

# 1 Introdução

Recentemente, inúmeras pesquisas têm sido conduzidas no campo da Codificação Distribuída de Vídeo (DVC – *Distributed Video Coding*), um novo paradigma para compressão de vídeo. DVC tem sua origem nos resultados provenientes da teoria da informação obtidos por Slepian & Wolf [1] para codificação distribuída de fonte (DSC – *Distributed Source Coding*) sem perdas e nos resultados obtidos por Wyner & Ziv [2] para o caso de codificação distribuída com perdas. Em resumo, seus teoremas estabelecem que, sob as mesmas condições, o desempenho taxa-distorção alcançado quando se realiza uma codificação conjunta de fontes correlatadas pode também ser obtido efetuando-se uma codificação separada (distribuída) e uma decodificação conjunta dessas fontes. Neste último caso, o da codificação distribuída, é tarefa do decodificador explorar a correlação entre as fontes de informação para realizar a decodificação, significando que o balanço da complexidade entre o codificador e o decodificador pode ser potencialmente revertido no que diz respeito aos métodos de codificação tradicionais, ou seja, o decodificador passa a ter maior complexidade em virtude das diversas ferramentas adicionais (e mais complexas) presentes no mesmo.

No caso da codificação distribuída de vídeo, uma carga bastante significativa, que é a estimação de movimento, pode ser (inteira ou parcialmente) deslocada do codificador para o decodificador. Como resultado, a codificação distribuída de vídeo vem se tornando um campo muito atrativo para uma ampla faixa de aplicações práticas, onde a carga computacional, a memória e a bateria são escassas no codificador, tais como: câmeras de vídeo *wireless*, vídeo-vigilância *wireless* de baixa potência, escaneador de documentos móvel, vídeo-conferência através de dispositivos móveis, redes de sensoriamento visual, câmeras de vídeo de baixa capacidade, transmissão de mensagens de vídeo via telefone celular (uso do MMS - *Multimedia Messaging Service*), sistemas de multiaquisição de imagens e *streaming* de vídeo distribuído. Assim, há uma série de aplicações, com uma pequena sobreposição entre algumas delas,

particularmente no que se refere a sistemas em tempo real ou não, sistemas unidirecionais ou bidirecionais, monoexibição ou multiexibição e em relação a restrições de complexidade e de bateria.

As primeiras implementações práticas em DVC foram realizadas por Puri & Ramchandran [3] e por Aaron *et al.* [4]. Puri & Ramchandran [3], introduz-se o codec PRISM (Power-efficient, Robust, high compression, Syndrome-based Multimedia coding), o qual é baseado na codificação da síndrome dos blocos de pixels codificados (blocos de entrada do codificador). Aaron *et al.* [4] propôs o uso de um codificador baseado em códigos turbo operando sobre um frame completo.

### 1.1. Fundamentos da Codificação Distribuída

Codificação distribuída refere-se à codificação de duas (ou mais) sequências aleatórias estatisticamente dependentes, mas com uma diferença especial que é a utilização de um codificador distinto para cada uma, conforme ilustrado pela figura 1.1. Cada codificador envia um *bitstream* separado para um único decodificador, o qual pode operar conjuntamente sobre todos os *bitstreams* entrantes e assim, explorar suas dependências estatísticas a fim de aumentar a eficiência do processo de decodificação.

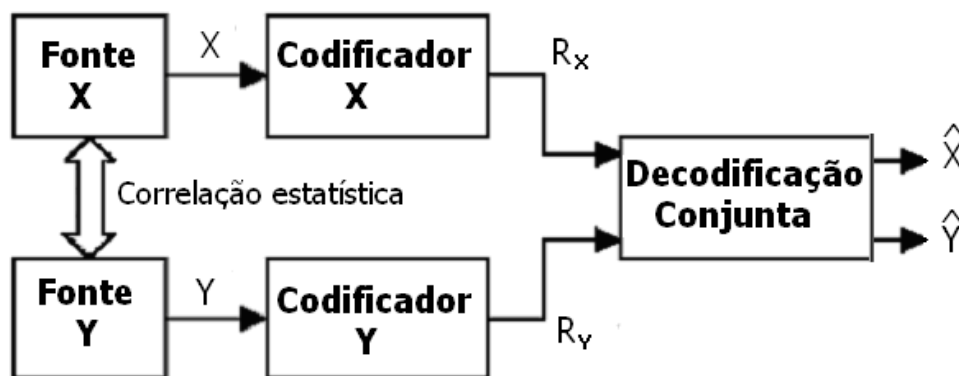


Figura 1.1: Codificação distribuída de duas sequências aleatórias estatisticamente dependentes,  $X$  e  $Y$ .

Para melhor compreensão deste conceito, considere duas sequências aleatórias estatisticamente dependentes e igualmente distribuídas  $X$  e  $Y$ . Com a utilização de codificadores e decodificadores entrópicos convencionais separados,

para uma compressão sem perdas, chega-se a uma taxa mínima  $R_X \geq H(X)$  e  $R_Y \geq H(Y)$ , onde  $H(X)$  e  $H(Y)$  são as entropias de  $X$  e de  $Y$ , respectivamente. Ao contrário do que se imaginava, pode-se conseguir um resultado bastante satisfatório com uma decodificação conjunta, mas codificação separada, dada uma probabilidade de erro muito baixa na recuperação de  $X$  e  $Y$ , a qual pode ser considerada arbitrariamente baixa (mas, em geral, não zero) para codificação de longas sequências. A figura 1.2 ilustra a região de taxa de transmissão admissível (mínima taxa alcançável ou maior compressão possível) para codificação distribuída de duas fontes estatisticamente dependentes e igualmente distribuídas.

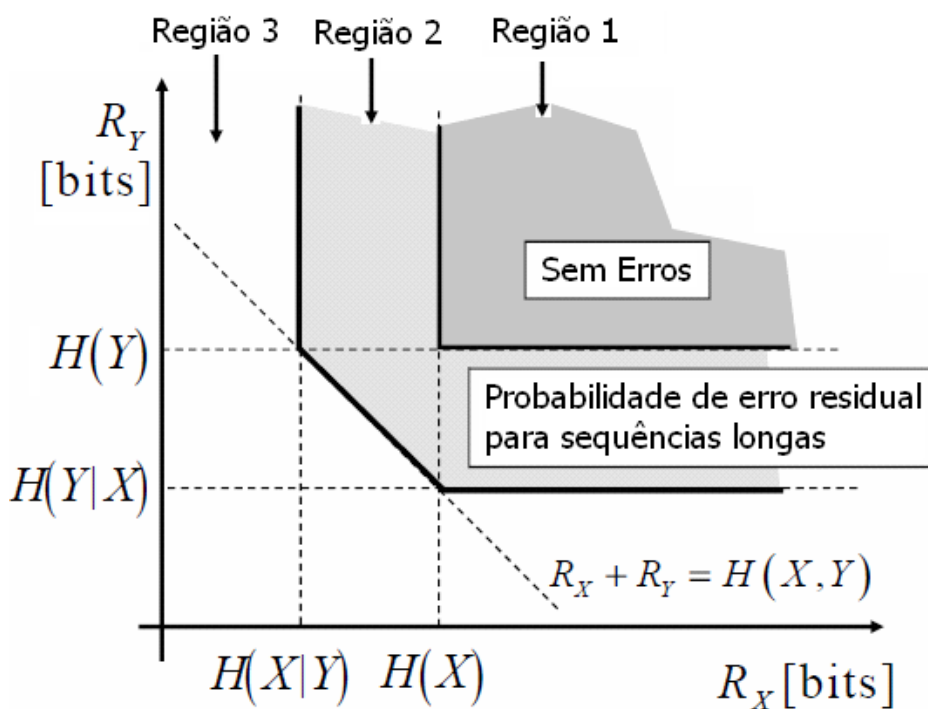


Figura 1.2: Teorema de Slepian-Wolf: Região de taxa admissível para codificação distribuída de duas fontes estatisticamente dependentes e igualmente distribuídas.

Neste caso, conforme observado na figura 1.2, o teorema de Slepian-Wolf [1] estabelece a região de taxa admissível, a qual corresponde às áreas hachuradas do gráfico (região 1  $\cup$  região 2), descritas matematicamente como:

$$R_X + R_Y \geq H(X, Y) \quad (1.1)$$

e

$$R_X \geq H(X | Y), \quad R_Y \geq H(Y | X) \quad (1.2)$$

Surpreendentemente, a soma das taxas  $R_X + R_Y$  pode alcançar a entropia conjunta da fonte  $H(X,Y)$  (como é obtida na codificação conjunta de  $X$  e de  $Y$ ), apesar de serem utilizados codificadores distintos para  $X$  e  $Y$ . A codificação com uma informação auxiliar no decodificador, chamada informação lateral, conforme ilustra a figura 1.3, é um caso especial de codificação distribuída apresentada na figura 1.1.

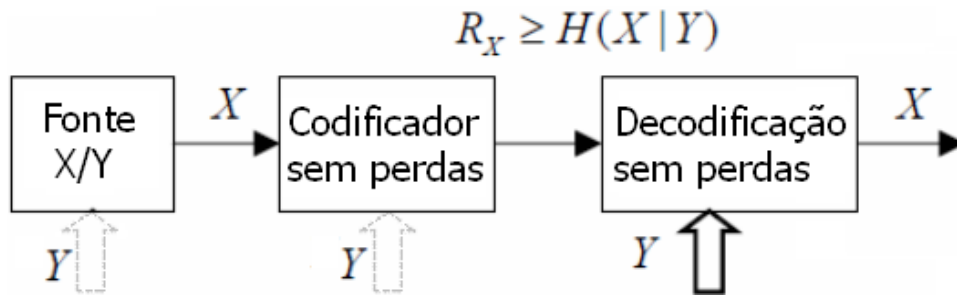


Figura 1.3: Codificação de uma sequência de símbolos aleatórios  $X$  utilizando na decodificação a informação lateral  $Y$  correlacionada a  $X$ .

Assim, segundo a figura 1.3, a fonte produz uma sequência  $X$  com estatísticas que dependem da informação lateral  $Y$ . Neste caso, o interesse está focado no caso em que a informação lateral  $Y$  está disponível no decodificador, mas não no codificador. Visto que a taxa  $R_Y = H(Y)$  é a taxa mínima para (convencionalmente) codificar  $Y$ , uma compressão com a informação lateral disponível no receptor pode alcançar para a fonte  $X$  a taxa  $R_X \geq H(X|Y)$ , sem considerar, obviamente, o acesso do codificador à informação lateral, o que corresponde à quina da região 2 da figura 1.2 formada pela interseção da reta  $R_Y = H(Y)$  com a reta  $R_X = H(X|Y)$ , lembrando-se que  $H(X|Y) < H(X)$  para fontes  $X$  e  $Y$  estatisticamente dependentes.

## 1.2.

### Teorema de Slepian-Wolf para Codificação Distribuída

Embora o teorema de Slepian-Wolf tenha origem na década de 70, foi somente nos últimos anos que aplicações emergentes motivaram tentativas mais sérias de implementação de técnicas práticas de codificação distribuída. Contudo, já era conhecido há muitos anos atrás que a codificação baseada no teorema de Slepian-Wolf é uma “parenta próxima” da codificação de canal [2].

Girod *et al.* [68] considera as sequências  $X$  e  $Y$  apresentadas na figura 1.3 (anterior) como duas sequências binárias igualmente distribuídas. Assim, se  $X$  e  $Y$  são similares, uma hipotética “sequência de erros”  $\Delta = X \oplus Y$  consistirá basicamente de 0s, exceto para alguns 1s que marcam as posições onde  $X$  e  $Y$  diferem. Para proteger  $X$  contra erros  $\Delta$ , pode-se aplicar uma codificação de canal e transmitir somente os bits de paridade resultantes da codificação de  $X$ , sabendo-se que uma estimativa desta sequência,  $Y$ , estará presente no decodificador. No decodificador, concatenam-se os bits de paridade e a informação lateral  $Y$  e realiza-se uma decodificação com correção de erros. Se  $X$  e  $Y$  forem bastante similares, poucos bits de paridade necessitam ser enviados, resultando numa alta compressão.

Enfatizamos que esse método não desempenha correção de erros para proteger a sequência transmitida de erros introduzidos no canal. Porém, tenta minimizar os erros de estimação na decodificação através da modelagem de um “canal de correlação” virtual, o qual captura as dependências estatísticas entre  $X$  e a informação lateral  $Y$ .

Em uma interpretação alternativa, o alfabeto  $X$  pode ser dividido em conjuntos de palavras-código conhecidos como cosets (*codeword sets*) e o codificador envia o índice do coset ao qual  $X$  pertence [3]. O receptor faz a decodificação escolhendo, naquele coset, a palavra-código mais provável de ter sido enviada, segundo a informação lateral. Ambas as interpretações (transmissão dos bits de paridade e do índice do coset, chamado síndrome) são equivalentes. Segundo a interpretação da paridade utilizada em códigos de bloco lineares, nós enviamos um vetor binário  $X_P = X.P$ , onde  $P$  é a matriz-paridade e  $G = [I \ / \ P]$  é a matriz geradora de um código de bloco linear  $C_P$ . Conforme a interpretação de cosets, enviamos a síndrome  $S = X.H$ , onde  $H$  é a matriz de verificação de paridade de um código de bloco linear  $C_S$ . Se  $P = H$ , os *bitstreams* transmitidos de  $X_P$  e de  $S$  serão idênticos. Neste trabalho, utiliza-se esta última abordagem no processo de codificação distribuída de vídeo.

Atualmente, a maior parte das técnicas de codificação distribuída de fonte é derivada de idéias relacionadas à codificação de canal. A principal idéia utilizada nos trabalhos mais recentes sobre codificação distribuída foi iniciada em 1999 por Pradhan & Ramchandram [3]. Inicialmente, eles endereçaram o caso assimétrico de codificação de fonte com informação lateral no decodificador para

fontes gaussianas binárias estatisticamente dependentes utilizando construções de cosets através de treliças. Em seus últimos trabalhos, Pradhan *et al.* [5-8] consideraram o caso em que  $X$  e  $Y$  são codificados com a mesma Taxa.

Alguns autores têm mostrado que métodos baseados em códigos turbo podem ser aplicados à codificação de símbolos estatisticamente dependentes não necessariamente binários [15, 16] e fontes gaussianas [14, 17], assim como podem ser aplicados à codificação de uma única fonte [11, 18-20]. Códigos de canal iterativos também podem ser usados para decodificação conjunta da fonte-canal, pela inclusão das estatísticas tanto da fonte quanto do canal, no processo de decodificação [14, 18-22].

Liveris *et al.* [22-24], Schonberg *et al.* [25, 26] e outros autores [27-30] têm sugerido que os códigos de verificação de paridade de baixa densidade (*Low Density Parity-Check* – LDPC) são alternativas poderosas no lugar dos códigos turbo para codificação distribuída. Com códigos turbo ou LDPC sofisticados, quando a modelagem do canal de correlação reflete de maneira precisa a real correlação entre  $X$  e  $Y$ , o desempenho do código se aproxima, em termos de compressão, do limite de taxa de compressão estabelecido por Slepian-Wolf.

### 1.3. Codificação de Síndrome

Uma interpretação do teorema de Slepian-Wolf pode ser exemplificada no seguinte problema: considere  $X$  e  $Y$  como duas sequências binárias (duas fontes) com comprimento de 3 bits equiprováveis. No entanto,  $X$  e  $Y$  são variáveis aleatórias correlatadas. A correlação entre elas é tal que a distância de Hamming entre  $X$  e  $Y$  é no máximo 1. Isto é, dado  $Y$  (por exemplo, [0 1 0]),  $X$  ou é igual a  $Y$  ([0 1 0]) ou difere no primeiro bit ([1 1 0]), ou difere no bit do meio ([0 0 0]), ou difere no último bit ([0 1 1]). O objetivo é codificar eficientemente  $X$  nos dois cenários mostrados na figura 1.4, para que ele possa ser perfeitamente reconstruído no decodificador.

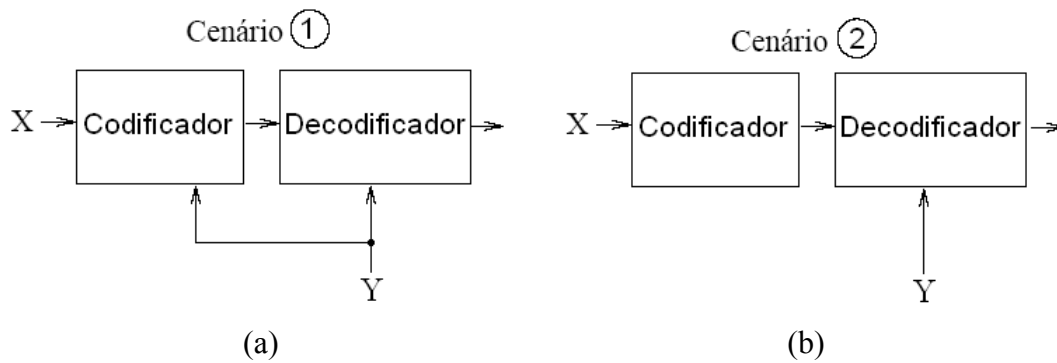


Figura 1.4: (a) Codificador e decodificador utilizam a informação lateral  $Y$ , a qual é correlatada a  $X$ . Aqui,  $X$  pode ser codificado com 2 bits. (b) Aqui, somente o decodificador tem acesso a  $Y$  e ainda assim  $X$  pode ser codificado usando apenas 2 bits.

### Cenário 1

No primeiro cenário (ver figura 1.4(a)),  $Y$  está presente tanto no codificador quanto no decodificador (podemos fazer com que  $Y$  esteja disponível no decodificador usando 3 bits). Aqui,  $X$  pode ser predito a partir de  $Y$ . Assim, pode-se codificar somente o resíduo de  $(X \oplus Y)$ , também chamado de padrão de erros de  $X$  com respeito a  $Y$ , o qual toma 3 valores distintos e, portanto, pode ser codificado com dois bits ao invés de 3. Esta é a menor (ou melhor) taxa possível necessária para codificar  $X$ . Fazendo-se uma analogia à codificação distribuída de vídeo, a fonte  $X$  representaria o bloco de vídeo atual que está sendo codificado e  $Y$  seria o bloco preditor presente no *buffer*, logo, a correlação entre  $X$  e  $Y$  é análoga à correlação temporal entre frames sucessivos (ou macroblocos) de vídeo e, portanto, este modo de codificação é similar à codificação preditiva.

### Cenário 2

No segundo cenário (ver figura 1.4(b)), o codificador para  $X$  não tem acesso a  $Y$ . Assim, o desempenho deste cenário é limitado por aquele do primeiro cenário. Porém, ele conhece a estrutura de correlação entre  $X$  e  $Y$ , ou seja, a distribuição do resíduo  $(X \oplus Y)$  e também sabe que o decodificador tem acesso a  $Y$ . A pergunta que cabe seria: qual a taxa mínima em que o bitstream associado à sequência  $Y$  pode ser transmitido neste caso? A resposta surpreendente é que mesmo no aparentemente pior cenário, pode-se alcançar o mesmo desempenho que no primeiro. Ou seja, também neste segundo cenário,  $X$  pode ser codificado com 2 bits! Isto pode ser feito utilizando o seguinte método: o espaço de palavras-código de  $X$  pode ser particionado em 4 conjuntos, contendo duas palavras-código

cada, com diferença (distância de Hamming) de 3 bits entre as palavras-código de um mesmo *coset*. Assim, os conjuntos são:

***Coset 1*** ([000] e [111])

***Coset 2*** ([001] e [110])

***Coset 3*** ([010] e [101])

***Coset 4*** ([100] e [011])

O codificador para  $X$  identifica o conjunto (*coset*) que contém a palavra código de  $X$  e envia o índice do conjunto, ao invés da palavra-código. Uma vez que há 4 conjuntos (4 *cosets*), eles podem ser indexados com 2 bits. O decodificador, por sua vez, na recepção do índice do *coset*, utiliza  $Y$  para desambiguar o  $X$  correto do conjunto, através da declaração da palavra-código que está mais perto de  $Y$ . Note que a distância (módulo-2) entre  $X$  e  $Y$  é no máximo 1, e a distância entre duas palavras-código em qualquer conjunto é 3. Logo, pode-se obter uma decodificação perfeita.

Por exemplo, se  $Y$  é [0 0 1] e  $X$  é [0 1 1] (ou seja,  $X$  pertence ao *coset 4*), então o codificador envia para o decodificador o índice do *coset 4*. Após receber este índice, o decodificador calcula a distância entre [0 0 1] e [1 0 0] (primeira palavra-código do *coset 4*), que é igual a 2, e entre [0 0 1] e [0 1 1] (segunda palavra-código do *coset 4*), que é igual a 1. Sabendo-se que a distância entre  $X$  e  $Y$  é no máximo 1, [0 1 1] é decodificado como a palavra código observada (estimada). Assim, a correlação entre  $X$  e  $Y$  pode reduzir bastante a taxa de transmissão. Na prática, quanto maior a correlação entre  $X$  e  $Y$ , maior poderá ser a taxa de compressão da informação  $X$ ; ou quanto maior essa correlação, para uma mesma taxa de transmissão de  $X$ , melhor será a qualidade da mensagem reconstruída.

Em seguida faremos algumas observações baseadas neste exemplo, as quais se aplicam também a casos mais gerais, sendo bastante úteis no projeto de codec distribuído.

- ✓ Note que o *coset 1* funciona como um código (de canal) de repetição [86] de distância 3, e os outros conjuntos são *cosets* comuns deste código no espaço de palavras-código de  $X$  [87, 88]. Usou-se um código de canal que é “casado” com a distância de correlação (equivalentemente, com o ruído) entre  $X$  e  $Y$  para



particionar o espaço de palavras-código da fonte  $X$ . Isto resulta em um sistema de codificação com informação lateral que gera uma alta taxa de compressão, idêntico ao sistema de codificação preditiva.

- ✓ Na prática, o particionamento do espaço de palavras-código e a rotulação dos índices dos *cosets* (os rótulos dos índices para *cosets* também são chamados de síndromes) podem ser feitos de uma forma computacionalmente eficiente, através da forma em que são estruturados os códigos do *coset* [78, 79]. Assim, em um sistema de codificação com informação lateral no decodificador, o codificador incorre em uma baixa complexidade computacional.
- ✓ Note que o particionamento de  $X$  é também universal. Isto é, o mesmo particionamento de  $X$  funciona para todos os  $Y$ , não importando o valor de  $Y$ , já que  $X$  e  $Y$  satisfazem a estrutura de correlação. Por exemplo, se  $X$  é  $[0\ 1\ 0]$ , então a mesma codificação para  $X$  (índice do coset 3) será aplicada a todos os casos de  $Y$ , ou seja,  $[0\ 1\ 0]$ ,  $[1\ 1\ 0]$ ,  $[0\ 0\ 0]$  e  $[0\ 1\ 1]$  (todas com uma distância de Hamming de no máximo 1 em relação a  $X$ ). Assim, se  $Y$  toma o valor  $[0\ 1\ 0]$  e por alguma razão vem “corrompido” para  $[1\ 1\ 0]$ , fazendo com que o decodificador tenha uma versão corrompida de  $Y$ ,  $X$  ainda pode ser recuperado corretamente. Isto porque a versão corrompida de  $Y$  também satisfaz a estrutura de correlação. Portanto, diferentemente de uma configuração de codificação preditiva, não existe dependência entre a codificação de  $X$  e o valor da informação correlata  $Y$ , provendo, desta forma, robustez no sistema.

#### 1.4. Principais Objetivos da Tese

A tese tem como objetivo apresentar uma nova arquitetura de codificação distribuída de vídeo, desenvolvendo e analisando algoritmos mais eficientes para a mesma, reduzindo desta forma, a diferença (o *gap*) de desempenho quando comparado ao dos sistemas tradicionais de codificação de vídeo.

Esforços práticos em direção a soluções para codificação distribuída estão apenas começando e a tecnologia ainda não é suficientemente madura. Os resultados disponíveis no *estado da arte* em codificação distribuída de vídeo, em termos de desempenho taxa-distorção, são promissores. Entretanto, é essencial aperfeiçoar e criar ferramentas para o cenário DVC com o propósito de alcançar melhores desempenhos taxa-distorção que os métodos de codificação distribuída disponíveis hoje na literatura. Neste contexto, os principais objetivos desta tese são:

1. Analisar, implementar e aperfeiçoar os métodos e algoritmos mais relevantes na codificação distribuída de vídeo disponíveis na literatura.
2. Desenvolver métodos e ferramentas que aumentem a eficiência dos processos de codificação e decodificação. Partindo dessas premissas, o codec proposto apresenta as seguintes contribuições:

➤ **Arquitetura do codec DVC:** tanto o codificador quanto o decodificador proposto contêm ferramentas extras, as quais são apresentadas abaixo, que contribuem para melhoria da qualidade do frame reconstruído, sem alterar significativamente a taxa. Conseqüentemente, surgem novas opções de configuração do codificador e decodificador utilizando o codec proposto, adequando-os de maneira mais apropriada à sequência de vídeo a ser codificada.

➤ **Arquitetura do módulo de geração da Informação Lateral e otimização de suas ferramentas:** na codificação de vídeo Wyner-Ziv de baixa complexidade, a estimação do movimento e a compensação são realizadas somente no decodificador (o codificador é aliviado desta tarefa), através da utilização de métodos simples de interpolação de frame. O objetivo é alcançar uma maior eficiência na codificação em termos de taxa-distorção, pela geração, no decodificador, de uma melhor estimativa da informação lateral (a partir de frames adjacentes temporalmente), pelo uso eficiente de ferramentas de estimação e compensação

bidirecionais, de interpolação e extrapolação dos frames resultantes.

- **Inclusão de um módulo de Cálculo de Correlação entre frames no codificador:** a adição deste módulo visa aumentar a precisão da medida de similaridade entre os frames Wyner-Ziv e os intraframes, chamados de frames *Key*, aumentando a eficiência do processo de geração da informação lateral.
- **Otimização do processo de quantização Wyner-Ziv:** criando uma nova parametrização e desenvolvendo novas técnicas de quantização mais apropriadas à codificação distribuída, consegue-se uma melhor qualidade do frame reconstruído para uma mesma taxa de compressão.
- **Codificação adaptativa de bitplanes:** o codificador proposto dispõe da opção de enviar somente os bitplanes mais significantes, sem grande perda da qualidade, mas com um significativo ganho de compressão.
- **Otimização da modelagem do canal de correlação:** a implementação deste módulo visa aumentar o desempenho da modelagem da correlação entre o frame codificado e a informação lateral, pelo desenvolvimento de um método mais eficiente de estimação de parâmetros da distribuição característica desta correlação.
- **Inclusão do módulo de Normalização da Informação Lateral:** este módulo tem por finalidade tornar a informação lateral (gerada no decodificador) mais semelhante ao frame Wyner-Ziv que está sendo decodificado, o que resulta em uma maior eficiência no processo de reconstrução do mesmo.

- **Adequação (modificação) do codificador de canal à arquitetura proposta:** nos métodos DVC, os códigos de canal são utilizados para realizar a codificação de fonte, juntamente com outras ferramentas de codificação. Portanto, é essencial otimizar o código de canal, já que este desempenha um papel relevante na codificação e decodificação da fonte. Para solucionar o problema de codificação distribuída de fonte com informação lateral no decodificador e com o desafio, a saber, da taxa adaptativa, propôs-se neste trabalho a utilização de um código de verificação de paridade de baixa densidade, acumulado, o qual já foi desenvolvido por [65], mas utilizando, nesta tese, sua forma irregular e adaptativa à taxa, o qual foi denominado LIA (LDPC Irregular, acumulado e Adaptativo).

## 1.5. Apresentação do Trabalho

Esta tese descreve em detalhes uma solução de arquitetura para codificação distribuída de vídeo, baseada nos princípios de Wyner-Ziv e Slepian-Wolf, dispondo de novas ferramentas, cujo objetivo é melhorar o desempenho do codec DVC.

- ✓ A contextualização e a motivação para este trabalho são apresentadas no capítulo 1, juntamente com a definição dos principais objetivos e contribuições da tese.
- ✓ No capítulo 2 são apresentados os métodos de codificação desenvolvidos por Wyner e Ziv aplicados à codificação distribuída de vídeo.
- ✓ No capítulo 3, descreve-se o funcionamento do código de canal implementado e adequado ao codec DVC proposto.
- ✓ No capítulo 4, descreve-se detalhadamente o funcionamento do codec proposto, descrevendo todo o processo de codificação e decodificação de vídeo baseado na teoria de Wyner-Ziv, assim como todas as ferramentas e técnicas utilizadas nesse processo, inclusive as que não

fazem parte das contribuições deste trabalho, mas que são necessárias à sua compreensão.

- ✓ No capítulo 5 são feitas as simulações relativas ao codec proposto, com o objetivo de avaliar as técnicas e ferramentas propostas, assim como validar o codec de uma maneira geral.
- ✓ Enfim, no capítulo 6, são apresentadas as conclusões e sugestões para trabalhos futuros.

## **1.6. Publicações**

Parte do trabalho apresentado nesta tese resultou na publicação em uma conferência internacional.

- R. Mello, C. Escouper, M. A. Grivet, “A Proposed Framework for Implementation of Distributed Video Coding”, 17th International Conference on Systems, Signals and Image Processing, Março 2010.