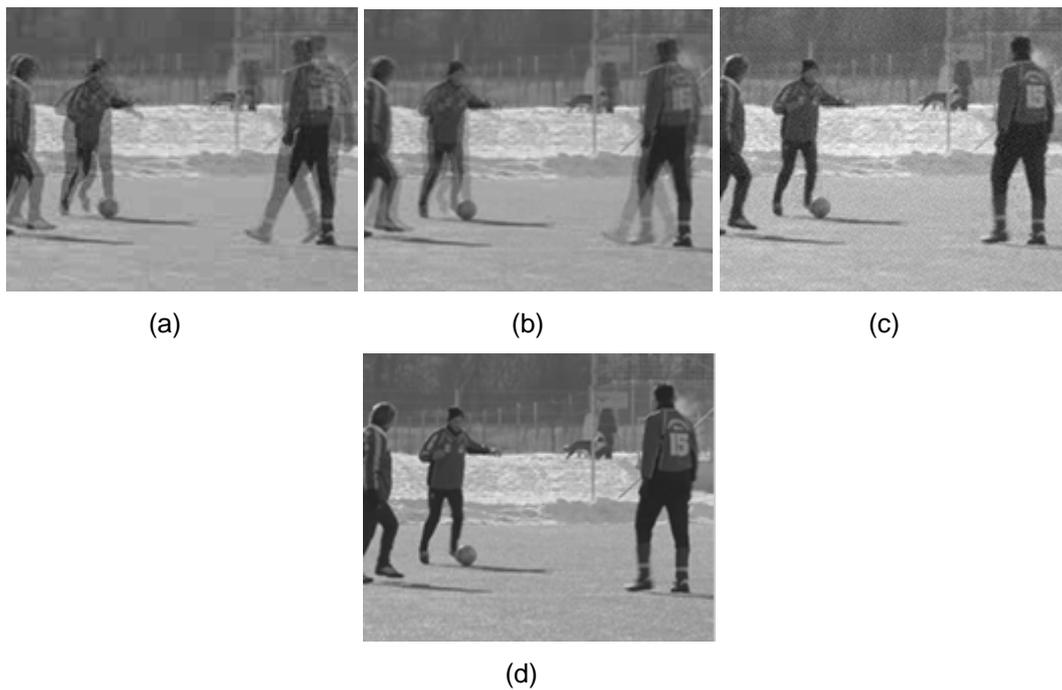


Figura 4.68: Frame WZ 138 reconstruído utilizando a seguinte técnica de geração da informação lateral: (a) interpolação simples; (b) Interpolação ponderada (proposta); (c) Método proposto (b) Frame WZ original da sequência *Soccer*.

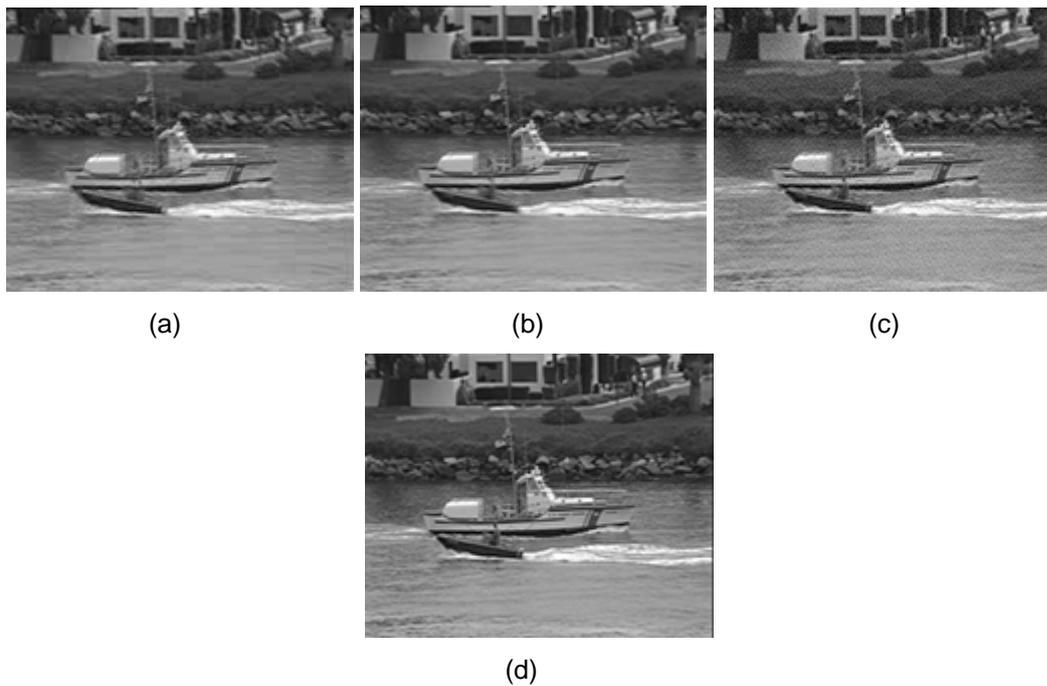


Figura 4.69: Frame WZ 138 reconstruído utilizando a seguinte técnica de geração da informação lateral: (a) interpolação simples; (b) Interpolação ponderada (proposta); (c) Método proposto (b) Frame WZ original da sequência *Coastguard*.

4.7.

Avaliação Objetiva das Ferramentas Propostas para o Módulo de Geração da Informação Lateral

Neste experimento, pretende-se avaliar a eficiência de cada ferramenta proposta utilizada no processo de geração da informação lateral. Logo, para duas taxas de compressão diferentes, ou seja, para dois passos de quantização diferentes ($QP=24$ e $QP=38$), foram codificados 100 frames das sequências *Foreman* e *News*, como é comum nas simulações em DVC, e foi medida a PSNR (*Peak Signal-to-Noise Ratio*) [50] da informação lateral resultante de cada processo em relação ao frame Wyner-Ziv original, ao longo da sequência de vídeo. Os resultados destas simulações são apresentados nas figuras 4.70 a 4.77. Uma observação importante é que a escolha da tabela de quantização (TQ) não influencia na qualidade da informação lateral, mas somente na qualidade do frame WZ reconstruído e, por isso, não houve necessidade de estabelecer um certo TQ para este experimento. Além disso, estas figuras apresentam a PSNR da informação lateral gerada com base nos seguintes métodos de obtenção da mesma:

I. Repetição do frame anterior, ou seja, $Y_{2i} = \tilde{X}_{2i-1}$.

II. Interpolação Simples, ou seja:

$$Y_{2i} = \frac{\tilde{X}_{2i-1} + \tilde{X}_{2i+1}}{2} \quad (5.1)$$

III. Interpolação Ponderada:

$$Y_{2i} = m_1 \cdot \tilde{X}_{2i-1} + (1 - m_1) \cdot \tilde{X}_{2i+1} \quad (5.2)$$

onde m_1 e m_2 são os valores da MAD entre os frames X_{2i+1} e X_{2i-1} , respectivamente, e o frame Wyner-Ziv X_{2i} original.

IV. Estimação direta e reversa (bidirecional), com blocos 8x8, fazendo uma interpolação ponderada dos frames compensados em movimento, segundo o grau de correlação de cada frame adjacente (*key*) com o frame WZ atual, ou seja:

$$Y_{2i} = m_1 \cdot F_{DIR,MB8}^{Comp} + (1 - m_1) \cdot F_{REV,MB8}^{Comp},$$

onde $F_{DIR,MB8}^{Comp}$ e $F_{REV,MB8}^{Comp}$ são os frames compensados derivados da estimação direta e reversa, respectivamente, usando blocos 8x8.

V. Estimação bidirecional, com blocos 16x16, fazendo uma interpolação ponderada dos frames compensados em movimento, conforme a fórmula: $Y_{2i} = m_1 \cdot F_{DIR,MB16}^{Comp} + (1 - m_1) \cdot F_{REV,MB16}^{Comp}$, onde $F_{DIR,MB16}^{Comp}$ e $F_{REV,MB16}^{Comp}$ são os frames compensados derivados da estimação direta e reversa, respectivamente, usando blocos 16x16.

VI. Interpolação dos frames resultantes (das duas estimções bidirecionais) baseada no coeficiente de movimento da sequência de vídeo, conforme seção 4.6.9.3.

VII. Procedimento de extrapolação de pixels, conforme demonstra o algoritmo apresentado na seção 4.6.9.4. Esta é a última técnica utilizada no processo de geração da informação lateral para a arquitetura proposta.

Observando os resultados das figuras 4.70 a 4.77, observa-se que não há variações relativas entre os resultados utilizando-se QPs diferentes ou sequências diferentes, ou seja, a ordem de eficiência dos métodos propostos permanece a

mesma. Assim, podem-se ordenar os métodos em ordem crescente de eficiência: repetição do frame anterior; interpolação simples dos frames adjacentes; interpolação ponderada dos mesmos; interpolação bidirecional 8x8; interpolação bidirecional 16x16; interpolação dos frames resultantes (da etapa anterior) baseada na quantidade de movimento; e extrapolação de pixels adicionada após esta última interpolação, como era de se esperar, já que cada etapa do processo de geração da informação lateral completa o processo anterior, ou melhor, cada etapa tem o objetivo de superar alguma falha ou imperfeição da etapa anterior.

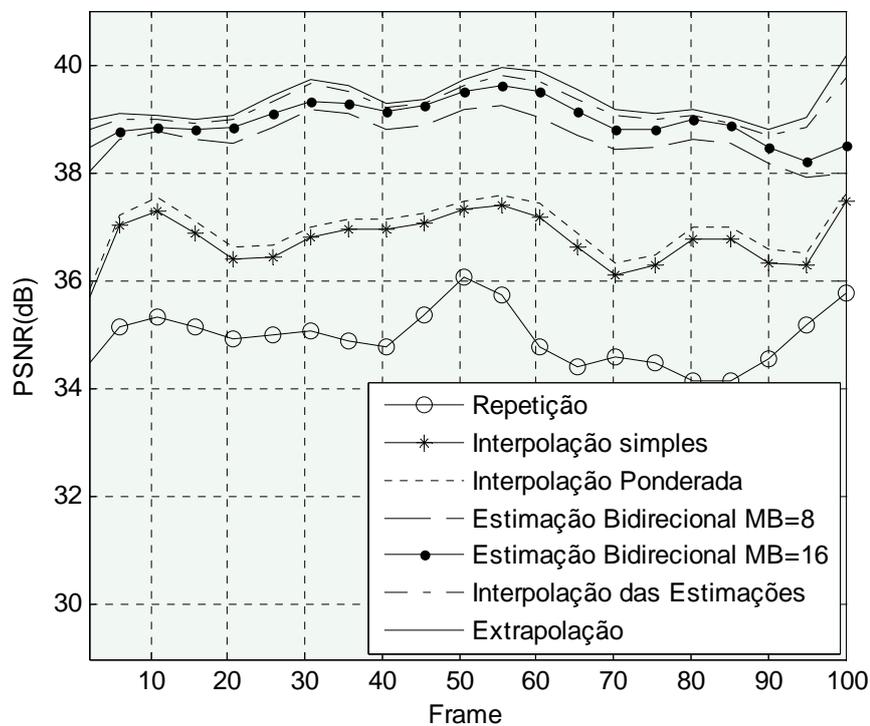


Figura 4.70: PSNR para as diversas etapas do processo de geração da informação lateral, para a sequência *Foreman*, QCIF, com passo de quantização QP = 24 (intra).

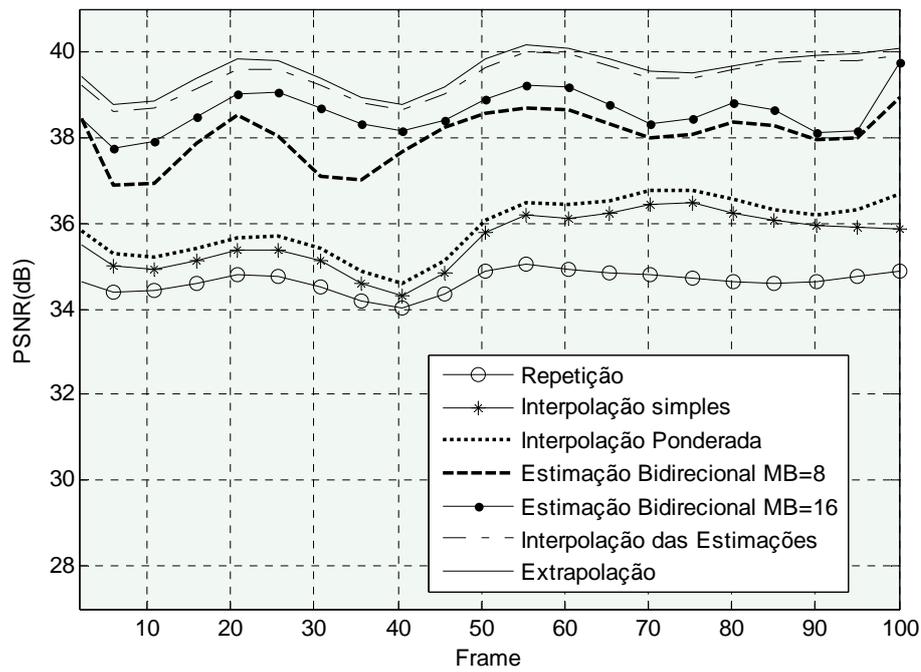


Figura 4.71: PSNR para as diversas etapas do processo de geração da informação lateral, para a sequência *News*, QCIF, com passo de quantização $QP = 24$ (intra).

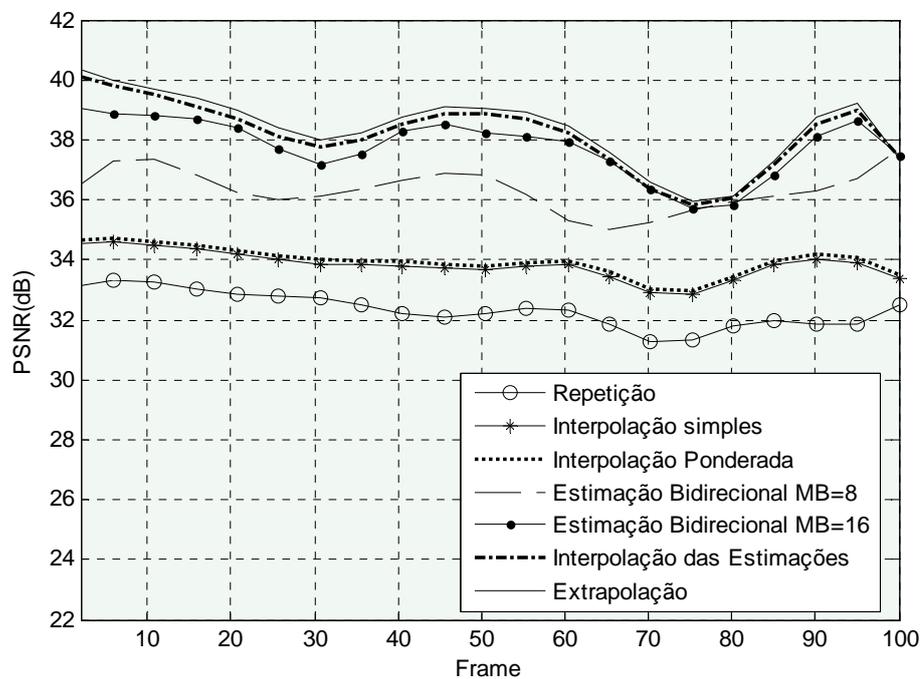


Figura 4.72: PSNR para as diversas etapas do processo de geração da informação lateral, para a sequência *Coastguard*, QCIF, com passo de quantização $QP = 24$ (intra).

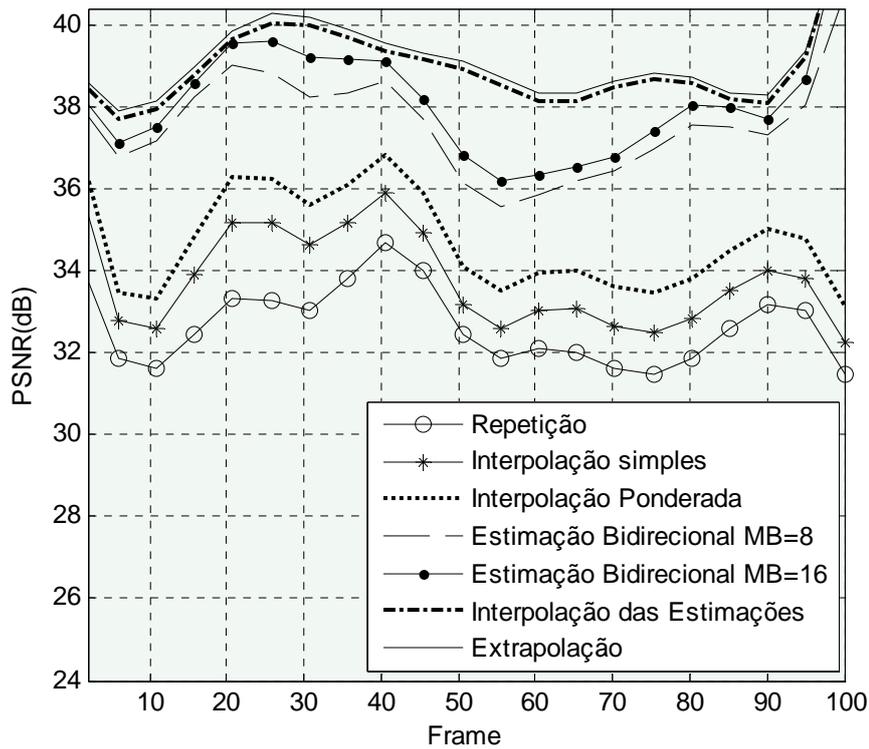


Figura 4.73: PSNR para as diversas etapas do processo de geração da informação lateral, para a sequência *Soccer*, QCIF, com passo de quantização $QP = 24$ (intra).

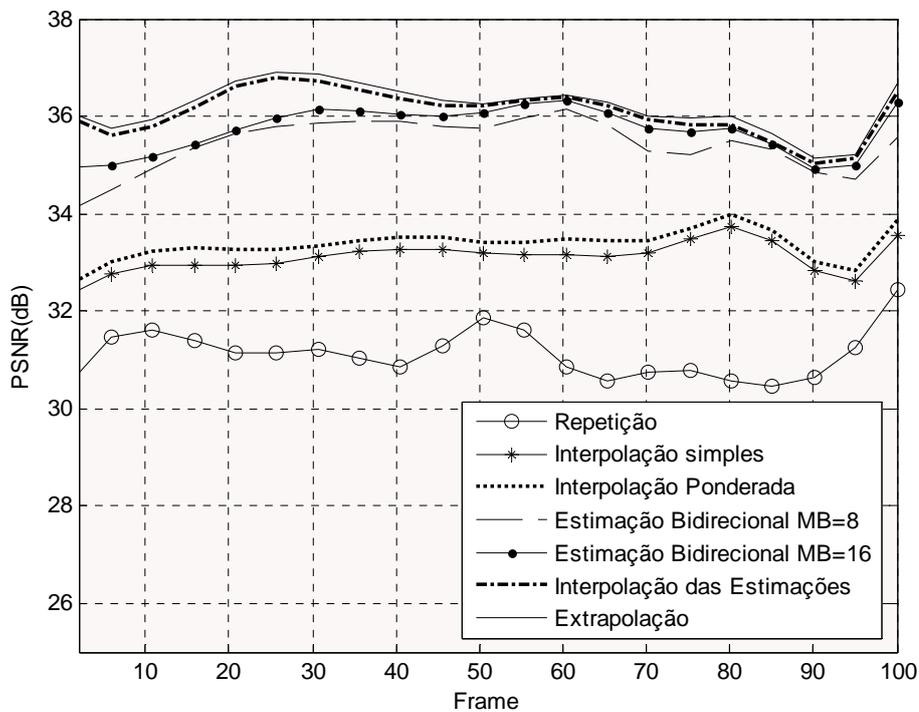


Figura 4.74: PSNR para as diversas etapas do processo de geração da informação lateral, para a sequência *Foreman*, QCIF, com passo de quantização $QP = 38$ (intra).

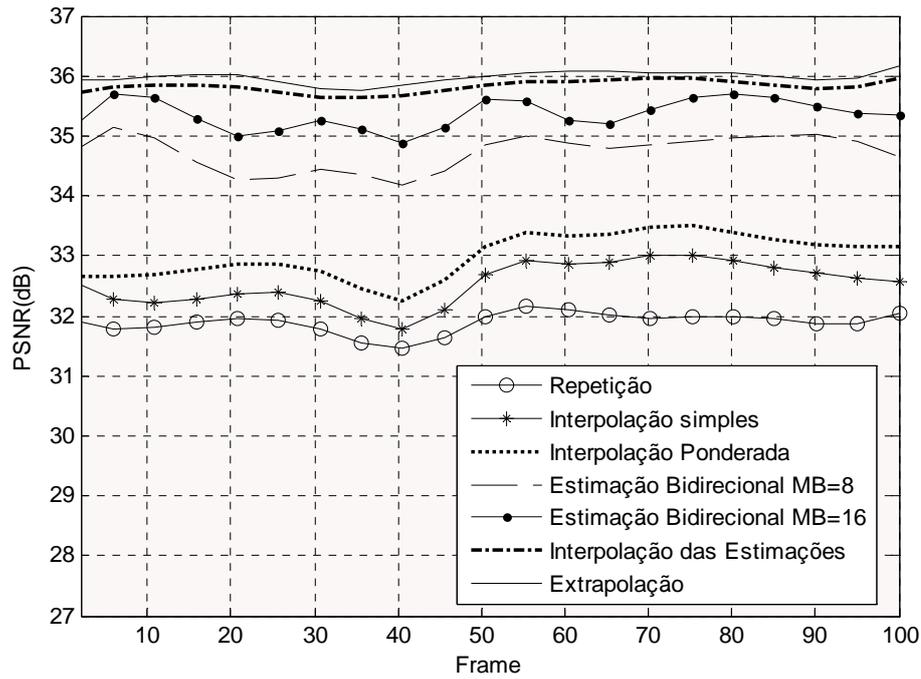


Figura 4.75: PSNR para as diversas etapas do processo de geração da informação lateral, para a sequência *News*, QCIF, com passo de quantização $QP = 38$ (intra).

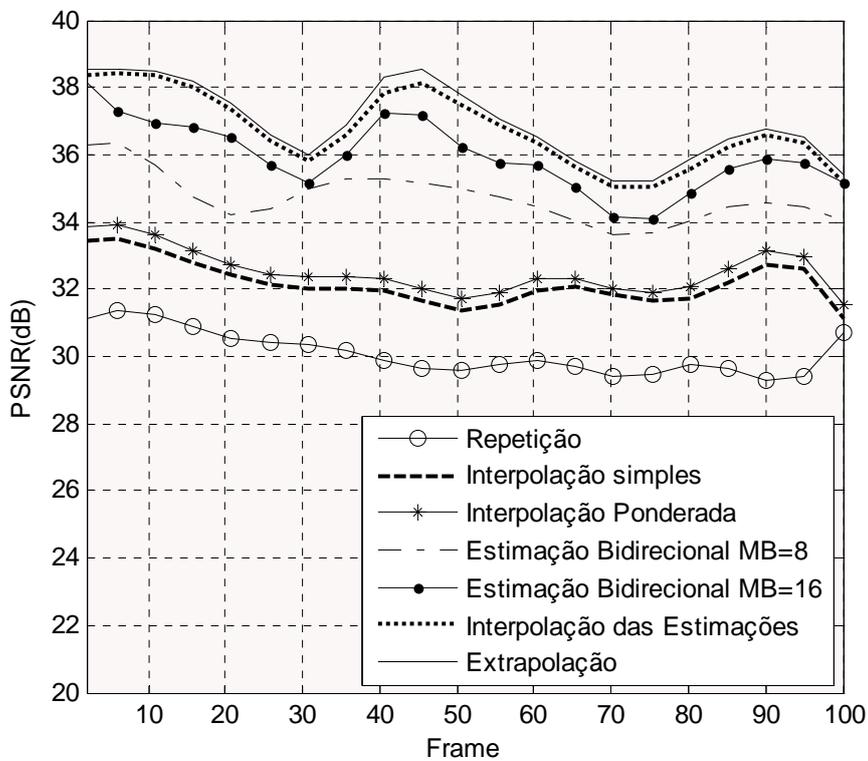


Figura 4.76: PSNR para as diversas etapas do processo de geração da informação lateral, para a sequência *Coastguard*, QCIF, com passo de quantização $QP = 38$ (intra).

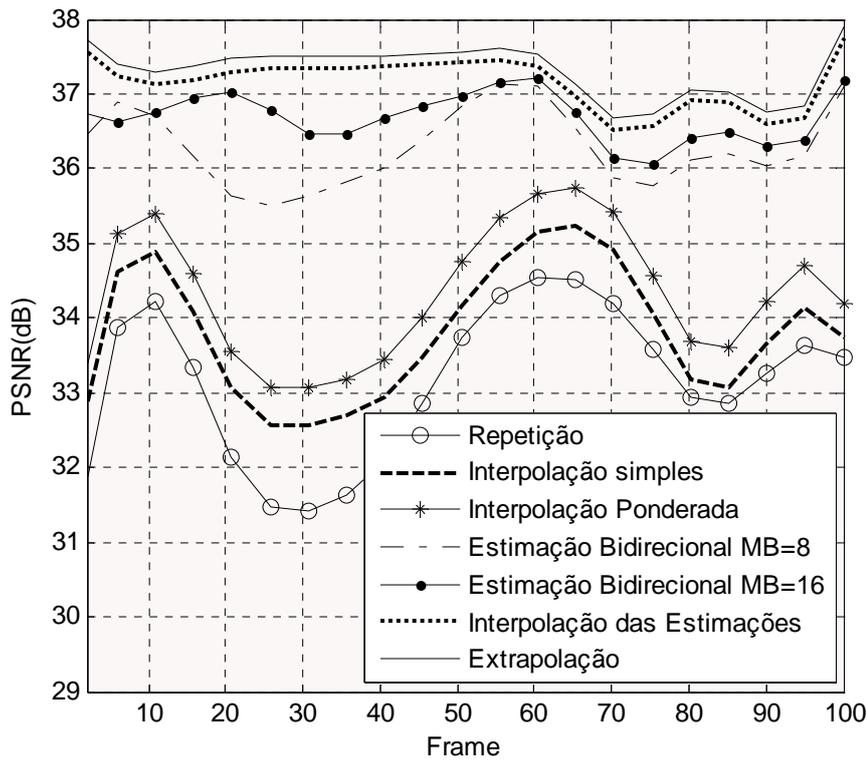


Figura 4.77: PSNR para as diversas etapas do processo de geração da informação lateral, para a sequência Soccer, QCIF, com passo de quantização QP = 38 (intra).

4.7.1. Avaliação Objetiva do módulo de Normalização

Neste experimento, avalia-se a influência ou contribuição do processo de normalização da informação lateral, cujo objetivo é aproximar a distribuição de valores da informação lateral à do frame WZ correspondente. Logo, as figuras 4.78 a 4.83 mostram as curvas da PSNR do frame WZ reconstruído ao longo da sequência de vídeo, para diversas propostas ou fatores de normalização da informação lateral. Assim, as formas de normalização são as seguintes: sem normalização; com fator de normalização 240; com fator 250; com fator 255; e com fator igual ao valor máximo do frame WZ original. Observando as figuras 4.78 a 4.83, vê-se que na maioria dos casos o fator de normalização 255 é o que resulta em uma maior PSNR, sendo adotado, portanto, como fator padrão no codec proposto. Além disso, pode-se verificar através das curvas do valor médio de WZ e da informação lateral Y, apresentadas nas figuras 4.78 a 4.83, as grandes

diferenças na luminância de cada um destes frames ao longo da sequência. Estas diferenças são causadas por falhas no processo de estimação de movimento e interpolação, as quais resultam em várias áreas com pixels de baixa intensidade, o que impulsionou o autor deste trabalho a desenvolver uma etapa de normalização para atenuar este problema.

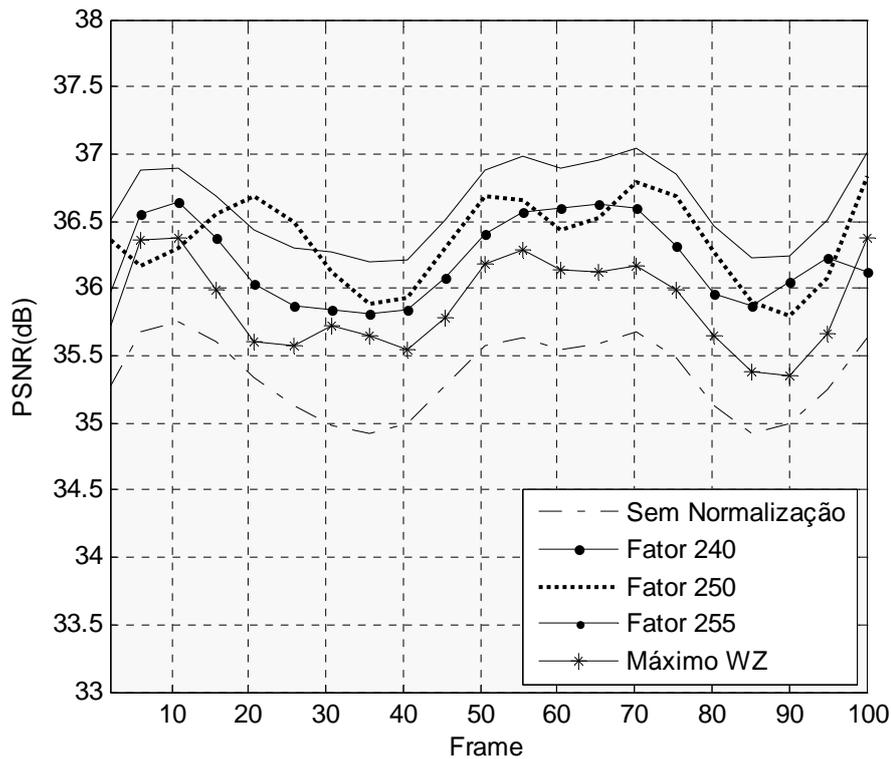


Figura 4.78: PSNR do frame Wyner-Ziv utilizando diferentes fatores de normalização, para sequência *Foreman*, com QP = 38 (intra) e TQ = 7 (WZ).

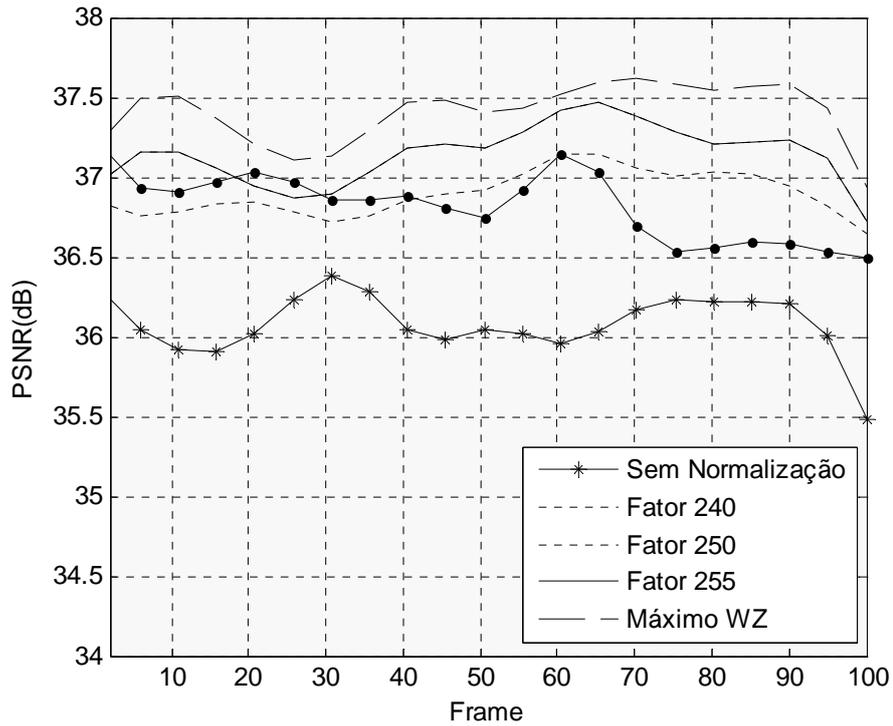


Figura 4.79: PSNR do frame Wyner-Ziv utilizando diferentes fatores de normalização, para sequência *News*, com QP = 38 (intra) e TQ = 7 (WZ).

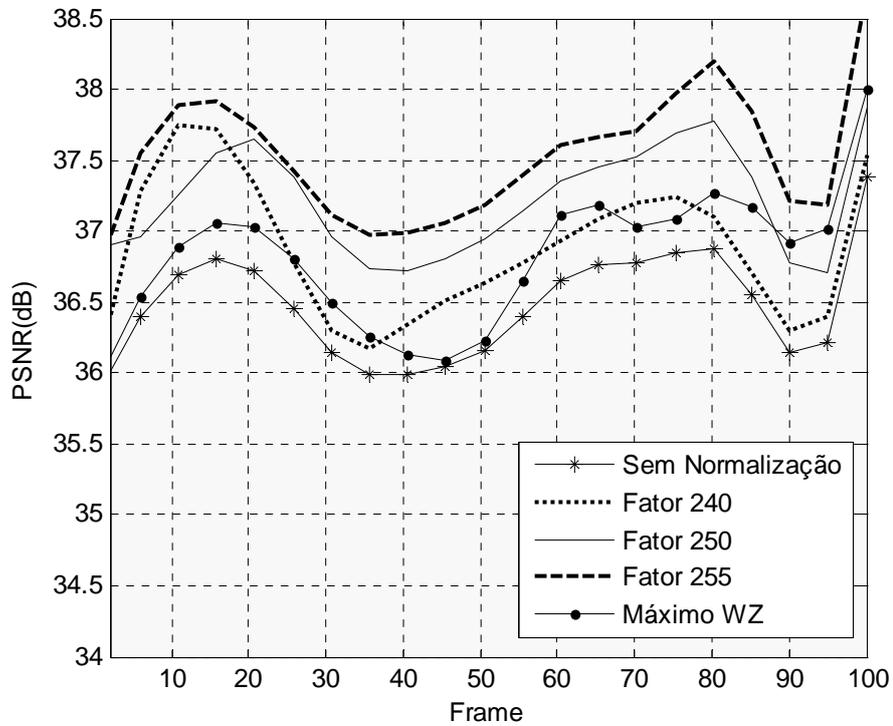


Figura 4.80: PSNR do frame Wyner-Ziv utilizando diferentes fatores de normalização, para sequência *Coastguard*, com QP = 38 (intra) e TQ = 7 (WZ).

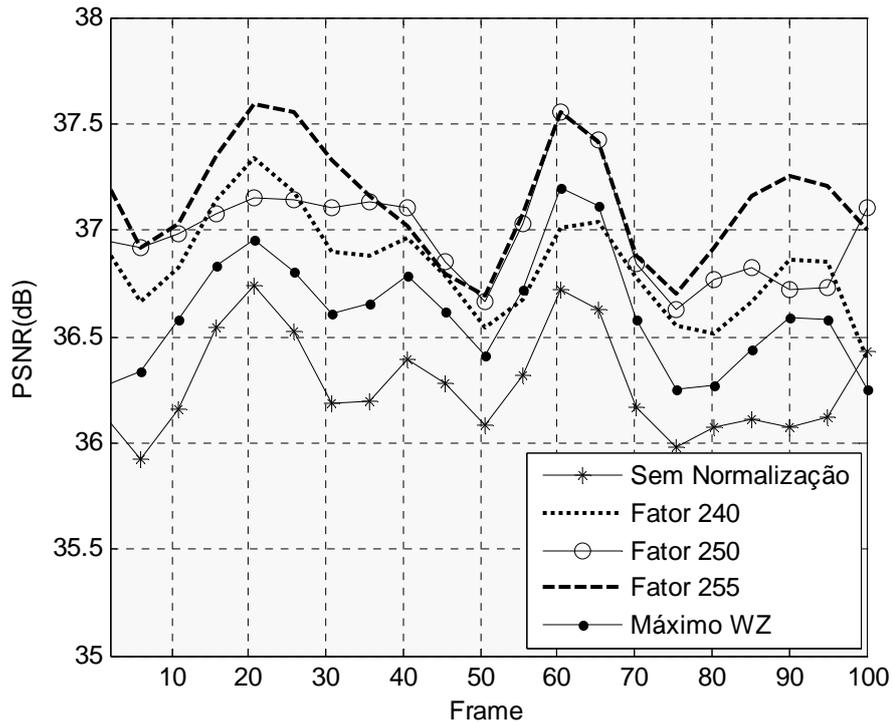


Figura 4.81: PSNR do frame Wyner-Ziv utilizando diferentes fatores de normalização, para sequência Soccer, com QP = 38 (intra) e TQ = 7 (WZ).

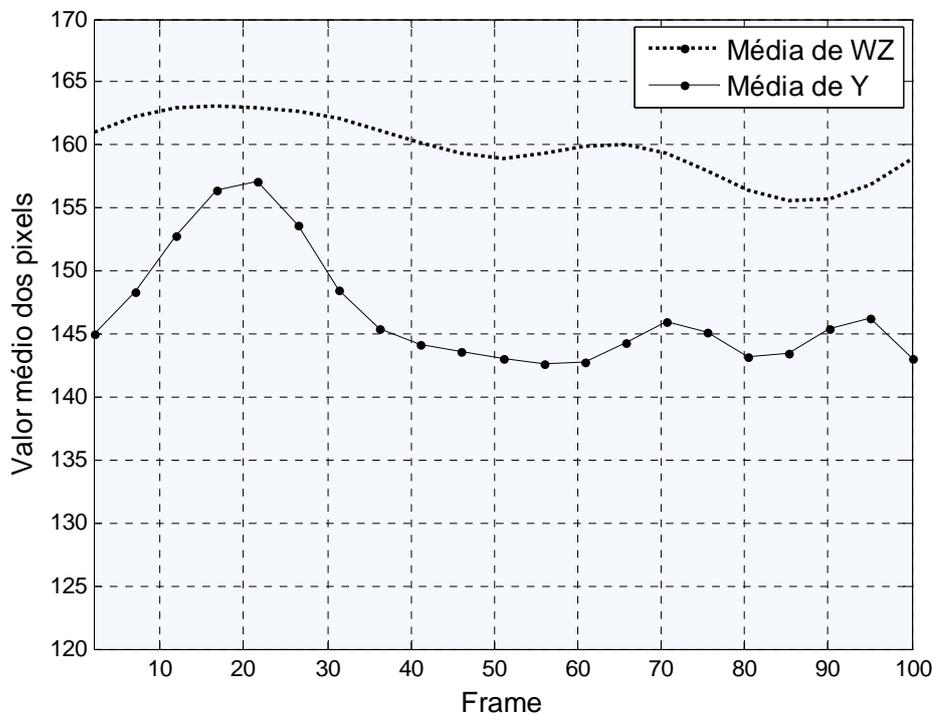


Figura 4.82: Comparação entre o valor médio dos pixels do frame WZ e da informação lateral, para sequência News.

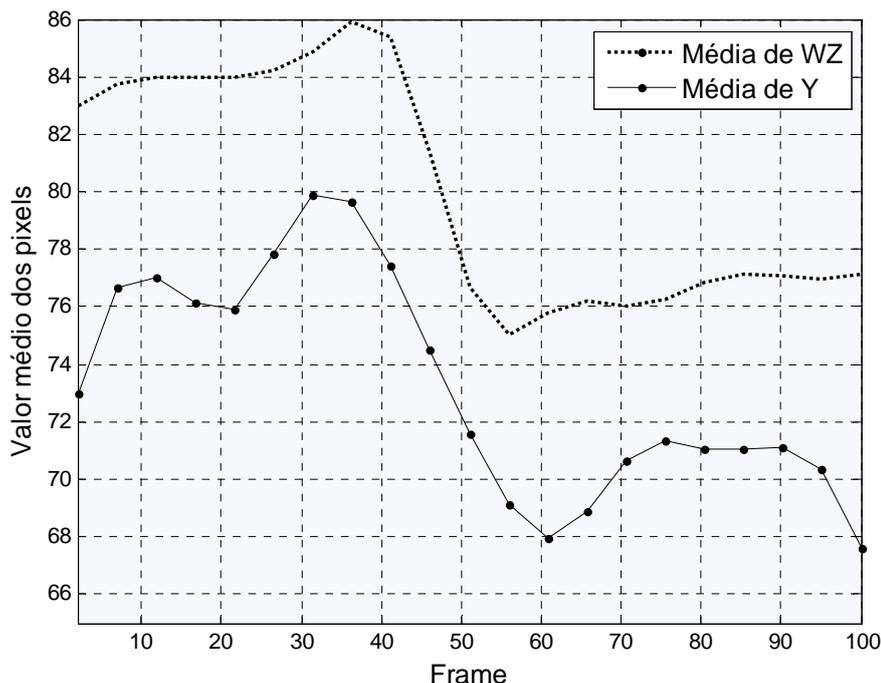


Figura 4.83: Comparação entre o valor médio dos pixels do frame WZ e da informação lateral, para sequência *Foreman*.

4.7.2.

Avaliação Subjetiva Conjunta do processo de Normalização e de geração da Informação Lateral

Neste experimento, pretende-se avaliar subjetivamente a eficiência de algumas técnicas propostas neste trabalho para geração da informação lateral, assim como do módulo de normalização, através da apresentação e avaliação subjetiva da imagem resultante em cada etapa. Desta forma, utilizando-se $QP=34$ (intra) e $TQ=8$ (WZ), na figura 4.84 são apresentados os frames resultantes dos processos de: interpolação ponderada do frame resultante da estimação bidirecional de movimento 8×8 (figura 4.84(a)), extrapolação do frame interpolado (figura 4.84(b)) e normalização da informação lateral (figura 4.84(c)). O frame original é apresentado na figura 4.84(d).

Observando a figura 4.84(a), nota-se a presença de segmentos de reta pretos, que correspondem a pixels de valor zero, os quais são chamados de pixels descobertos, ou seja, eles não foram preenchidos pela estimação direta de movimento nem pela reversa. Após o processo de extrapolação os pixels são “preenchidos” através de uma interpolação dos pixels ao seu redor (ver figura 4.84(b)), porém, como na maioria das vezes estes pixels vizinhos também

possuem valor zero, o pixel “coberto” resultante possuirá um valor mais baixo que deveria, formando segmentos de retas de menor luminância que os pixels ao seu redor. Depois da informação lateral ser gerada, aplica-se a normalização da imagem a fim de que esta fique mais próxima à original, como podemos observar claramente, se comparamos as figuras 4.84(b) e 4.84(c) com o frame original 4.84(d).

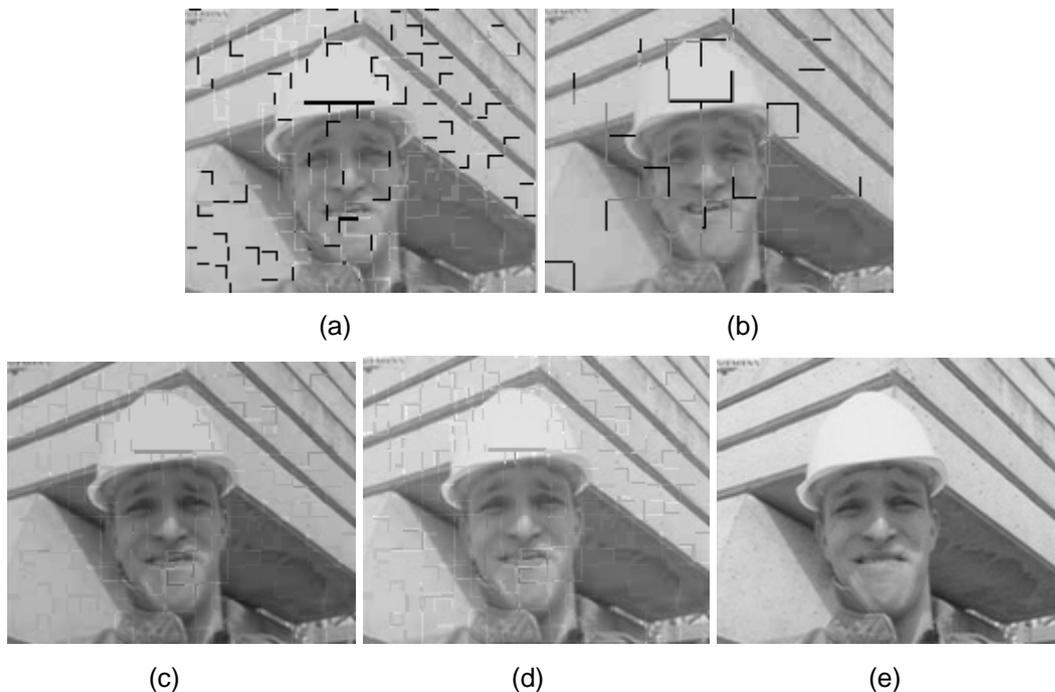


Figura 4.84: Frames resultantes do processo de: (a) interpolação ponderada da estimação bidirecional de movimento 8x8; (b) interpolação ponderada da estimação bidirecional de movimento 16x16; (c) extrapolação do frame interpolado; (d) normalização da informação lateral; (e) frame WZ original.

4.7.3. Verificador de Falha

Tendo recebido a informação lateral como uma entrada suave (valor real) e o valor do parâmetro laplaciano, o módulo de “cálculo de probabilidades” gera a distribuição laplaciana correspondente ao coeficiente daquela banda e calcula a probabilidade *a posteriori* associada a cada bit do *bitplane*. Em seguida, esses valores de probabilidades são passados para o decodificador de canal LIA. Tendo recebido os bits de síndrome do *bitplane* e as probabilidades de cada bit, o decodificador de canal (também chamado de decodificador Slepian-Wolf) corrige os erros de bit na informação lateral, e calcula o número inicial de síndromes acumuladas que foram necessárias à decodificação.

Para confirmar se a decodificação iterativa obteve sucesso, a convergência é testada segundo o critério de Hamming. Se a distância de Hamming for maior que um certo limiar, geralmente 10^{-3} [74], então o decodificador executa a próxima iteração. Depois de uma certa quantidade de iterações (na seção 3.4.1.1 mostrou-se experimentalmente que 20 a 50 iterações são suficientes), testa-se novamente o código resultante utilizando a própria síndrome recebida para verificação de erros na palavra código, através da equação:

$$\hat{\underline{C}}_{k,i}^T \cdot H_{k,i} = 0 \quad (4.51)$$

onde $\hat{\underline{C}}_{k,i}^T$ é a palavra código decodificada e $H_{k,i}$ a matriz de verificação de paridade, ambas variáveis relacionadas ao i -ésimo *bitplane* da banda b_k . Se a igualdade da eq. (4.51) for atendida, então a palavra código pertence ao código em questão, caso contrário, o *bitplane* é assumido como erroneamente decodificado e o verificador de falhas solicita mais síndromes acumuladas via canal de retorno e assim, garante-se uma baixa probabilidade de erro final. Caso a decodificação seja declarada com êxito, o *bitplane* decodificado \tilde{P}_{M_k} é armazenado em um buffer e prossegue-se com a decodificação do próximo *bitplane* \tilde{P}_{M_k-1} até que todos os outros *bitplanes* da banda b_k sejam decodificados. Após o buffer receber todos os *bitplanes* decodificados da banda b_k ele os entrega aos módulos de reconstrução.

4.7.4. Reconstrução dos Coeficientes DCT

- **Reagrupamento e decodificação binária:** o processo de reconstrução inicia-se com o rearranjo dos bits dos M_k *bitplanes* (banda b_k) em simples códigos binários e depois realiza sua decodificação, obtendo-se os índices de quantização \tilde{q}_k . Uma vez que todos os *streams* de símbolos quantizados decodificados foram obtidos, é possível reconstruir o vetor de coeficientes DCT de cada banda.
- **Cálculo do Passo de Quantização:** o decodificador conhece o valor absoluto mais alto dentro de cada banda AC (enviado pelo codificador), logo, o valor mínimo de cada banda pode ser calculado facilmente, pois $v_{k,min} = -v_{k,max}$

no caso dos coeficientes AC e $v_{1,min} = 0$ no caso DC. Além disso, o decodificador conhece também o número de níveis de quantização para cada banda, já que as tabelas contendo o número de níveis de quantização estão também disponíveis no decodificador. Assim, o tamanho do passo de quantização da k -ésima banda, W_k , pode ser calculado facilmente no decodificador a partir desses dois valores, conforme as eq. (4.11) e (4.13). Depois que o tamanho do passo de quantização de cada banda é calculado, é possível estabelecer os limites de cada intervalo de quantização para uma dada banda b_k , e assim, a reconstrução pode ser executada. O algoritmo 4.5, apresentado no final desta seção, descreve o método proposto neste trabalho para o cálculo dos limites de intervalos de quantização e dos coeficientes DCT reconstruídos.

- **Reconstrução dos Coeficientes DCT com auxílio da Informação Lateral:** para reconstruir os coeficientes DCT do frame WZ, calcula-se o valor esperado de cada j -ésimo coeficiente DCT da banda b_k , $\hat{v}_{k,j}$, dado o índice de quantização (reconstruído) associado a esta banda, $\tilde{q}_{k,j}$, e o correspondente coeficiente DCT da informação lateral $v_{y,k,j}$:

$$\hat{v}_{WZ,k,j} = E\{v_{k,j} \mid \tilde{q}_{k,j}, v_{y,k,j}\} \quad (4.52)$$

onde $\hat{v}_{WZ,k,j}$ é o valor estimado para o j -ésimo coeficiente da k -ésima banda do frame Wyner-Ziv X_{2i} .

- Aquelas bandas de frequências para as quais nenhuma informação foi transmitida pelo codificador, ou melhor, cujo número de níveis de quantização e bits são zero, são tomadas diretamente da informação lateral, $v_{y,k,j}$. O procedimento de reconstrução de cada coeficiente DCT da banda b_k pode ser representado por um dos três casos ilustrados na figura 4.27, dependendo do valor relativo do coeficiente DCT da informação lateral em relação ao índice de quantização $\tilde{q}_{k,j}$.

Caso I

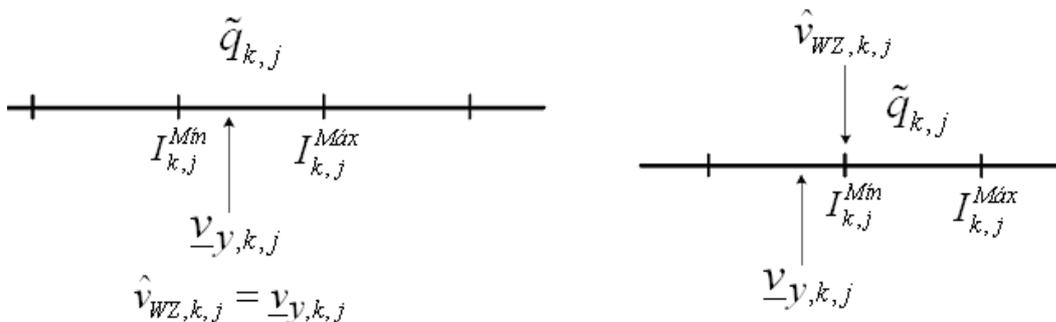
Conforme mostra a figura 4.85(a), se o valor do j -ésimo coeficiente DCT da k -ésima banda da informação lateral $\underline{v}_{y,k,j}$ estiver dentro dos limites $[I_{k,j}^{Min}, I_{k,j}^{Max}]$ que compreendem o símbolo quantizado decodificado $\tilde{q}_{k,j}$, então o coeficiente DCT reconstruído, $\hat{v}_{WZ,k,j}$, será estabelecido com o mesmo valor do coeficiente DCT da informação lateral, ou seja, é feita a atribuição: $\hat{v}_{WZ,k,j} = \underline{v}_{y,k,j}$.

Caso II

Se o valor do coeficiente DCT da informação lateral $\underline{v}_{y,k,j}$ for menor que o limite inferior do símbolo quantizado reconstruído (correspondente), $\tilde{q}_{k,j}$, então o coeficiente DCT reconstruído $\hat{v}_{WZ,k,j}$ assumirá o valor de menor intensidade relacionado ao símbolo quantizado decodificado, ou seja, o limite inferior do intervalo de quantização $I_{k,j}^{Min}$ (ver figura 4.85(b)).

Caso III

Se $\underline{v}_{y,k,j}$ for maior que o limite superior do símbolo quantizado reconstruído correspondente, $\tilde{q}_{k,j}$, então $\hat{v}_{WZ,k,j}$ irá assumir o valor de maior intensidade relacionado do símbolo quantizado decodificado, ou seja, o limite superior do intervalo de quantização $I_{k,j}^{Max}$ (ver figura 4.85(c)).



(a) Neste caso, adota-se $\hat{v}_{WZ,k,j} = \underline{v}_{y,k,j}$ (b) Neste caso, adota-se $\hat{v}_{WZ,k,j} = I_{k,j}^{Min}$

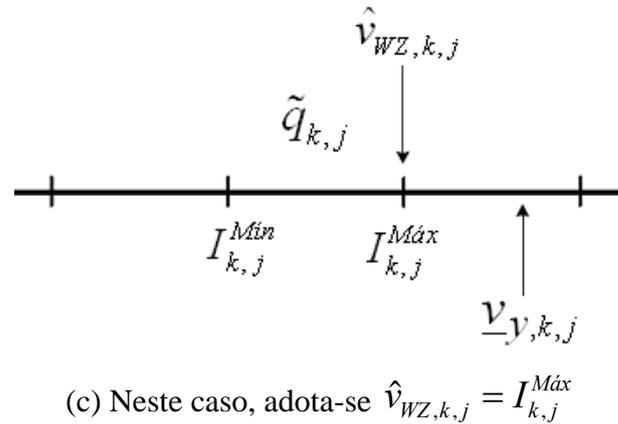


Figura 4.85: Procedimento de reconstrução de cada coeficiente DCT da banda b_k : (a) Caso I, (b) Caso II, (c) Caso III.

A seguir, é apresentado o algoritmo 4.7, o qual mostra o procedimento de reconstrução dos coeficientes DCT com auxílio da informação lateral.

Algoritmo 4.7: reconstrução dos coeficientes DCT

```

Para1 k=1:N2-1
  Para2 j=1:L
    Se3 k=1
      q_min(k,j)=(\tilde{q}(k,j))*W(k)+Vmin(k);
      q_max(k,j)=(\tilde{q}(k,j)+1)*W(k)+Vmin(k);
    caso contrário:
      q_min(k,j)=(2^{Mk-1}+\tilde{q}(k,j))*W(k)-Vmin(k);
      q_max(k,j)=(2^{Mk-1}+\tilde{q}(k,j)+1)*W(k)-Vmin(k);
    fim3
    Se4 (V_y(k,j)≤q_max(k,j) & V_y(k,j)≥q_min(k,j))
      \hat{v}_{Min}(k,j)=V_y(k,j);
    caso contrário se V_y(k,j)>Vq_max(k,j)
      \hat{v}_{Min}(k,j)=q_max(k,j);
    senão: %% caso V_y(k,j)<q_min(k,j)
      \hat{v}_{Min}(k,j)=q_min(k,j);
    fim4
  fim2
fim1

```

Visto que o valor do coeficiente DCT reconstruído está contido entre os limites do símbolo quantizado decodificado (o qual se considera possuir baixa probabilidade de erro), o erro entre os coeficientes DCT de X_{2i}^{DCT} e \tilde{X}_{2i}^{DCT} (também conhecido como distorção na reconstrução) é praticamente limitado à

precisão do quantizador, que é dada pelo tamanho do passo do quantizador W_k (ver figura 4.8), pois quanto maior o valor de W_k , maior será o erro na etapa de reconstrução e maior será a distorção no frame reconstruído.

- **Preenchimento de bandas, reagrupamento dos coeficientes e transformada inversa:** as bandas cujo número de níveis de quantização é zero são preenchidas com os coeficientes DCT correspondentes da informação lateral. Como última etapa, os coeficientes são reagrupados nos devidos blocos 4x4 no frame e em seguida uma transformada IDCT 4x4 (inversa) é aplicada, obtendo-se o frame reconstruído \tilde{X}_{2i} , o qual é armazenado no domínio do pixel.

4.7.5. Simulações Gerais com o Codec WZ-LIA

4.7.5.1. PSNR média em função do Tempo de Processamento

A figura 4.86 apresenta a PSNR ao longo dos 100 primeiros frames da sequência *Foreman*, formato QCIF, GOP de 2, para três níveis diferentes de compressão. Conforme se observa nesta figura, a eficiência da codificação WZ é um pouco superior à da *intraframe*. Na prática, de acordo com as figuras 2.6 e 2.7 [68], o desempenho relativo do codificador WZ é um pouco melhor do que o apresentado neste experimento, sendo ele mais próximo da codificação *interframe* que da *intraframe*. No entanto, esta diferença ainda não é maior, como em [4], por causa do tipo de programa utilizado nas simulações: Matlab 7; o qual torna relativamente muito mais lento o processo de decodificação, pois o programa utilizado (Matlab) possui baixa velocidade de processamento de rotinas contendo grandes laços, ou melhor, o processo iterativo de decodificação LDPC demanda um tempo de processamento bem maior e desproporcional ao aumento de tempo de processamento dos *intraframes*.

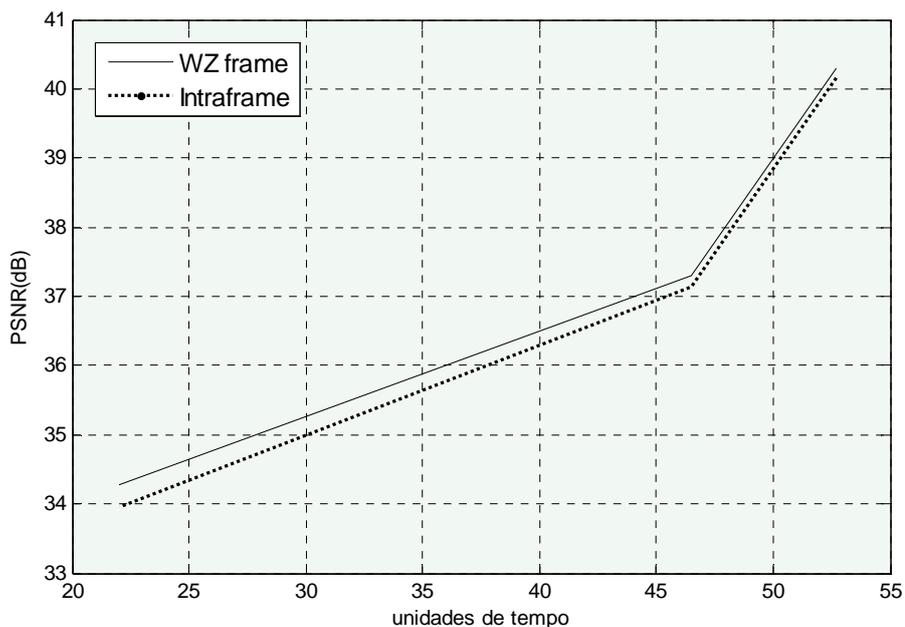


Figura 4.86: PSNR média para a sequência *Foreman*, para 3 níveis diferentes de qualidade (QP, TQ) versus tempo de processamento.

4.7.5.2.

PSNR ao longo da sequência para Diferentes Taxas de Compressão

A figura 4.87 apresenta o valor da PSNR ao longo dos frames WZ e em função do passo de quantização e do nível de qualidade, estabelecidos por QP e TQ, utilizados para codificar os frames *Key* e WZ, respectivamente. Observando-se a figura 4.87, nota-se que, embora se diminua 8 unidades do nível de qualidade 18 para o nível 10 e apenas 6 unidades do nível 10 para o 4, a diferença média entre a PSNR do nível mais alto de qualidade que é o 18 para o nível 10 é menor do que o do nível 10 para o 4. Isto porque quando se estabelece uma qualidade razoável para a quantização dos frames WZ ainda é possível ter bom êxito na reconstrução dos mesmos, o que não acontece para um nível de qualidade muito baixo.

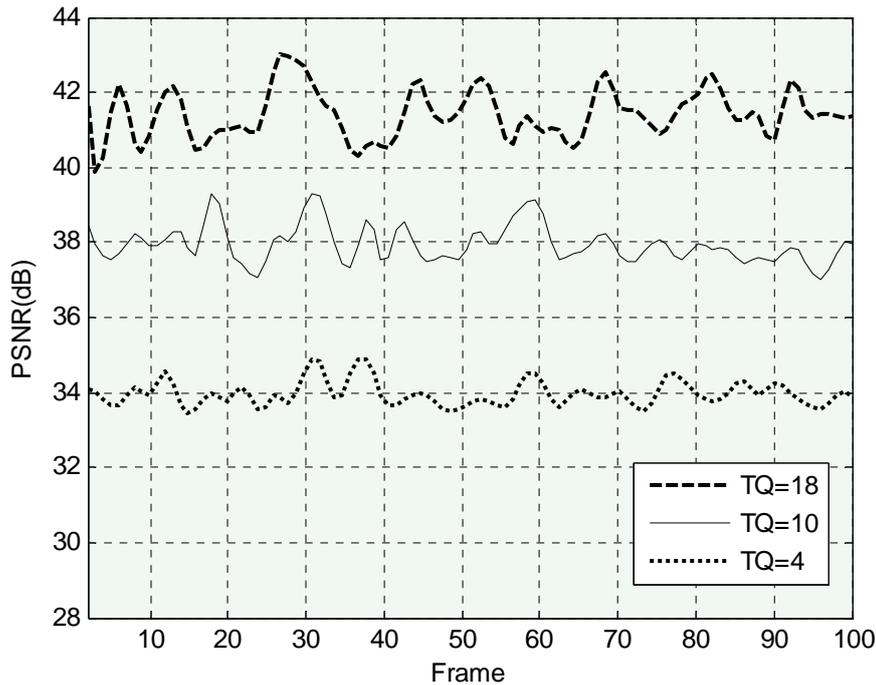


Figura 4.87: Variação da PSNR do frames WZ, ao longo dos da sequência *Foreman*, para diferentes passos de quantização *intraframe* e diferentes qualidades: {QP=24, TQ=18}, {QP=28, TQ=10}, {QP=32, TQ=4}.

4.7.5.3. Avaliação da influência da substituição das bandas zeradas do Frame WZ pela informação Lateral

Aqui, pretende-se avaliar a eficiência da substituição das bandas de coeficientes DCT do frame WZ para as quais não foi transmitido bit algum (bandas com número de níveis igual a zero na tabela de quantização, TQ, escolhida) pelas bandas correspondentes da informação lateral. As figuras 4.89 e 4.90 mostram os resultados deste experimento. A partir destas figuras, nota-se que há uma grande diferença entre a PSNR das curvas com e sem a substituição das “bandas zeradas” pelas correspondentes da informação lateral, ou melhor, esta é uma das etapas essenciais de uma correta decodificação do frame WZ. É importante ressaltar que quanto melhor a qualidade da informação lateral, ou seja, quanto mais a informação lateral for similar ao frame WZ original, maior será o peso ou a influência da substituição de seus coeficientes pelos correspondentes das bandas zeradas do frame WZ, o que resultará em uma maior diferença entre as referidas curvas das figuras 4.89 e 4.90. Outra conclusão é que quanto menor o nível de qualidade ou TQ, maior será a quantidade de bandas com número de níveis igual a zero ($L_k=0$).

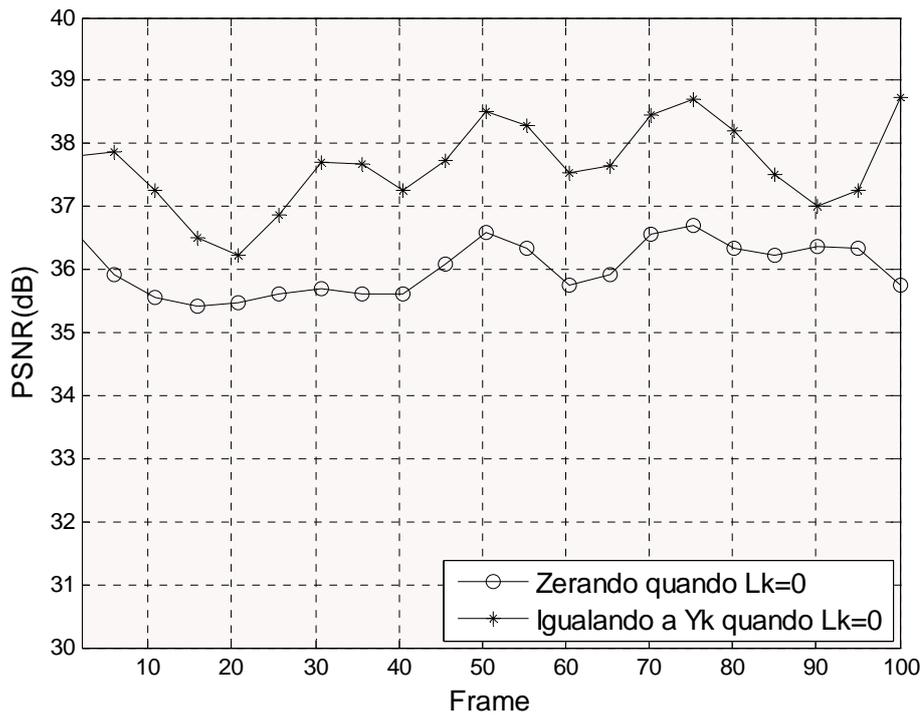


Figura 4.88: PSNR do frame Wyner-Ziv, substituindo-se as bandas com número de níveis igual a zero pelas correspondentes da informação lateral, para sequência *News*, com QP = 28 (intra) e TQ = 13 (WZ).

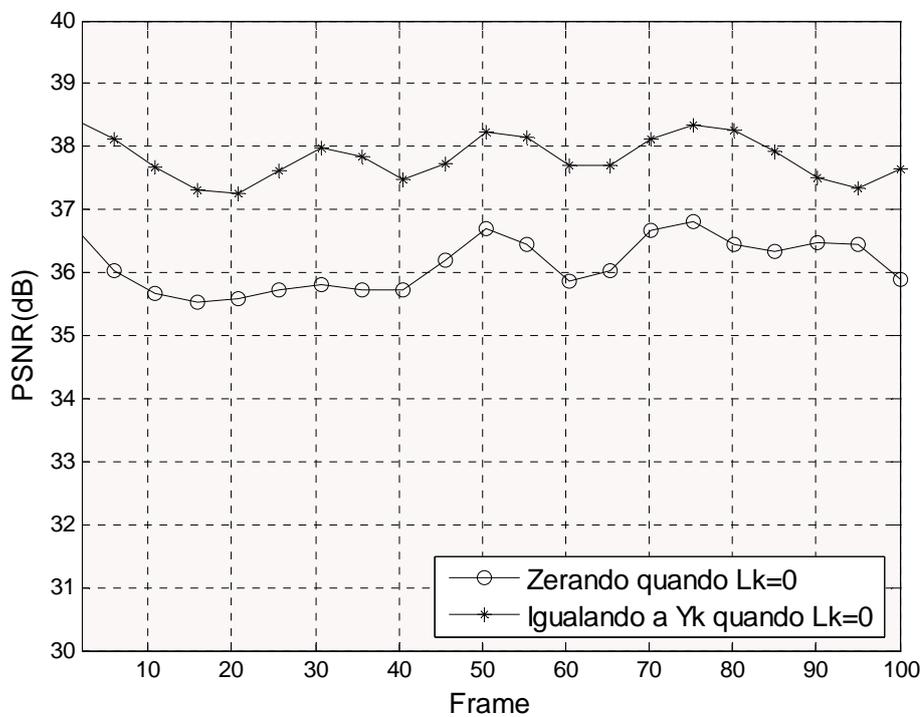


Figura 4.89: PSNR do frame Wyner-Ziv, substituindo-se as bandas com número de níveis igual a zero pelas correspondentes da informação lateral, para sequência *News*, com QP = 28 (intra) e TQ = 13 (WZ).