

Arquitetura Conceitual do *Codec* Open DVC

A arquitetura conceitual do *codec* DVC proposto e implementado nesta dissertação, através da ferramenta para simulação e avaliação de Codificação Distribuída de Vídeo Open DVC, é baseada num estudo conjunto com o trabalho de doutorado do aluno Rodrigo Silva Mello do CETUC.

Neste trabalho, foram realizadas atividades de levantamento e estudo bibliográfico do assunto DVC, estudo das arquiteturas já implementadas, análise e testes da arquitetura proposta em ferramenta de linguagem interpretada (MATLAB), implementação e arquitetura orientada a objetos da ferramenta para simulação e análise de DVC para linguagem de alto nível executável (Java).

O *codec* proposto tem sua arquitetura baseada na última versão do DVC de Stanford [28], considerada “estado da arte” em DVC. A principal alteração está no codificador de canal, o codificador original é baseado em códigos turbo, e é aqui substituído por um codificador de canal baseado em LDPC, acrônimo de *low-density parity-check code*. Essa melhoria foi sugerida pela arquitetura DVC do projeto DISCOVER [48], que apresenta os resultados e contribuições mais recentes desta tecnologia de codificação de vídeo. Além disso, a arquitetura implementada sugere um refinamento na geração da informação lateral, novos vetores de níveis de quantização na matriz de quantização utilizada e outras customizações que serão explicadas posteriormente.

O diagrama de blocos da arquitetura conceitual do *codec* Open DVC é apresentado na figura 3.1.

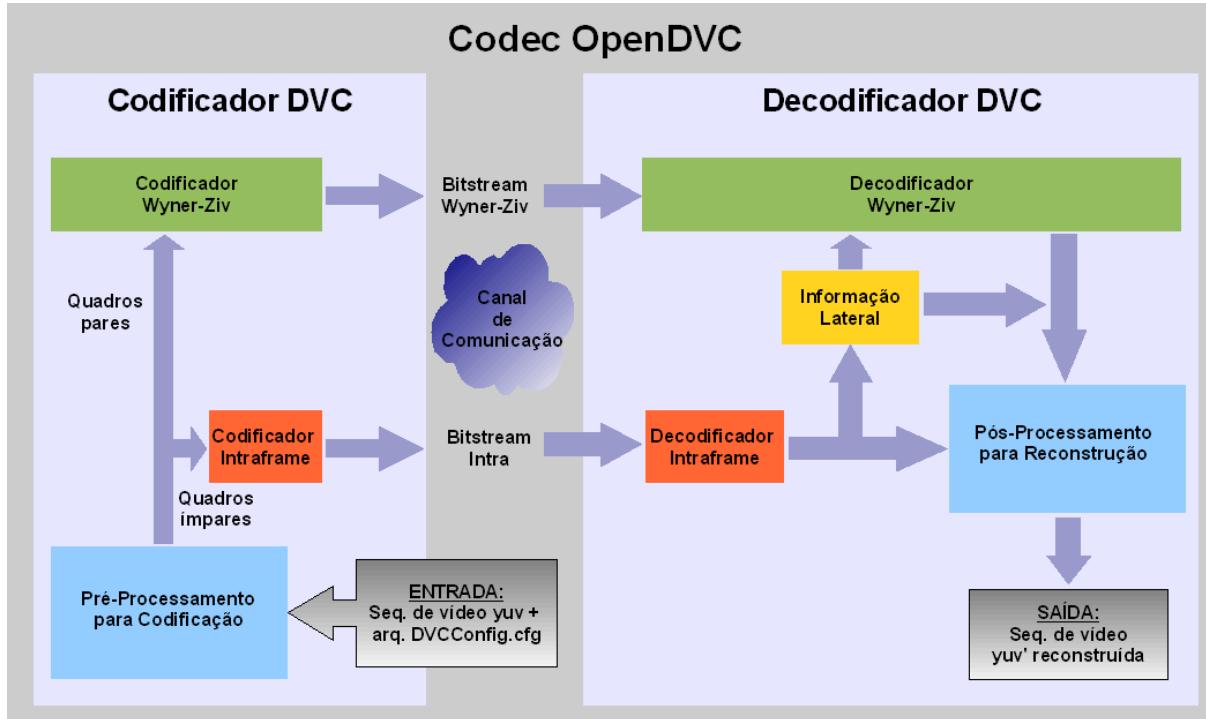


Figura 3.1 – Arquitetura Conceitual do Codec Open DVC (vista alto nível dos blocos ou módulos)

3.1

Codificação no Open DVC

O processo de codificação do Open DVC segue a arquitetura conceitual da figura 3.2, na qual apresentamos a mesma arquitetura mostrada na figura 3.1, mas com os blocos internos do codificador e do decodificador de vídeo com mais detalhes (partes internas dos blocos ou submódulos).

A seguir, explicaremos o processo de codificação passo a passo, com o auxílio das figuras 3.3 e 3.4, que são infográficos para ilustração e melhor entendimento do fluxo de informações deste processo.

Antes de iniciar as operações matemáticas e de manipulação de dados da codificação propriamente dita, a sequência de vídeo a ser codificada deve passar por um pré-processamento, necessário para adequar o arquivo de vídeo original às entradas do processo de codificação.

Este pré-processamento é composto pelas operações realizadas pelos submódulos Leitor YUV, Divisor de Quadros e Comutador de Quadros, conforme ilustrado pela figura 3.2 e que são explicados a seguir na figura 3.3.

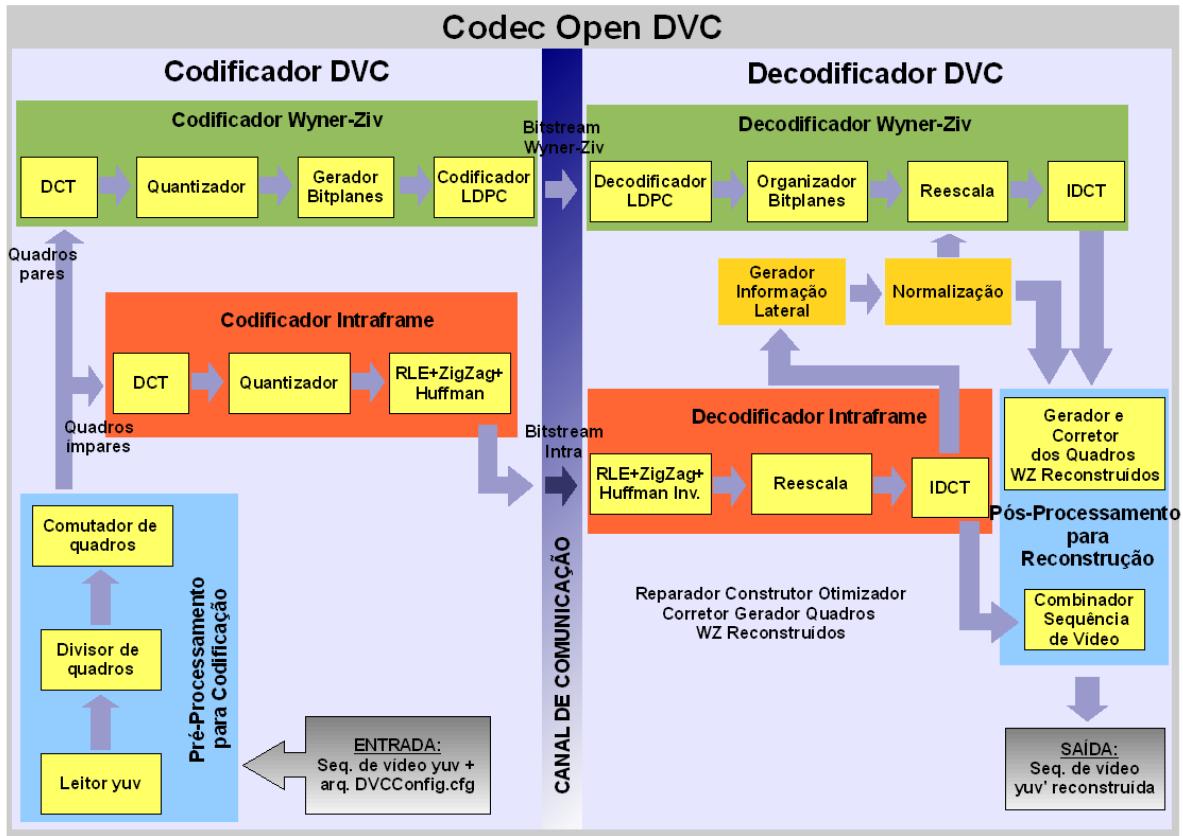


Figura 3.2 – Arquitetura Conceitual do Codec Open DVC (vista detalhada dos submódulos)

O *codec* Open DVC e consequentemente seu pré-processamento, já que é o primeiro módulo, tem como objeto de entrada uma sequência de vídeo bruto a ser codificada no formato YUV (arquivo com extensão *.yuv*), conforme podemos ver na figura 3.3, seta 1. Esta sequência possui n quadros e deve vir acompanhada de um arquivo de configuração DVCCConfig.cfg, que contém informações do vídeo que serão utilizadas no processo de codificação, como número de quadros da sequência, número de linhas e colunas de *pixels* por quadro, subamostragem e taxa de quadro, além de parâmetros que definem os arquivos de entrada (arquivo YUV original), arquivos de saída (*bitstream* Wyner-Ziv, *bitstream* Intra e arquivo YUV reconstruído) e níveis de quantização da Matriz de Quantização.

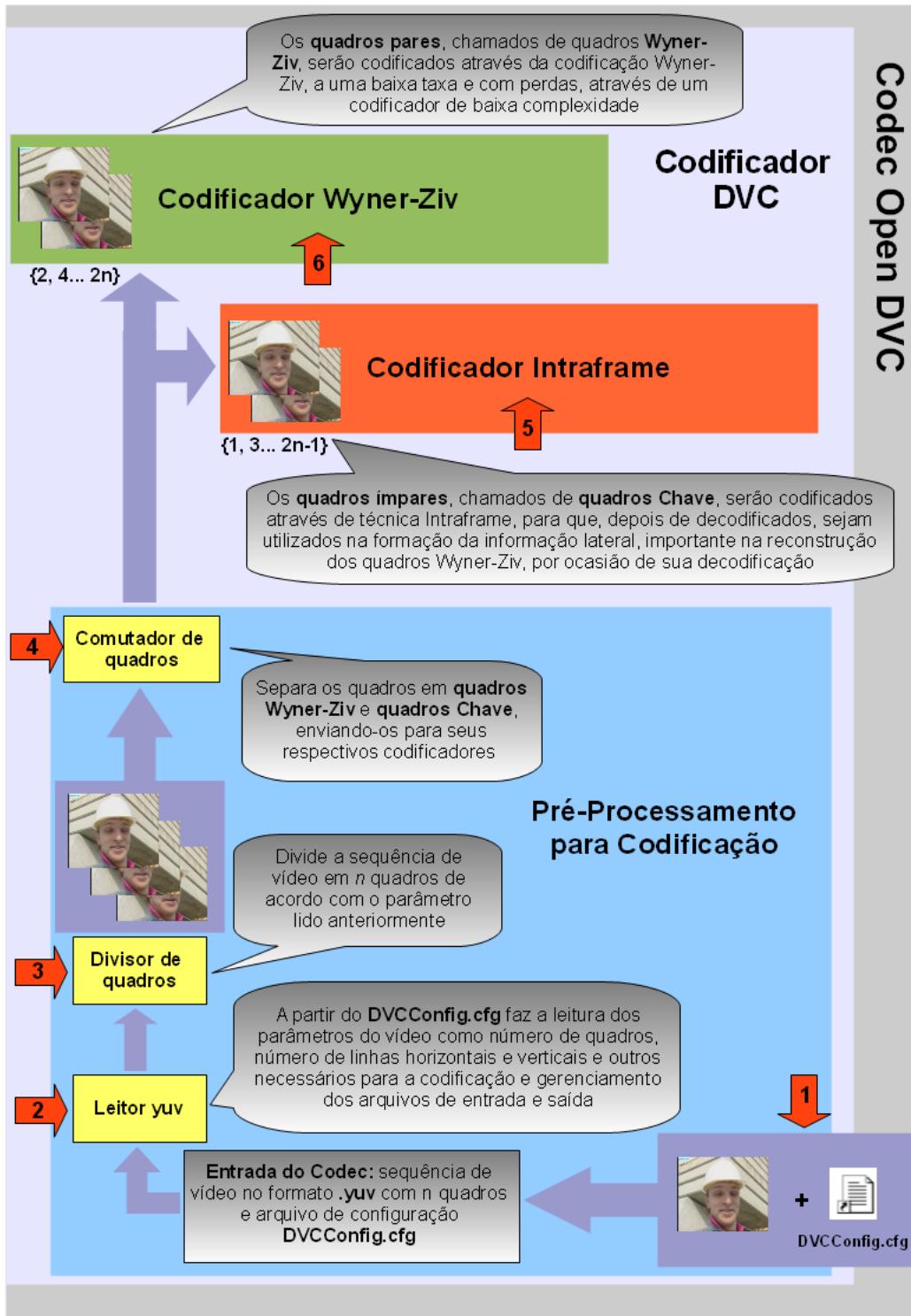


Figura 3.3 – Submódulos do bloco de Pré-Processamento no Codificador Open DVC

Após ser iniciado o processo de codificação, o submódulo Leitor yuv do bloco de Pré-Processamento, recebe e carrega na memória o arquivo **.yuv**, que contém a sequência de vídeo a ser codificada e o arquivo **DVCConfig.cfg**,

fazendo a leitura de todos os seus parâmetros e armazenando-os para serem utilizados posteriormente (figura 3.3, seta 2).

No submódulo seguinte, Divisor de quadros, a sequência de vídeo é dividida em n quadros, de acordo com o parâmetro lido do arquivo de configuração (figura 3.3, seta 3), armazenando suas informações num vetor. Então o Comutador de quadros (figura 3.3, seta 4), a partir do vetor anteriormente criado, distribui corretamente os quadros da sequência de vídeo nos seus respectivos codificadores, sendo os quadros ímpares ou quadros Chave, codificados um a um através de técnica Intraframe (figura 3.3, seta 5), para que, depois de decodificados, sejam utilizados na formação da informação lateral, importante na reconstrução dos quadros Wyner-Ziv, por ocasião de sua decodificação. Já os quadros pares, chamados de quadros Wyner-Ziv, serão, um a um, codificados através da codificação Wyner-Ziv (figura 3.3, seta 6), a uma baixa taxa e com perdas, num codificador de baixa complexidade, o que será explicado com detalhes na figura 3.4.

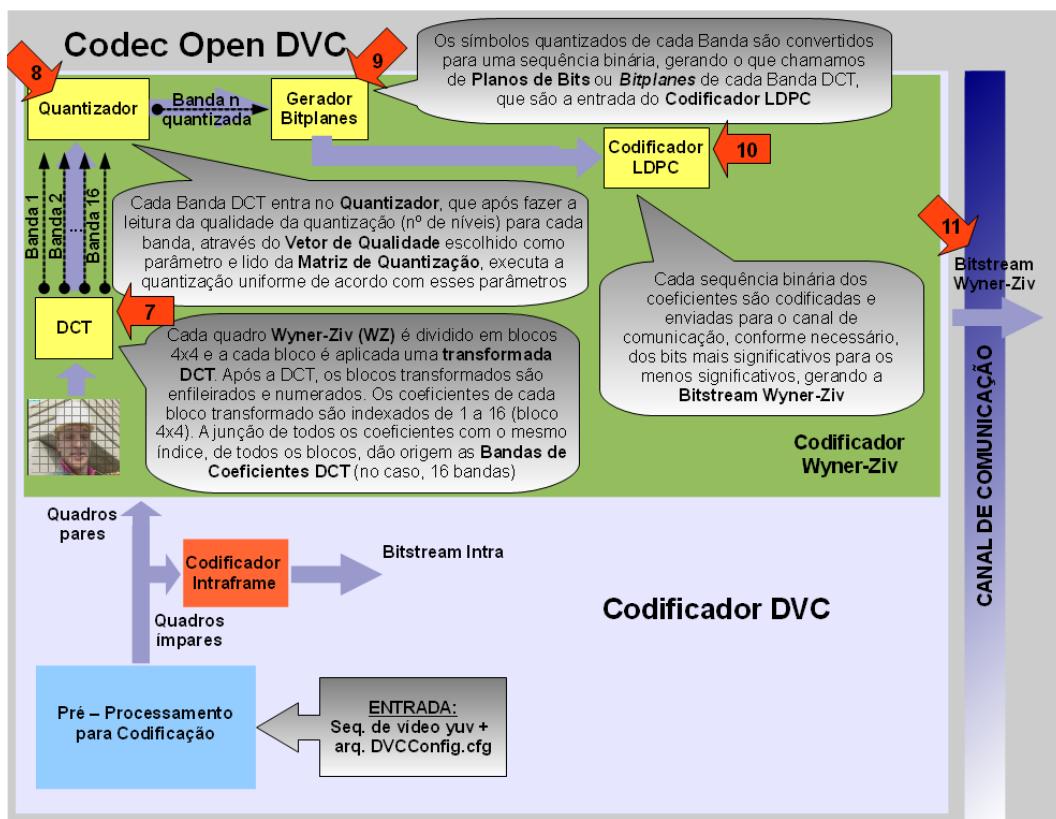


Figura 3.4 – Codificador Open DVC: fluxo do Codificador Wyner-Ziv

Uma vez divididos os quadros da sequência de vídeo e comutados para os respectivos codificadores, conforme explicado anteriormente, começa-se o processo de codificação propriamente dito, através da codificação Intraframe, para os quadros ímpares (quadros Chave) e codificação Wyner-Ziv para os quadros pares (quadros Wyner-Ziv), conforme mostrado na figura 3.3.

Na codificação Wyner-Ziv, no submódulo DCT (figura 3.4, seta 7), cada quadro Wyner-Ziv X_{2i} (WZ) é dividido em blocos 4x4 e a cada bloco é aplicada uma transformada DCT utilizando a mesma técnica da codificação de vídeo do padrão H.264/AVC [22]. Após a DCT, os blocos transformados são empilhados e numerados. Os coeficientes de cada bloco transformado são indexados de 1 a 16 (bloco 4x4). A junção de todos os coeficientes com o mesmo índice num vetor, de todos os blocos, dão origem às Bandas de Coeficientes DCT, conforme ilustrado na figura 3.4, seta 7.

Como a transformada foi aplicada a cada bloco 4x4 em cada quadro, existem $4^2 = 16$ posições possíveis, dando origem a 16 Bandas de Coeficientes DCT. Suponha que o *codec* esteja sendo aplicado a um vídeo no formato QCIF, com resolução de 176x144 *pixels* (176 colunas e 144 linhas), então temos 25.344 *pixels* no total, dividindo em blocos 4x4, teremos 1.584 blocos por quadro, assim, teremos 16 bandas de coeficientes DCT, número este fixo, com 1.584 coeficientes em cada posição, um de cada bloco.

Depois da transformada, cada banda de coeficientes DCT é uniformemente quantizada com 2^{M_k} níveis no submódulo Quantizador, conforme mostrado na figura 2.4, seta 8, onde 2^{M_k} varia de acordo com a banda b_k de coeficientes e está definido no vetor de níveis de quantização da matriz de quantização, que é um dos parâmetros lidos do arquivo de configuração DVCCConfig.cfg. Este passo tem grande importância nas reduções da complexidade computacional e da utilização da banda de transmissão pelas informações que passam pelo *codec*, pois uma das técnicas utilizadas para racionalizar esses parâmetros, é a utilização de vetores de níveis de quantização com várias bandas zeradas, naquelas que contém menos informação sobre o vídeo. Além disso, por ocasião da decodificação, essas bandas zeradas serão recompõtadas com o auxílio da informação lateral, como veremos posteriormente, contribuindo para a melhoria da qualidade da sequência de vídeo.

O resultado da quantização aplicada a cada banda DCT é convertido para uma sequência binária, dando origem aos *Bitplanes* ou Planos de Bits, operação realizada pelo submódulo Gerador Bitplanes, ilustrado na figura 3.4, seta 9.

Os Planos de Bits são a entrada do submódulo Codificador LDPC [33], que calcula a síndrome dos planos de bits codificados, conforme é mostrado na figura 3.4, seta 10.

Cada sequência binária dos coeficientes são codificadas e enviadas para o canal de comunicação, conforme necessário, dos bits mais significativos para os menos significativos, gerando a Bitstream Wyner-Ziv, que é o último passo do módulo Codificador Wyner-Ziv e da Codificação DVC implementada (figura 3.4, seta 11).

3.2

Decodificação no Open DVC

O processo de decodificação do *codec* Open DVC segue a arquitetura conceitual da figura 3.2, através dos blocos e submódulos representados no lado direito do infográfico, após a passagem dos Bitstream Wyner-Ziv e Intra pelo Canal de Comunicação.

O processo de decodificação então se inicia, com a recepção dos Bitstream, sendo o Bitstream Wyner-Ziv direcionado para o bloco Decodificador Wyner-Ziv e o Bitstream Intra direcionado para o bloco Decodificador Intraframe, conforme está ilustrado na figura 3.2.

O Decodificador sabe identificar automaticamente os *bitstreams* e direcioná-los corretamente, uma vez que elas estão definidas no arquivo de configuração DVCCConfig.cfg.

Uma vez recebido o Bitstream Wyner-Ziv, seu processo de decodificação é iniciado no bloco Decodificador Wyner-Ziv, conforme ilustrado na figura 3.5, através da passagem pelos processos definidos nos seus submódulos.

O primeiro processo é o de decodificação de canal, executado no submódulo Decodificador LDPC (figura 3.5, seta 1), no qual as síndromes recebidas pelo são convertidas numa sequência binária correspondente aos Planos de Bits dos coeficientes DCT reconstruídos.

Após isso, o segundo submódulo do Decodificador Wyner-Ziv, o Organizador Bitplanes, é responsável por recuperar os símbolos quantizados de cada banda, transformando as sequências binárias, da mais significativa para a menos significativa, recuperando os números inteiros desta sequência, que agrupados correspondem aos 1584 símbolos quantizados da banda b_k reconstruída, no nosso exemplo anterior de uma sequência com 300 quadros, o que é ilustrado na figura 3.5, seta 2.

O próximo processo a ser executado é o submódulo de Reescala, que é a operação inversa da quantização, no qual os símbolos quantizados são transformados nas Bandas b_k de Coeficientes DCT reconstruídos, o que é mostrado na figura 3.5, seta 3. Os processos de quantização e reescala incluem uma série de operações, todas elas inversíveis, como cálculo de faixa dinâmica, cálculo do número de bits de quantização, interações com a Matriz de Quantização e seus vetores, entre outros e serão explicados detalhadamente nas próximas subseções deste capítulo.

Por fim, o Decodificador Wyner-Ziv processa o último submódulo, IDCT, no qual as 16 Bandas de Coeficientes DCT reconstruídos são agrupadas e divididas em camadas, dando origem novamente aos 1584 blocos 4x4, que passam um a um pela operação de DCT inversa, conforme ilustrado na figura 3.5, seta 4, recuperando a informação original de cada bloco reconstruído. Esses dados serão utilizados como entrada dos próximos processos a serem executados no Decodificador DVC do Open DVC.

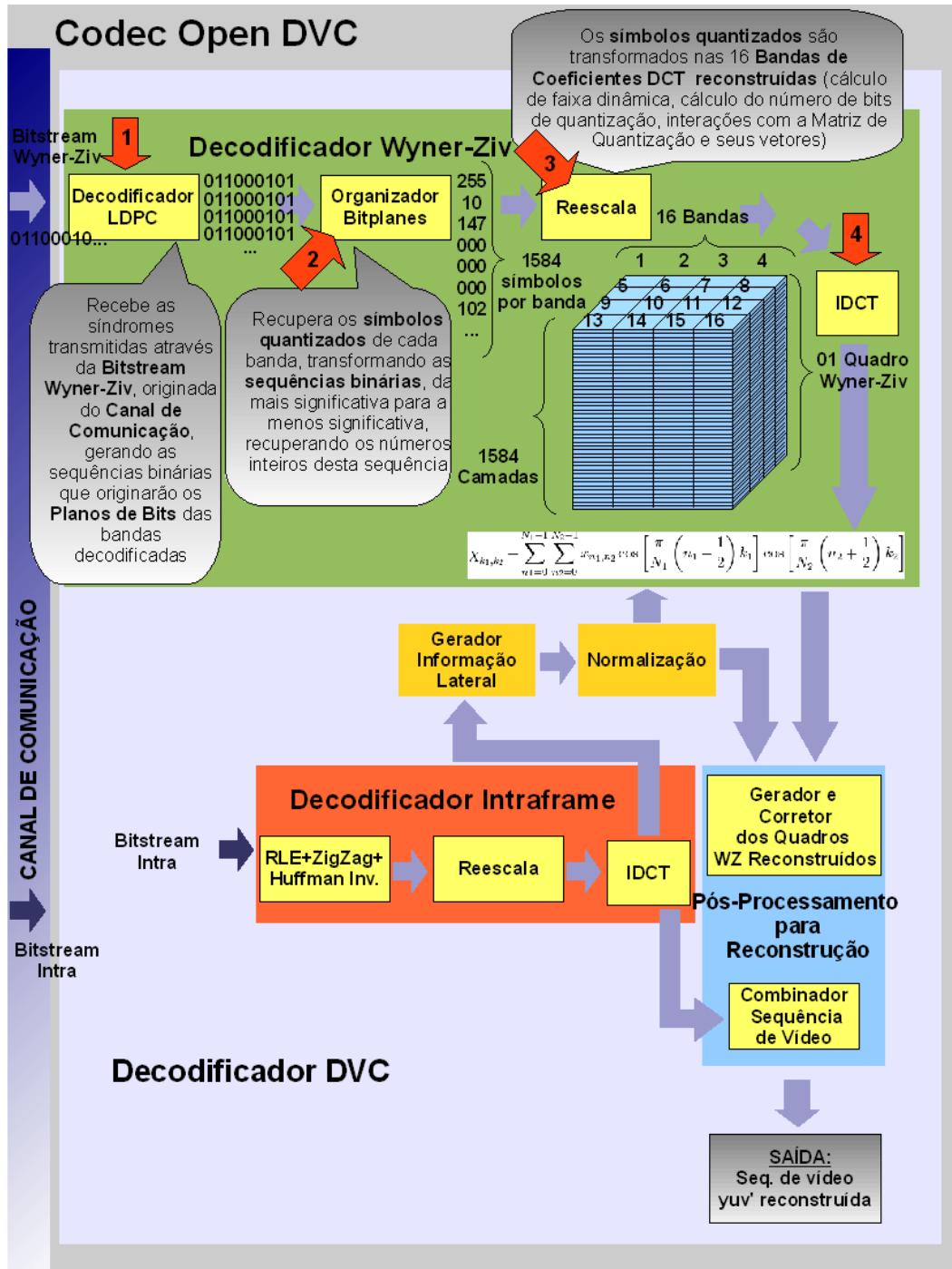


Figura 3.5 – Decodificador Open DVC: fluxo do Decodificador Wyner-Ziv

A partir da Decodificação Wyner-Ziv, temos as informações relativas aos quadros Wyner-Ziv reconstruídos. Porém, com o objetivo de reduzir a complexidade do processamento e utilizar uma baixa taxa para transmissão da informação, uma das técnicas utilizadas foi criar vetores de níveis de quantização da matriz de quantização com vários índices zerados, conforme podemos ver na figura 3.6.

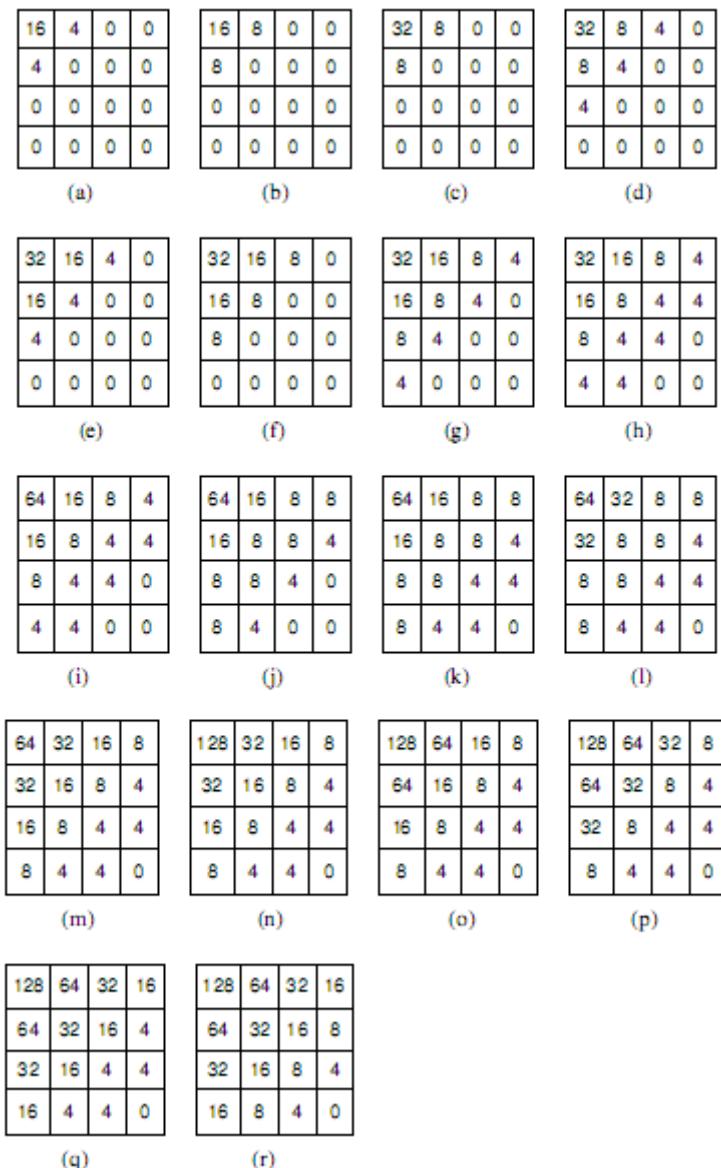


Figura 3.6 – Os 18 Vetores de Níveis de Quantização da Matriz de Quantização definida para o Open DVC

Como cada um desses índices serão aplicados em uma banda de informação, esses procedimentos tem como consequência zerar a informação da banda inteira, que na hora de ser codificada utilizará zero bits, já que cada banda é codificada com o número exato de bits necessários para sua codificação, o que pode ser calculado com os vetores de níveis de quantização, utilizando exatamente a quantidade de informação necessária para cada banda.

Os índices zerados são aplicados às bandas que correspondem às posições dos blocos do quadro que contêm a informação menos significativa para o vídeo, fazendo com que não contribua para reduzir em muito a qualidade da sequência.

Além disso, por ocasião da reconstrução do quadro, temos a oportunidade de recompletar essas bandas zeradas através da informação lateral, como veremos posteriormente, sem comprometer a taxa de transmissão, já que a informação lateral é gerada com o Bitstream Intra, que de qualquer forma será enviado, sendo, portanto, um aproveitamento do êxito para melhoria da qualidade do vídeo reconstruído.

O processo de geração e tratamento da informação lateral, conforme ilustrado na figura 3.2, tem como base, o tratamento da informação advinda do Decodificador Intraframe, que decodificou a Bitstream Intra. A partir de sua saída, os quadros Chave reconstruídos passam pelos processos definidos pelos submódulos Gerador Informação Lateral e Normalização, mostrados na figura 3.2 e que serão detalhados a seguir.

O submódulo Gerador Informação Lateral é responsável pela primeira versão da informação lateral, que é gerada através da técnica matemática de interpolação bilinear em um grid regular, bem aplicável ao caso dos quadros de uma sequência de vídeo, já que numa mesma sequência o tamanho do quadro é sempre constante.

A interpolação bilinear resolve o problema do preenchimento estimado dos pixels com qualidade aceitável, porém, existe uma limitação que pode deixar áreas não preenchidas, o que é corrigido através de uma extração em nível de pixel, melhorando os índices de qualidade da sequência. Esta operação é executada no submódulo de Normalização do Decodificador DVC.

Uma vez que o Decodificador DVC tem os quadros Chave reconstruídos, a Informação Lateral e os quadros Wyner-Ziv reconstruídos, o módulo de Pós-Processamento para Reconstrução opera sobre estes quadros, para conseguir a melhor sequência possível, através das operações de seus submódulos.

O primeiro submódulo do Pós-Processamento é o Gerador e Corretor dos Quadros WZ reconstruídos, que tem a finalidade de a partir da Informação Lateral (melhor estimativa de um quadro Wyner-Ziv a partir de seus quadros Chave adjacentes), fazer o recompletamento das Bandas de Coeficientes DCT reconstruídas zeradas, o que contribui para a construção mais fidedigna do quadro Wyner-Ziv reconstruído em relação ao seu original.

Lembremos que essa operação de zerar as Bandas de Coeficientes DCT foi originada na codificação a partir da escolha de Vetores de Níveis de Quantização da Matriz de Quantização com quantidades de zeros, o que se fez necessário para reduzir a complexidade do processamento e a banda utilizada, um dos objetivos principais da arquitetura DVC.

Uma vez completados os quadros Wyner-Ziv, a próxima operação do Pós-Processamento é realizada pelo submódulo Combinador Sequência de Vídeo, que tem a finalidade de ordenar os quadros, intercalando entre os quadros Chave, seus respectivos quadros Wyner-Ziv associados, o que gera, ao final do processo, a sequência de vídeo reconstruída, que é gravada no arquivo binário YUV definido numa das linhas do arquivo de configuração DVCCConfig.cfg (seu caminho absoluto ou relativo e nome), terminando assim toda a operação de codificação e decodificação de uma sequência de vídeo no *codec* Open DVC.

3.3

Conclusões

Este capítulo apresentou a arquitetura conceitual do *codec* Open DVC, implementada como parte dos trabalhos desta dissertação, mostrando todas as suas caixas em diagrama de blocos alto-nível, fazendo posterior detalhamento de cada funcionalidade.

Seguindo os princípios do DVC, uma sequência de vídeo YUV a ser codificada é dividida em quadros, sendo os quadros ímpares chamados de quadros chave e os quadros pares de quadros Wyner-Ziv. Os quadros chave são codificados e decodificados através de um *codec intra-frame* e no decodificador são utilizados para compor os quadros ímpares da sequência de vídeo reconstruída, mas também têm um outro papel importante, que é a geração da informação lateral, que será utilizada na formação e correção dos quadros Wyner-Ziv reconstruídos.

Os quadros Wyner-Ziv originais são codificados e decodificados por um *codec* Wyner-Ziv, que utiliza a informação lateral, originada dos quadros chave, para gerar os quadros Wyner-Ziv reconstruídos no decodificador, como explicado no parágrafo anterior.

nalizando a arquitetura apresentada, podemos concluir que ela atende às premissas da tecnologia DVC, que visa reduzir a quantidade de informação a ser trafegada pelo canal de comunicação, do codificador no transmissor para o decodificador no receptor, porém com requisitos especiais a cumprir, como manter uma baixa complexidade na codificação, devido à suposta baixa capacidade de processamento e limitação de consumo de bateria pelos dispositivos que estão deste lado do sistema de comunicações. Este requisito tem como consequência, uma reconstrução com maior complexidade no lado do decodificador (o que não é problema já que teríamos dispositivos mais robustos deste lado), com o intuito de manter um bom nível de qualidade, o que será buscado e comprovado pelos experimentos e resultados nos próximos capítulos desta dissertação.