UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

PROPOSTA DE UM ESQUEMA DE CODIFICAÇÃO DE VÍDEO A BAIXAS TAXAS DE TRANSMISSÃO PARA COMUNICAÇÕES MÓVEIS CELULARES

por

VICTOR ENRIQUE VERMEHREN VALENZUELA

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco como parte dos requisitos para a obtenção do grau de **Mestre em Engenharia Elétrica.**

ORIENTADOR: VALDEMAR CARDOSO ROCHA JR, Dr
CO-ORIENTADOR: MARCELO SAMPAIO DE ALENCAR, Dr

Recife, Outubro de 2006.

© Victor Enrique Vermehren Valenzuela, 2006.

V161p **Valenzuela, Victor Enrique Vermehren.**
Proposta de um esquema de codificação de vídeo a baixas taxas de transmissão para comunicações móveis celulares. – Recife: O Autor, 2006.
v, 63 folhas. : il. ; fig., tabs.

Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Engenharia Elétrica, 2006.

Inclui bibliografia.

1. Engenharia elétrica. 2. Codificação de vídeo – Melhoria de desempenho. 3. Decodificador H.263 – Cancelamento de erro. 4. Decodificador H.263 – Recuperação de sincronismo. 5. Telefonia móvel. I. Título.

621.3 CDD (22.ed.)

UFPE
BCTG/2006-130

Universidade Federal de Pernambuco

Pós-Graduação em Engenharia Elétrica

PARECER DA COMISSÃO EXAMINADORA DE DEFESA DE
TESE DE MESTRADO PROFISSIONAL DE

VICTOR ENRIQUE VERMEHREN VALENZUELA

TÍTULO

"PROPOSTA DE UM ESQUEMA DE CODIFICAÇÃO DE VÍDEO A BAIXAS TAXAS DE TRANSMISSÃO PARA COMUNICAÇÕES MÓVEIS CELULARES"

A comissão examinadora composta pelos professores: VALDEMAR CARDOSO DA ROCHA JÚNIOR, DES/UFPE, MARCELO SAMPAIO DE ALENCAR, DEE/UFCG, RAFAEL DUEIRE LINS, DES/UFPE e FRANCISCO MADEIRO BERNARDINO JÚNIOR, DEI/UNICAP sob a presidência do primeiro, consideram o candidato **VICTOR ENRIQUE VERMEHREN VALENZUELA** **APROVADO**.

Recife, 30 de outubro de 2006.

JOAQUIM FERREIRA MARTINS FILHO
Coordenador do PPGEE

VALDEMAR CARDOSO DA ROCHA JÚNIOR
Orientador e Membro Titular Interno

RAFAEL DUEIRE LINS
Membro Titular Interno

MARCELO SAMPAIO DE ALENCAR
Co-Orientador e Membro Titular Externo

FRANCISCO MADEIRO BERNARDINO JÚNIOR
Membro Titular Externo

AGRADECIMENTOS

Em primeiro lugar gostaria de agradecer aos meus pais, Silvia e Roland, que não mediram esforços pela minha educação.

A minha querida e amada esposa Waldenira que em momento algum permitiu que Eu esmorecesse, sempre me incentivando, levantando minha motivação e não reclamando minhas ausências.

Aos meus filhos Victor e Lucas que se prontificaram para participar das simulações desta dissertação.

Aos meus orientadores Marcelo Alencar e Valdemar Rocha que muito contribuíram com seus conhecimentos e incentivo.

E finalmente, a todos aqueles que direta ou indiretamente contribuíram de alguma forma para tornar possível a realização deste trabalho.

Resumo da Dissertação apresentada à UFPE como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica.

PROPOSTA DE UM ESQUEMA DE CODIFICAÇÃO DE VÍDEO A BAIXAS TAXAS DE TRANSMISSÃO PARA COMUNICAÇÕES MÓVEIS CELULARES

Victor Enrique Vermehren Valenzuela

Outubro/2006

Orientador: Valdemar Cardoso da Rocha, Dr.

Co-orientador: Marcelo Sampaio Alencar, Dr.

Área de Concentração: Telecomunicações.

Palavras-chave: Codificação de vídeo, H.263, EREC.

Número de Páginas: 63.

RESUMO: Codificação de vídeo abaixo de 64 kbps é essencial para serviços e aplicações envolvendo videofone e sistemas multimídia. O aumento de interesse e demanda por telefonia móvel, TV interativa e serviços de multimídia tem motivado pesquisas em codificação de vídeo no mundo. Neste trabalho é apresentada uma técnica de cancelamento de erro e recuperação de sincronismo no decodificador H.263 com a finalidade de melhorar o desempenho da decodificação de vídeo a baixas taxas de transmissão.

Abstract of Dissertation presented to UFPE as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

PROPOSAL FOR LOW BIT RATE VIDEO CODING FOR CELLULAR PHONES COMMUNICATIONS

Victor Enrique Vermehren Valenzuela

October /2006

Supervisor(s): Valdemar Cardoso da Rocha, Dr; Marcelo Sampaio Alencar, Dr.
Area of Concentration: Telecommunications.
Keywords: Video coding, H.263, EREC.
Number of Pages: 63.

ABSTRACT: *Very low bit rate coding below 64 kbit/s will be an important issue in forthcoming visual services such as videophone and multimedia systems. The trend of increasing interest in, and demand for, mobile phone, interactive TV and multimedia services, has led to significant acceleration of video coding research worldwide.*

This paper introduces and presents simulation results of an error canceling technique and synchronization recovery with the H.263 decoder for improving the quality of decoded video at very low bit rate. The paper concludes with a discussion on future directions regarding possible improvements to the coding algorithm.

SUMÁRIO

Introdução..... i

Objetivo..... iv

Conteúdo..... v

1. CODIFICAÇÃO DE FONTE..... 1

1.1. Codificação de Fonte de Vídeo e Imagem..... 2

1.2. Codificação de Fonte Espacial..... 4

1.2.1. Transformada Discreta de Coseno..... 4

1.2.2. Quantização..... 6

1.2.3. Codificação de Entropia..... 7

1.3. Codificação de Fonte Temporal..... 11

1.3.1. Diferença de quadro..... 11

1.3.2. Compensação e estimativa de movimento..... 12

1.4. Qualidade de Cena..... 13

2. PADRONIZAÇÃO DA CODIFICAÇÃO DE VÍDEO NO H.263.....15

2.1. O codificador de fonte H.263.....15

2.1.1 Formatos do Quadro de Vídeo (Cena).....16

2.1.2 Organização das Camadas do Quadro de Vídeo.....17

2.1.3 Estimador e Compensador de Movimento.....18

2.1.4 Transformada Discreta de Coseno (DCT).....18

2.1.5 Quantização.....18

2.1.6 Codificação de Entropia.....19

2.1.7 Armazenador de Quadros.....19

2.1.8 Controle de Codificação.....19

2.2. Sintaxe do Trem de Pulsos H.263.....20

2.2.3. Camada de Cena.....20

2.2.4. Camada de Grupo de Blocos.....21

2.2.5. Camada de Macroblocos.....21

2.2.6. Camada de Blocos.....22
2.3. Erro recuperável adicionado ao Padrão de Codificação de Vídeo.....23
2.3.3. H.261.....23
2.3.4. H.263 Versão 2.....23

3. CODIFICAÇÃO DE CANAL.....26
3.1. Código Linear de Bloco.....27
3.2. Códigos Cíclicos.....28
3.2.1 Códigos BCH.....29
3.2.2 Códigos Reed-Solomon.....29
3.3. Códigos Convolucionais.....30
3.3.1 Código RCPC.....31

4. O CODIFICADOR DE VÍDEO MODIFICADO.....34
4.1. Características da Seqüência de Vídeo.....34
4.2. Significância da Informação de Fonte.....35
4.2.1. Propagação de Erro no trem de pulsos do Codificador H.263.....38
4.3. Proteção de Erro Não-Uniforme.....41
4.3.1. Classificação da Significância.....41
4.4. Sincronização.....44
4.4.1. O Algoritmo EREC.....45
4.5. Cancelamento de Erro.....53
4.5.1 Cancelamento de erro no domínio da freqüência.....53
4.5.2 Cancelamento de erro no domínio do tempo.....55

5. SIMULAÇÃO E RESULTADOS.....56
5.1 Resultados.....57
5.1.1 Comparação Quantitativa.....57
5.1.2 Comparação Qualitativa.....58
5.2 Conclusão.....59

REFERÊNCIAS BIBLIOGRÁFICAS..... 61

Lista de Figuras

1.1 Diagrama em blocos de um codificador de fonte para compressão de imagem e vídeo.....3
1.2 Exemplo de Transformada Discreta de Coseno em bloco de 8x8 pixels.....5
1.3 Resultado de Transformada Discreta de Coseno Inversa.....6
1.4 Resultado de coeficientes DCT quantizados.....7
1.5 Estimativa de movimento para codificação interquadro.....12
2.1 Codificador de Fonte H.263.....16
2.2 Subdivisão de um quadro QCIF em 99 macroblocos de 16x16.....17
2.3 Sintaxe do trem de pulsos H.263.....20
2.4 Arranjo de blocos dentro de um macrobloco.....22
2.5 Caminho em zig-zag na transformação dos coeficientes.....23
2.6 Ilustração das diferenças posicionais entre usar os cabeçalhos GOB do H.263 padrão e o H.263 Versão 2, anexo K(modo estruturado de faixas) para melhorar o erro recuperável.....24
3.1 Transmissão com codificação de canal e proteção de erro.....26
3.2 Codificação de bloco sistemática *k/n*, onde *k* = 4 e *n* = 7.....28
3.3 Modelo de codificador convolucional para explicar os termos básicos.....30
3.4 Diagrama de estado do código RCPC dado na Tabela 3.2.....32
3.5 Codificador RCPC com código mãe à taxa de ¼.....33
4.1 Erro residual em palavras código DQUANT de comprimento fixo.....39
4.2 Erro residual em palavras código MCBPC de comprimento variável causando erro de decodificação de fonte de MCBPC e a subseqüente palavra código CBPY. Pode ocorrer a propagação do erro ao restante do trem de pulsos 39
4.3 Classificação do trem de pulsos do H.263 dentro das várias classes de Significância. Quanto maior o número, maior a significância.....43
4.4 Codificador RCPC com taxa de código mãe de 1/4 e matrizes puncionadas às taxas de 1/3, 1/2 e 2/3.....44
4.5 Diagrama em blocos de sistema usado na transmissão de vídeo H.263 protegido contra erro não uniforme sobre canais de rádio móveis.....44
4.6 Reordenação EREC. No final de cada estágio todas as fontes de blocos são deslocadas para baixo por um.....46

4.7 Decodificação EREC. Ao final de cada estágio os blocos fonte são deslocados para baixo de uma unidade.....49
4.8 Saída do codificador EREC.....50
4.9 Exemplo de propagação de erro sem EREC.....51
4.10 Exemplo de propagação de erro com EREC.....52
4.11 (a) Quadro de uma seqüência mostrando o artefato visual, que lembra um padrão xadrez. (b) O mesmo quadro com aplicação do cancelamento de erro.....53
5.1a PSNR da seqüência LUCAS.....57
5.1b PSNR da seqüência VICTINHO.....58
5.2a Seqüência VICTINHO original Quadro 54.....58
5.2b Seqüência VICTINHO decodificada sem EREC.....58
5.2c Seqüência VICTINHO decodificada com EREC.....58
5.2d Seqüência LUCAS original Quadro 02.....59
5.2e Seqüência LUCAS decodificada sem EREC.....59
5.2f Seqüência LUCAS decodificada com EREC.....59

Lista de Tabelas

1.1 Exemplo de diferença de um código prefixo.....9
1.2 Construção de palavras código de Huffman.....10
2.1 Formatos de cena do codificador H.263.....17
3.1 Propriedades de Códigos RS.....29
3.2 Gerador Polinomial e as matrizes puncionadas para códigos RCPC com código mãe à taxa de ¼.....32
4.1 Códigos de comprimento fixo para DQUANT.....38
4.2 Códigos de comprimento variável para MCBPC (inter).....40
4.3 Códigos de comprimento variável para CBPY.....40
4.4 Gerador Polinomial e as matrizes puncionadas para códigos RCPC às taxas de 1/3, 1/2 e 2/3.....43
4.5 Códigos de comprimento fixo para LEVELS.....54
5.1 Parâmetros de simulação para o sistema de vídeo com erro recuperável.....56

Lista de Siglas e Símbolos

3G – *3rd Generation*

AIC – *Advanved Intra Coding*

ARQ – *Automatic Repeat Request*

BCH – Código Bose-Chaudhuri-Hocquenghem

BER – *Bit Error Rates*

CCI – *Co-Channel Interference*

CD - *Compact Disc*

CIF - *Common Intermediate Format*

COD – COD*ed macroblock indication*

CPM - *Continuous Presence Multipoint*

DCT – *Discrete Cosine Transform*

DF – *Deblocking Filter*

DFT – *Discrete Fourier Transform*

DVD – *Digital Video Disk*

EOS - *End of Sequence*

EREC - *Error Resilient Entropy Coding*

ERPC – *Error Resilient Positional Code*

FEC – *Forward Error Correction*

GBSC - *Group of Block Start Code*

GF - *Galois Field*

GN - *Group Number*

GOB – *Group of Block*

H – Notação para Entropia

H.263 – Padrão de codificação de vídeo para baixas taxas de transmissão

HVS – *Human Visual System*

IDCT - *Inverse Discrete Fourier Transform*

ISDN - *Integrated Service Digital Network*

ITU – *International Telecommunications Union*

JPEG - *Joint Pictures Experts Group*

$\overline{L}$ - Notação para comprimento médio de uma palavra código

MPEG – *Motion Picture Experts Group*

MSE – *Mean Squared Error*

MV – *Motion Vector*

MVD – *Moving Vector Data*

MQ – *Modified Quantization*

PDA - *Portable Data Assistent*

PEI – *Picture Extra Insertion*

PSC – *Picture Start Code*

PSNR – *Peak Signal to Noise Ratio*

PSTN - *Public Switched Telephony Network*

PTYPE - *Picture Type Information*

QCIF – *Quarter Common Intermediate Format*

RCPC - *Rate Compatible Punctured Convolutional Code*

RLE – *Run Length Encoding*

RS – Código *Reed-Solomon*

SAD - *Sum of Absolute Differences*

SPIHT - *Set Partitioning in Hierarchical Trees*

TR – *Temporal Reference*

INTRODUÇÃO

A indústria da tecnologia sem fio tem canalizado grandes esforços nos últimos anos desenvolvendo soluções para sistemas de terceira geração (3G). Nestes sistemas 3G pretende-se proporcionar aos usuários recursos avançados, especialmente suporte a serviços de multimídia, *roaming* global e serviço de dados de alta velocidade. A *International Telecommunications Union* (ITU), trabalhando em conjunto com órgãos que representam a indústria mundial, implementaram o programa IMT-2000 para o desenvolvimento de padrões para estes sistemas 3G.

As novas redes permitirão que os serviços venham a ser moldados a cada usuário individualmente e a interatividade abrirá espaço para aplicações ainda desconhecidas. Serviços de vídeo tal como *video-on-demand* e a difusão dos padrões de compressão de imagem recentemente criados (JPEG – *Joint Picture Experts Group*, MPEG – *Motion Picture Experts Group* e H26x) refletem a corrente migração dos telefones para terminais multimídia. A definição do primeiro padrão internacional para taxas abaixo de 64 kbps, o padrão H263, que permite transmissão de vídeo com razoável qualidade por meio das redes de telefonia existentes, dá assim um importante passo para essa migração. Quando essas novas tecnologias vierem a estar disponíveis as demandas de serviços multimídia em comunicações móveis aumentará. Mas apesar da tecnologia para estes serviços já estar disponível, ainda falta a definição de padrões. Isto se deve à natureza dos canais de comunicações móveis, limitados em termos de largura de banda e taxas de erros. Conseqüentemente, este futuro da multimídia sem fio motiva a investigação de possíveis esquemas de vídeo que se adaptem aos requisitos padronizados para esta nova tecnologia.

Hoje em dia as taxas de dados abaixo de 64 kbps estão na faixa de 14 a 30 kbps. Já para as redes 2.5G/3G projetam-se taxas da ordem de 64 kbps a 2Mbps. O objetivo deste aumento na taxa de dados é a expansão dos serviços para muito além da simples comunicação de voz, chegando, por exemplo, a mensagens escritas, fotos com mensagens escritas e voz, vídeo-clip e finalmente o objetivo maior: a conversação com áudio e vídeo em tempo real.

Estes serviços terão ainda que dar suporte a aplicações tais como um e-mail ou um vídeo incluindo notícias, economia, esportes, apresentações, tráfego de trânsito, relatórios do tempo. Usos domésticos incluirão jogos, segurança ou primeiros socorros.

Videofone interativo *full-duplex* enriquecerá a comunicação, podendo ser usado para *chat* ou outro tipo de comunicação de vídeo.

O aumento da complexidade de vídeo sem fio cria novos desafios de projeto. Por exemplo, o vídeo é comprimido a uma razão muito maior que a voz de modo que vem a ser muito mais sensível a erros. A qualidade do vídeo enviado por meio de uma rede 3G pode ser crítica para o sucesso de algumas aplicações.

Existem gargalos técnicos para acessar vídeo sem fio, tais como a não confiabilidade causada pela propagação multipercurso e a severa interferência advinda de outras transmissões. Já é sabido que o ambiente sem fio é freqüentemente não estacionário devido ao desvanecimento do sinal transmitido. Desvanecimento é a conseqüência da somatória de múltiplas cópias atrasadas do sinal com diferentes relações de fase devido às reflexões do sinal em várias superfícies. Além disso, usuários de celulares podem mover-se quando falando ao telefone, de modo que o ambiente em volta, o qual afetará a comunicação, está sempre mudando.

Um outro desafio no projeto de sistemas de comunicação sem fio é aliviar a interferência co-canal (CCI – *Co-Channel Interference*), que vem a ser um problema severo quando muitos usuários usam o canal ao mesmo tempo. Devido à natureza variante no tempo do canal de propagação e a imprevisibilidade da interferência, taxas de erros de bit (BER – *Bit Error Rates*) variarão amplamente causando erros em surto que podem determinar a perda total da informação de vídeo. Medidas estão sendo tomadas para combater estes efeitos: *turbo codes*, diversidade de transmissão, codificadores de taxa variável com mecanismos de controle da taxa de bit, *interleaving* e avançados algoritmos de detecção multi-usuário. Apropriado planejamento celular e novas tecnologias de antenas ajudarão a reduzir o impacto da interferência co-canal.

Em transmissões de vídeo a perda de quadros e erros em bits tende a causar notável degradação da qualidade da cena. Técnicas de correção e cancelamento de erros são recursos para aumentar a eficiência dos codificadores minimizando a perda de qualidade. Técnicas de correção de erros, que removem o erro e restauram a informação original, aplicam-se perfeitamente a dados sem a condição de tempo real. Já na transmissão de vídeo que terá de ser em tempo real não poderão ser tolerados atrasos como em dados com a tradicional requisição de repetição automática (ARQ – *Automatic Repeat Request*).

O projeto de videofones celulares 3G certamente será um desafio para engenheiros de sistema, tanto nos algoritmos de processamento de vídeo como na implementação da parte física para se sobrepor às adversidades inerentes à transmissão de vídeo sobre canais sem fio.

OBJETIVO

Esta dissertação tem como meta ajudar a definir os desafios de projeto de celulares 3G com vídeo componente e os métodos usados para **projetar/desenvolver/adaptar um sistema de codificação/decodificação de vídeo para comunicações móveis**, reunindo e/ou adaptando os recursos existentes atualmente para implementar esse sistema.

Para isso algumas metas foram traçadas e alcançadas:

- Primeiro, desenvolver, adaptar ou combinar um esquema de codificação/ decodificação de vídeo que garantisse à saída uma qualidade aceitável. Para tal foi realizada ao longo dos capítulos uma explanação dos parâmetros de medição dessa qualidade de cena, os métodos de codificação de imagem, o algoritmo de compressão e descompressão de imagens, e os formatos de vídeo que se adaptem às especificações para videofones móveis.
- Segundo, descrever o codificador padrão de vídeo H.263 imprescindível para este esquema.
- E por último, a reunião desses recursos em um codificador e sua conseqüente simulação em um software gentilmente cedido pela UBLive que o adaptou às características/ especificações aqui descritas.

Durante este desenvolvimento alguns problemas deste sistema proposto teriam que ser resolvidos, como manter a qualidade da imagem e o desempenho relação sinal ruído a baixas taxas de transmissão. Para isso, foram reunidas algumas técnicas de cancelamento de erros e sincronização de dados e palavras código de comprimento fixo e variável, além de esquemas de codificação de fonte e canal.

O estudo e a simulação do desempenho deste *codec* por meio de canais suscetíveis a erro tais como Gauss, Rice e Rayleigh ficará para uma continuação deste trabalho, mas mesmo assim em várias etapas do estudo foram descritas e colocadas situações de transmissão nesses tipos de canais.

CONTEÚDO

O Capítulo 1 trata sobre codificação de fonte destacando a necessidade de compressão de dados. Para isso, descreve alguns algoritmos, códigos e quantificadores utilizados.

No Capítulo 2 há uma descrição do padrão de codificação de vídeo (H.263) seguido pelo codificador de vídeo proposto, incluindo seus blocos e sintaxe.

No Capítulo 3 é visto a situação do controle de erro na transmissão da informação em um canal, introduzindo algumas técnicas de codificação de canal disponíveis.

No Capítulo 4 é introduzido um método para gerar seqüências de vídeo com erro recuperável baseado em codificação combinada de canal/fonte e técnicas de cancelamento de erro e sincronização, objeto do codificador desta dissertação.

O Capítulo 5 contém a simulação prática do codificador objeto desta dissertação e seus respectivos resultados.

E finalmente, nas Conclusões destacam-se as contribuições do trabalho, as perspectivas futuras de trabalhos que podem ser realizados sobre codificação de vídeo, as tendências e uma possível continuação deste trabalho.

1. CODIFICAÇÃO DE FONTE

O tratado de C. E. Shannon [1] em 1948 foi o primeiro a definir matematicamente a quantidade de informação presente em uma fonte e colocar as fundações para a teoria da informação. Essa teoria introduziu alguns fundamentos dentro das comunicações. É nessa teoria que a codificação de fonte é baseada. Codificação de fonte, que inclui compressão de dados, é o processo pela qual a informação irrelevante e redundante de uma fonte é removida, resultando em considerável economia de espaço de memória ou largura de banda de transmissão. A informação pode ser reconstruída dentro de certos critérios de aceitação por um processo reverso conhecido como decodificação de fonte ou descompressão de dados. Quando a codificação de fonte é aplicada a imagens ou seqüência de imagens é conhecida como compressão de imagem ou de vídeo, respectivamente. Aplicações que se baseiam na compressão de imagens são a videoconferência, armazenamento de imagens digitais e armazenamento e transmissão de filmes.

O grande obstáculo enfrentado para o armazenamento de uma grande quantidade de informações, como a necessária para o caso de um filme, é a limitação da mídia empregada. Ou seja, a necessidade de compressão de vídeo ou seqüência de imagens deve-se às grandes larguras de banda requeridas para transmitir ou armazenar tais dados. Tome-se, por exemplo, uma seqüência de vídeo sendo transmitida a 30 quadros por segundo, com uma resolução de 640 x 480 pixels a 24 bits de cores. A taxa de transmissão requerida seria de (30 x 640 x 480 x 24)/8 = 27.648 megabytes por segundo (ou 221.184 megabits por segundo). Dada a capacidade do *compact disc* (CD) de uso mais comum que é de 650 MB, somente haveria aproximadamente 23.5 segundos de dados, o suficiente para um curto comercial de televisão. A compressão pode resolver este problema de forma que um CD pode conter um filme com 100 minutos de duração.

Mas mesmo quando há a redução da resolução espacial e temporal da seqüência de imagem (por exemplo, QCIF – *Quarter Common Intermediate Format*) a taxa resultante ainda é muito alta para os canais existentes hoje. Redes tais como PSTN (*Public Switched Telephony Network*) e ISDN (*Integrated Services Digital Network*) além de redes locais de computadores funcionam com baixas taxas de transmissão e não podem suportar a quantidade de dados provenientes da transmissão de vídeo sem compressão. Mais crítica ainda é a situação de sistemas de transmissão sem fio como os usados por

telefones celulares e PDA's (*Portable Data Assistent*). Tais redes operam em taxas da ordem de 14,4 a 56 kbps. Conseqüentemente, é essencial um eficiente tratamento da seqüência de imagens, de modo que possam ser transmitidas adequadamente.

Existem duas técnicas básicas para se obter a redução da quantidade de dados para se enviar uma informação. A primeira, sem perdas, quando a descompressão permite a recuperação total da informação que estava presente na entrada do sistema (exemplo, arquivos PKZip). A compressão sem perdas seria a ideal, porém ela não fornece a taxa de compressão desejável nas aplicações práticas mencionadas. Se mesmo com a eliminação da redundância não se conseguir ainda uma compressão nos níveis desejados para o sinal, deve-se apelar para um recurso adicional que seria o descarte de parte das informações, aproveitando o fato que algumas informações do sinal de vídeo não são percebidas pelo sistema visual humano. Obtém-se nesse caso a segunda técnica, um sistema de compressão que descarta a informação irrelevante, ou seja, a recuperação dos dados não é total, mas sim com perdas. Essa perda de dados tem um limite permissível, pois influi diretamente na qualidade de imagem. Contudo, a quantidade de memória requerida nesse caso é muito menor que a requerida para compressão sem perdas.

1.1 CODIFICAÇÃO DE FONTE DE VÍDEO E IMAGEM

Num sinal de vídeo existem dois tipos de informação: a imprevisível e a previsível. A primeira componente é denominada entropia e é a informação essencial dentro do sinal. O restante é chamado de redundância, já que é uma repetição dos dados e pode, sob certas condições, ser descartado.

A redundância pode ser espacial, quando ela está presente em grandes áreas da imagem e os pixels adjacentes têm o mesmo ou quase o mesmo valor. A redundância também pode ser temporal quando uma imagem é repetida sucessivamente ou não muda em um determinado intervalo de tempo.

Os sistemas de compressão conceitualmente operam separando a entropia da redundância. Somente a entropia deve ser gravada ou transmitida ficando por conta do decodificador reintroduzir a redundância do sinal transmitido. Na realidade um decodificador desse tipo seria muito complexo e caro, então o que ocorre na prática é o envio de uma parte da redundância.

A codificação de imagem pode ser tratada como um caso especial de codificação de vídeo, bem como a codificação de vídeo pode ser considerado uma seqüência de imagens.

Numa conversação em vídeo imaginária o que se pode observar é a existência de uma grande quantidade de informação redundante, já que basicamente haveria uma seqüência de imagens em que, na maior parte delas, apenas ocorre movimento labial (codificação envolvendo rostos somente). E dentro de cada quadro esta redundância também existe devido à correlação existente entre os pixels vizinhos (correlação espacial). A redundância ou similaridade presente entre dois quadros consecutivos também é chamada de correlação temporal.

O diagrama em blocos de um sistema codificador de fonte pode ser representado como na Figura 1.1.

O processador pode realizar a compressão com perdas ou sem perdas. No caso de compressão com perdas é necessário um processamento adicional para considerar, por exemplo, as diferenças entre os quadros, bem como calcular os vetores de movimento requeridos no decodificador para a compensação de movimento. Outro ponto importante é que a saída pode ser reduzida ainda mais pela codificação por entropia, não fazendo diferença a compressão ser com perdas ou sem perdas.

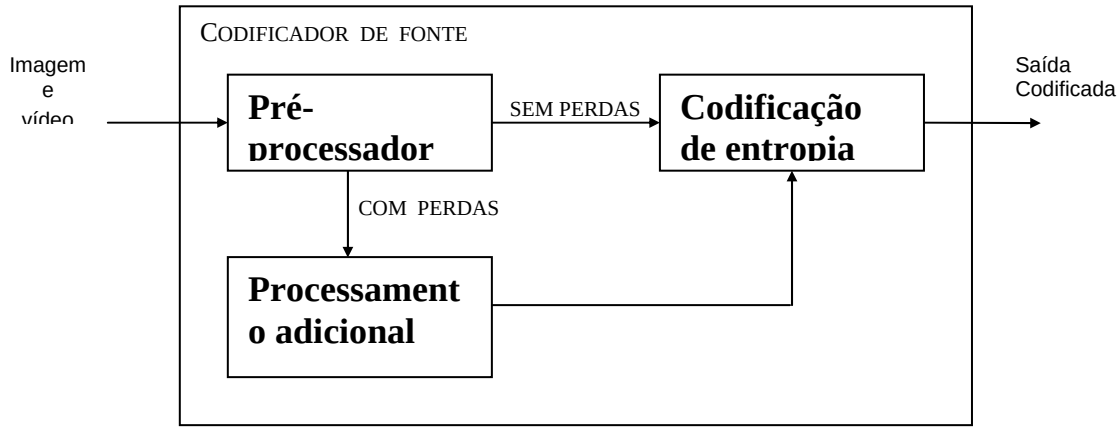


Figura 1.1 - Diagrama em blocos de um codificador de fonte para compressão de imagem e vídeo.

1.2 CODIFICAÇÃO DE FONTE ESPACIAL

Na codificação de fonte espacial, a principal área de compressão é a imagem do quadro. Isto também é conhecido como codificação intraquadro, uma expressão utilizada extensamente. As técnicas de compressão podem ser divididas em dois tipos: sem perdas e com perdas. Exemplos de esquemas de compressão de imagens sem perdas incluem codificação de *Huffman*, codificação pelo comprimento (RLE – *Run Length Encoding*). O padrão JPEG (*Joint Pictures Experts Group*) [2] é um exemplo de formato de imagem com perdas. A codificação intraquadro remove a redundância espacial no domínio da freqüência. Dados com alta correlação espacial podem ser bastante comprimidos no domínio da freqüência.

Outro fato importante, já mencionado, é que existem dentro de cada quadro dados que, quando removidos, são imperceptíveis do ponto de vista visual humano. Estudos fisiológicos [Gonzáles, 2000] têm mostrado que o sistema visual humano é mais sensível às componentes de baixa freqüência do que de alta freqüência. Isto se deve ao fato de que as componentes de baixa freqüência são equivalentes ao valor de luminância médio de toda a imagem, ou em outras palavras, quão luminosa ou escura é a imagem. Componentes de alta freqüência representam as áreas de definição da imagem, tais como as bordas dos objetos ou áreas de mudanças da tonalidade da cor.

1.2.1 A Transformada Discreta de Coseno

Para se realizar a codificação espacial de um sinal, o primeiro passo consiste em efetuar uma análise das freqüências espaciais usando uma transformada.

A saída de uma transformada consiste em um conjunto de coeficientes que descrevem as freqüências que um sinal contém e como as diversas harmônicas que formam esse sinal se distribuem.

A Transformada Discreta de Coseno (DCT – *Discrete Cosine Transform*) é um dos métodos mais usados para compressão de imagens com perdas no domínio da freqüência, inclusive nos padrões de codificação JPEG e H.263.

A DCT pode ser conceituada como uma transformação no domínio espacial de um bloco de 8 X 8 pixels para um bloco de mesmo formato com 64 coeficientes no domínio da freqüência. A Figura 1.2 mostra um exemplo de tal transformação.

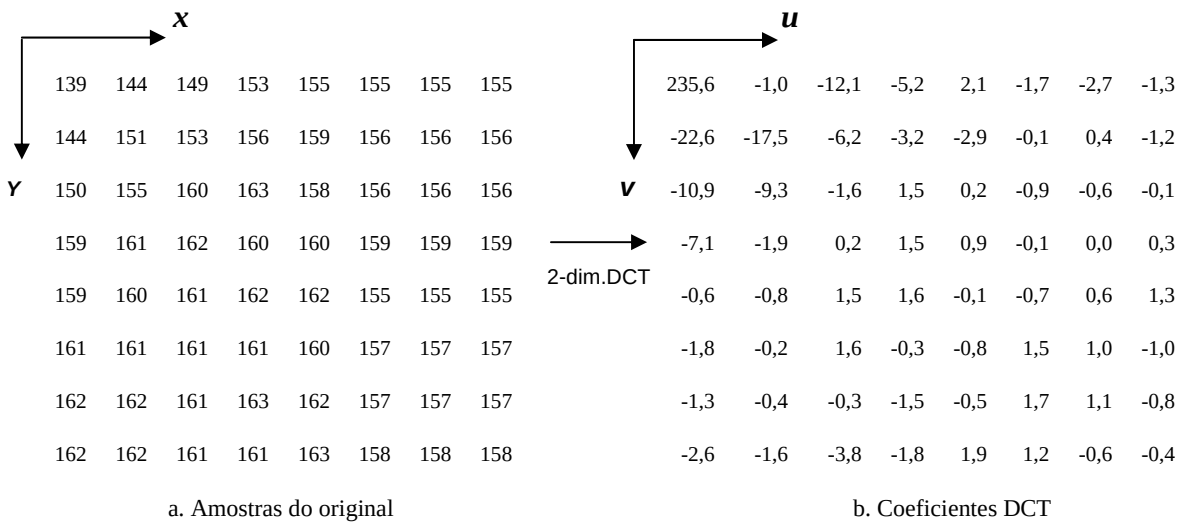


Figura 1.2 – Exemplo de Transformada Discreta de Coseno em bloco de 8x8 pixels.

A DCT é similar à Transformada Discreta de Fourier (DFT – *Discrete Fourier Transform*) exceto que todos os termos na expansão são reais em vez de complexos. Para um bloco de imagem $f(x, y)$ de colunas e linhas N x N, a DCT - 2D direta, $F(u, v)$, é dada por

$$F(u,v) = \frac{4\,c(u)\,c(v)}{N^2} \times \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)\,\cos\left[\frac{(2x+1)\,\pi\,u}{2N}\right]\,\cos\left[\frac{(2y+1)\,\pi\,v}{2N}\right], \qquad (1.1)$$

em que N é o comprimento de um dos lados da imagem, x e y as coordenadas de pixel,

$$c(u) = \begin{cases} \frac{1}{\sqrt{2}} & u = 0 \\ 1 & \text{caso contrário.} \end{cases}, \quad c(v) = \begin{cases} \frac{1}{\sqrt{2}} & v = 0 \\ 1 & \text{caso contrário} \end{cases}.$$

$u = 0, ..., N - 1$, e $v = 0,..., N – 1$. As coordenadas de pixel são x e y, enquanto u e v são as coordenadas dos coeficientes.

Da mesma forma, a DCT inversa (IDCT) é dada por uma equação similar:

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u,v) \times \frac{4\,c(u)\,c(v)}{N^2} \times \cos\left[\frac{(2x+1)\,\pi u}{2N}\right] \cos\left[\frac{(2y+1)\,\pi v}{2N}\right].$$

(1.2)

Para valores elevados de *u* e *v* as componentes de DCT têm pequeno valor. Isto significa que a DCT retorna valores mais insignificantes em altas freqüências do que nas baixas freqüências. O coeficiente da DCT em *F*(0, 0), representa o valor médio de toda a imagem e é considerado o coeficiente de maior importância. Já os mais afastados da origem, *F* (*N* – 1, *N* - 1), são os coeficientes de menor importância.

A Figura 1.3 mostra os resultados de uma transformada inversa de cada um dos coeficientes de uma DCT 8 x 8. No caso do sinal de luminância, o coeficiente do topo à esquerda é o brilho médio ou a componente DC de todo o bloco. Movendo-se para a direita na linha superior, a freqüência espacial horizontal aumenta. Com movimento para baixo na coluna da esquerda, a freqüência espacial vertical aumenta. Nas imagens reais podem ocorrer simultaneamente freqüências verticais e horizontais diferentes e, com isso, um coeficiente em algum ponto dentro do bloco vai representar todas as combinações horizontais e verticais possíveis.

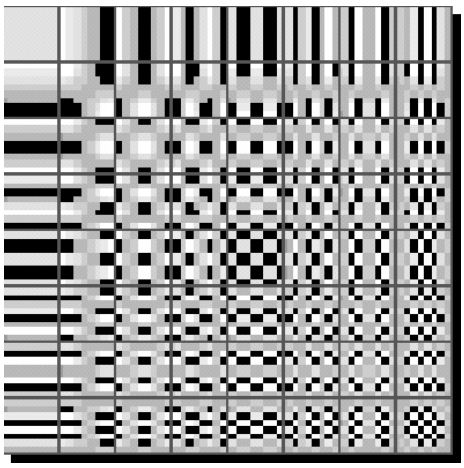


Figura 1.3 – Resultado de Transformada Discreta de Coseno Inversa.

1.2.2 Quantização

Seguindo o processo de transformação no domínio da freqüência, está a quantização dos coeficientes. Juntamente com a quantização ocorre a ponderação dos níveis do sinal. Nesse processo de ponderação, os coeficientes da DCT são divididos por constantes que

são função da freqüência bidimensional. Os coeficientes das baixas freqüências são divididos por pequenos números e os das altas freqüências são divididos por números maiores. Seguindo a divisão, o resultado é truncado até o inteiro mais próximo. Isto é ponderado dessa forma porque a percepção humana ao ruído é mais tolerada nas altas freqüências espaciais. Em outras palavras, no caso de ruídos de vídeo há efetivamente o mascaramento pelos detalhes mais finos de uma imagem como os contornos, enquanto nas áreas planas esse ruído se torna claramente visível.

A quantização dos coeficientes $G(u,v)$ pode ser implementada com a seguinte equação.

$$G_Q(u,v) = \text{arredondamento}(G(u,v) / Q(u,v)), \qquad (1.3)$$

em que u, v = freqüências espaciais;

$G_Q(u,v)$ = coeficientes DCT quantizados;

$G(u,v)$ = coeficientes DCT não quantizados;

$Q(u,v)$ = tamanho do passo de quantização.

O denominador, como foi dito, é geralmente diferente para cada freqüência espacial (u,v) para adaptar a faixa dinâmica do sistema visual humano. Esses valores são retirados de tabelas de percepção humana de luminância e de crominância. A Figura 1.4 mostra o resultado da quantização dos coeficientes DCT mostrados na Figura 1.2.

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figura 1.4 – Resultado de coeficientes DCT quantizados.

Observe-se que a repetição de zeros é uma clara redundância de dados, o que torna útil o resultado para compressão dos dados.

1.2.3 Codificação de Entropia

No vídeo real nem todas as freqüências espaciais estão presentes simultaneamente. Isso quer dizer que a matriz de coeficientes DCT deve conter termos que são zeros. Como

foi visto, a quantização desses coeficientes aumenta o número de zeros na eliminação de valores pequenos. O RLE, analisado a seguir, permite que esses coeficientes sejam manuseados de uma forma mais eficiente.

A codificação pelo comprimento - RLE é um método simples para compressão de dados sem perdas, e que tem a propriedade de funcionar melhor em dados com repetição de elementos. Um exemplo para o qual esta técnica poderia ser aplicável é a compressão de imagens monocromáticas consistindo somente de uns (preto) e zeros (branco). A técnica consiste em agrupar consecutivamente símbolos similares e designar uma soma e o símbolo para aquele grupo. Por exemplo, a seguinte seqüência de bits,

1111111111111111111111110000000000000000000001111111111111

pode ser agrupada como

$$\underbrace{1111111111111111111111111}_{1^{25}}\ \underbrace{0000000000000000000}_{0^{21}}\ \underbrace{1111111111111}_{1^{13}}$$

a qual requer somente a armazenagem ou transmissão da seguinte informação

$$\{1, 25, 0, 21, 1, 13\}. \qquad (1.4)$$

Podem ser salvos mais bits se o *codec* RLE possuir a premissa de que o primeiro número transmitido é sempre uma contagem do número de zeros binários seguido por uma contagem do número de uns binários e assim por diante. A seqüência em (1.4) seria reduzida, então, a

$$\{0, 25, 21, 13\}$$

Indicando, com o primeiro número sendo 0, não haver zeros para codificar.

O Código de Huffman, criado por D. A. Huffman em 1962 [3] é uma outra técnica sem perdas usada para reduzir a entropia de dados, codificando-os de tal maneira que o mais provável símbolo é representado pela palavra código mais curta. Pela definição de Shannon, a entropia de uma fonte é dada por

$$\eta = -\sum_i p_i \log_2 p_i \quad \text{bits /símbolo,}$$

(1.5)

em que p_i é a probabilidade de ocorrer o símbolo i. O valor de η para qualquer fonte é teoricamente o menor número médio de bits requeridos para representar os símbolo de fonte. Mostra-se que o comprimento médio das palavras código geradas pelo esquema proposto por Huffman satisfaz a inequação

$$\eta \leq \bar{L} \leq \eta + 1, \tag{1.6}$$

em que $\bar{L}$ é o comprimento médio de uma palavra código.

Devido às palavras código produzidas por um codificador por entropia serem de comprimento variável, então seria vantajoso que elas tivessem a propriedade de serem unicamente decodificáveis no receptor. A condição para isto é que as palavras-código devem satisfazer a regra do prefixo que estabelece que palavra código não pode ser prefixo de qualquer outra palavra código. As palavras código de Huffman possuem essa propriedade e são conhecidas como código de prefixo.

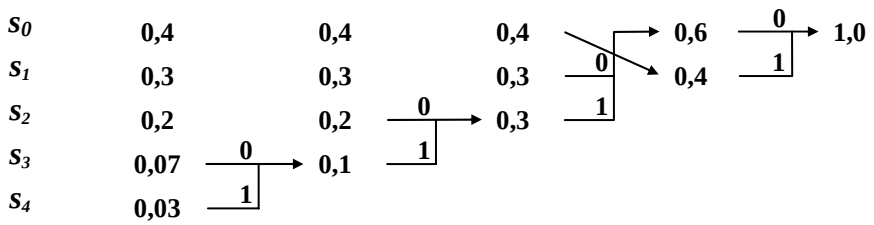
Poder-se-ia dizer que todos os códigos de prefixos são códigos unicamente decodificáveis, mas não vice-versa. A Tabela 1.1 ilustra a diferença. O código A é unicamente decodificável, porém não é um código de prefixo porque A_0 é um prefixo de A_1. Da mesma forma, A_1 é um prefixo de A_2, e assim por diante. O código B, por outro lado, é unicamente decodificável e um código de prefixo.

Tabela 1.1 – Exemplo de diferença de códigos prefixo.

Fonte Símbolo s_i	Probabilidade p_i	Código A A_i	Código B B_i
s_0	0,4	0	1
s_1	0,3	01	00
s_2	0,2	011	010
s_3	0,07	0111	0110
s_4	0,03	01111	0111

Com referência a Tabela 1.2, a construção de códigos de Huffman segue um algoritmo direto como mostrado a seguir:

Tabela 1.2 – Construção de palavras código de Huffman.



1. Obtém-se a probabilidade de ocorrer cada um dos símbolos de fonte, s_i;
2. Ordena-se em ordem crescente de probabilidade, com os símbolos de mais alta e de menor probabilidade no topo e no fundo da tabela, respectivamente;
3. Designa-se o binário 1 para o símbolo de menor probabilidade e o binário 0 para o próximo símbolo de maior probabilidade ;
4. Removem-se os dois símbolos de menor probabilidade da tabela somando-os para obter a probabilidade combinada;
5. Se a probabilidade combinada é 1, então para-se o procedimento. Se não, reinsire-se a probabilidade combinada na tabela e volta-se ao passo 2;
6. Para obter a palavra código de Huffman para um símbolo em particular, traça-se a seqüência de 0s e 1s binários do último estágio de volta para o símbolo em questão. Este é o código de Huffman para aquele símbolo.

Isto resulta nas palavras código de Huffman tal qual o Código B na Tabela 1.1. A partir das probabilidades dos símbolos de fonte, a entropia para esta fonte é

$$\eta_B = - 0{,}4.\log_2(0{,}4) - 0{,}3.\log_2(0{,}3) - 0{,}2.\log_2(0{,}2) - 0{,}07.\log_2(0{,}07) - 0{,}03.\log_2(0{,}03)$$

$$= 1{,}93 \text{ bits / símbolo.}$$

Quando η_B é comparado ao comprimento médio da palavra código de Huffman, obtém-se

$$\overline{L} = 0{,}4.(1) + 0{,}3.(2) + 0{,}2.(3) + 0{,}07.(4) + 0{,}03.(4)$$

$$\overline{L} = 2 \text{ bits / símbolo.}$$

Observe-se que valor de $\overline{L}$ (comprimento médio da palavra código) encontrado satisfaz a Inequação (1.6), sendo maior que η_B e menor que $\eta_B + 1$.

1.3 CODIFICAÇÃO DE FONTE TEMPORAL

A codificação de fonte temporal, também conhecida como codificação inter quadro, implementa-se em algoritmos codificadores de vídeo tais como **H.261** [4], **H.263** [5] e **MPEG-2** [6]. Como mencionado anteriormente, ela é indicada para a redução da quantidade de redundância temporal que está presente entre um quadro e o seguinte. Existem vários métodos que usam a codificação temporal, por exemplo, diferença de quadro, estimativa de movimento e predição compensada de movimento.

1.3.1 Diferença de quadro

Diferença de quadro é a subtração dos valores do quadro atual do anterior, com a codificação da diferença. Como normalmente há diferenças muito pequena entre dois quadros consecutivos, eles têm valores pequenos. A operação pode ser representada por

$$d_t = I_t - I_{t-1}, \tag{1.7}$$

em que I_t e I_{t-1} são respectivamente o quadro atual e o quadro anterior e d_t é a diferença de quadro. A codificação da diferença pode ser feita por meio de codificação por entropia. A palavra é então transmitida por meio do canal para o decodificador, que apresenta o processo inverso de maneira a recuperar a diferença. A diferença é então adicionada de volta a uma versão reconstruída do quadro para obter o quadro reconstruído atual, que é

$$\begin{aligned} \hat{I}_t &= \hat{I}_{t-1} + d_t \\ &= I_t + \hat{I}_{t-1} - I_{t-1} \\ &= I_t + e_t \end{aligned}$$

Em que $\hat{I}_t$ e $\hat{I}_{t-1}$ são o quadro reconstruído corrente e o quadro reconstruído anterior, respectivamente. O erro entre o quadro anterior reconstruído e o quadro original anterior, e_t, é zero se $\hat{I}_{t-1}$ e I_{t-1} são idênticos.

1.3.2 Compensação e estimativa de movimento

Compensação de movimento é a técnica de compensação para o movimento de pixels, grupos de pixels ou objetos de um quadro para o próximo. A estimativa de movimento tem que funcionar primeiro, de modo que a compensação de movimento trabalhe corretamente. Estimativa de movimento é o processo pela qual se obtém uma medida que diz ao codificador a direção na qual um pixel ou grupo de pixels têm-se movido. Isto é ilustrado na Figura 1.4. Essa medida é normalmente na forma de um vetor de movimento (MV – *Motion Vector*) e aplica-se a um grupo de pixels, ditos 16 × 16, que podem ser expressos como

$$e_t\,(x, y) \;=\; b_t(x, y) - b_{t-1}(x + x_1, y + y_1)\,, \qquad (1.8)$$

Em que $b_t\,(x, y)$ é o bloco no tempo t e $b_{t-1}(x + x_1, y + y_1)$ é o bloco no tempo $t - 1$ adicionado a um vetor (x_1, y_1). O erro entre os dois é $e_t\,(x, y)$.

Existem vários algoritmos que podem ser utilizados para calcular para os blocos o valor do vetor de movimento (x_1, y_1). Alguns deles são: método de pesquisa integral, pesquisa logarítmica 2-D, pesquisa hierárquica paralela 1-D, e outros métodos hierárquicos [7-10].

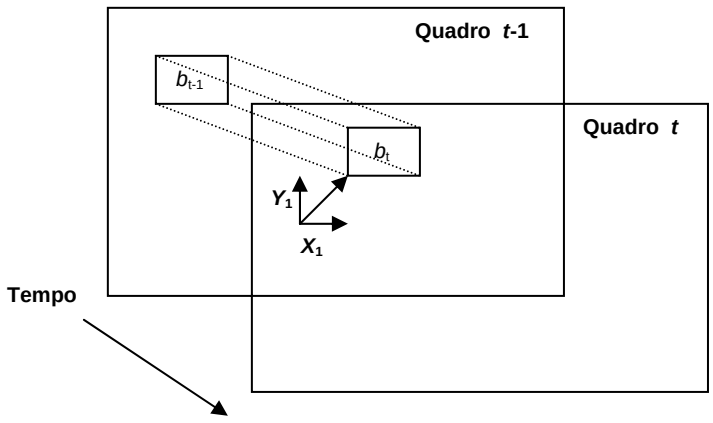


Figura 1.5 – Estimativa de movimento para codificação interquadro.

1.4 QUALIDADE DE CENA

Antes de finalizar este capítulo de codificação de fonte, precisam ser discutidas as técnicas usadas para medir a qualidade de cena da imagem resultante ou codificada. O primeiro método de avaliação é conhecido como Escore Médio de Opinião (MOS – *Mean Opinion Score*). O MOS é baseado na obtenção de opinião de um grupo de pessoas que julgam a qualidade subjetiva da imagem decodificada em relação à original. Essa avaliação é baseada em cinco diferentes níveis que o avaliador escolhe para as imagens. Os cinco níveis ou escores usados para determinar quanto uma imagem decodificada têm de distorção são os seguintes:

1. Muito ruim;
2. Ruim;
3. Levemente ruim;
4. Perceptível, mas não ruim;
5. Imperceptível.

Um outro quantitativo mais comum é o erro quadrático médio (MSE – *Mean Squared Error*). O MSE é uma medida de distorção entre a imagem original a e a imagem decodificada b, e é definido como

$$\text{MSE}\ (a,b) = \frac{1}{N \times M} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \left[a(x,y) - b(x,y) \right]^2 , \qquad (1.9)$$

em que N é o número de colunas e M o número de linhas; e (x, y) é a coordenada de pixel.

Contudo, o procedimento mais usado é o cálculo da Relação Sinal/Ruído de Pico (PSNR – *Peak Signal to Noise Ratio*). A PSNR usa o quadrado do valor de pico de pixel (255) para um pixel de 8 bits qualquer. A PSNR é medida em decibel (dB) e definida como

$$\text{PSNR}\,(a,b) = 10 \log_{10} \left[\frac{255^2}{\frac{1}{N \times M} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \left[a(x,y) - b(x,y) \right]^2} \right].$$

(1.10)

Finalmente, deve ser mencionado que a PSNR não é uma medida precisa de quão boa uma imagem parece quando comparada ao seu original. Existem casos em que por mais que uma imagem decodificada obtenha um resultado bom de PSNR, a qualidade subjetiva parece distorcida ao olho humano fornecendo um MOS baixo. Essa é a razão pela qual têm sido pesquisadas técnicas de medidas baseadas em entradas ponderadas pelos parâmetros de sistemas visuais humanos (HVS – *Human Visual System*). Visto que não há método padronizado de medida da qualidade de cena, será adotado a PSNR, que é amplamente aceita.

2. PADRÃO DE CODIFICAÇÃO DE VÍDEO NO H.263

O padrão utilizado no codificador objeto desta monografia é o H.263, portanto, este capítulo descreve as partes principais que interessam para o estudo.

Na última década, tem havido um significativo interesse em aplicações de vídeo digital. Conseqüentemente, têm emergido vários padrões que tornam possíveis aplicações tais como videotelefonia, videoconferência, DVD e TV Digital. Estes padrões fornecem os meios necessários para conseguir a interoperabilidade entre sistemas projetados por diferentes fabricantes, facilitando, inclusive, o crescimento do mercado. O ITU-T1 é uma das duas organizações formais que produz padrões para vídeo codificação – a outra é a ISO/IEC JTC12. Os padrões de vídeo codificação são chamados recomendações, e eles são designados como H.26x (isto é, H.261, H.262 e H.263). Os padrões ISO/IEC são designados como MPEG-x (ou seja, MPEG-1, MPEG-2 e MPEG-4).

De ordem a responder a então emergente necessidade de aplicações de vídeo telefonia e conferência, o ITU-T iniciou a elaboração do padrão H.261 em 1984, que foi especificado para operar em taxas múltiplas de 64 kbps. É fundamentada em um sistema de compensação de movimento e transformada discreta do coseno e forma a base de todos os padrões subseqüentes de codificação de vídeo. A versão final foi aprovada em 1990. E pela necessidade de vídeo comunicação a baixas taxas de transmissão (menor que 64 kbps), o ITU-T iniciou os trabalhos na elaboração do padrão H.263 em 1993 e aprovada a versão 1 em 1995.

Foram propostas melhorias à Versão 1 do padrão H.263 que culminaram com a elaboração da versão 2 (H.263+), aprovada em 1998. A Versão 3 (H.263++) introduzida a seguir, com o objetivo de aumentar a eficiência de codificação em aplicações de comunicações de vídeo sem fio e fontes de vídeo entrelaçadas, foi finalizada em 2001.

2.1 O CODIFICADOR DE FONTE H.263

O objetivo do codificador de fonte H.263 é fornecer uma representação compactada da cena removendo redundâncias espaciais que existem dentro do quadro, e as redundâncias temporais que existem entre quadros sucessivos. O padrão H.263 baseia-

se no uso da DCT em conjunto com a quantização para remover as redundâncias espaciais (costuma-se dizer que se remove informação irrelevante e redundante) e na estimativa e compensação de movimento para remover redundâncias temporais. Quando uma fonte é codificada usando a DCT, o codificador é dito estar operando no modo de *intra*codificação, e o quadro codificado correspondente é chamado de I-*picture*. No caso em que se usa o prognóstico temporal, o codificador é dito estar operando no modo de intercodificação, e o quadro codificado correspondente é chamado de P-*picture*. A Figura 2.1 mostra o codificador H.263 em diagrama em blocos operando nesses modos. As funções do codificador, de seus componentes e a sintaxe do trem de pulsos são descritas nas seções subseqüentes.

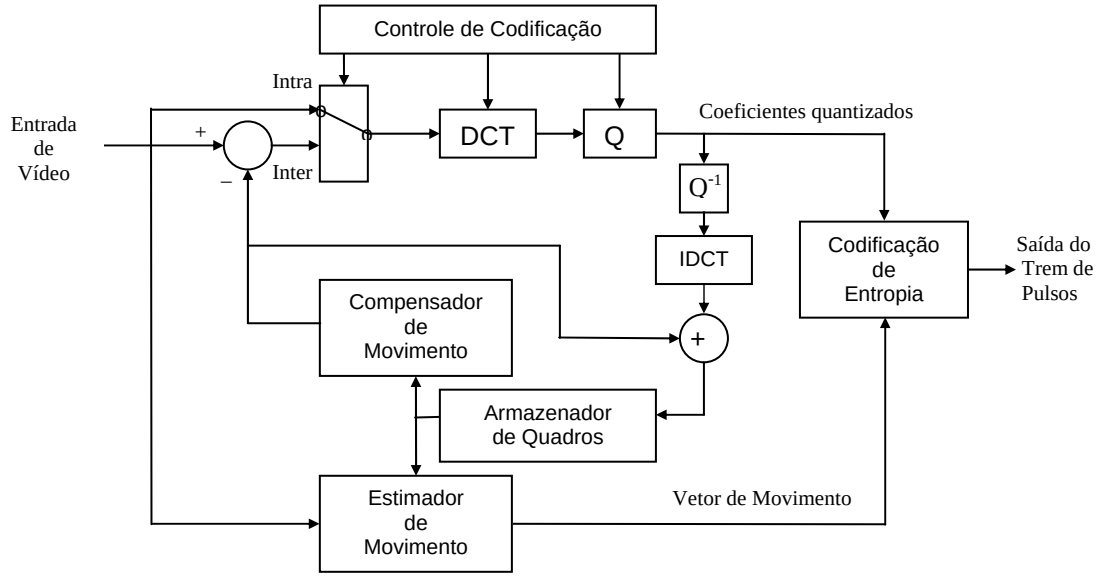


Figura 2.1 – Codificador de Fonte H.263.

2.1.1 Formatos do quadro de vídeo (cena)

Todas as versões do codificador de fonte H.263 possuem suporte para os formatos de vídeo: 128 × 96, 176 × 144, 352 × 288, 704 × 576 e 1408 × 1152 pixels. O tamanho central 352 × 288 pixels é conhecido como formato intermediário comum (CIF - *Common Intermediate Format*). Dividindo-o por dois, tanto a dimensão horizontal como a vertical, obtém-se *quarter* CIF (QCIF). O menor dos formatos é conhecido como sub-QCIF. Indo em outra direção, ou seja, dobrando-se as dimensões obtém-se o 4CIF e o 16CIF. Estes formatos estão relacionados na Tabela 2.1. Observe que existem

dois grupos de linhas e pixels, um para os componentes de luminância (Y) e o outro para os componentes de crominância (C). O formato usado nesta dissertação será o QCIF, que melhor se adapta a telas de videofone móveis.

Tabela 2.1 – Formatos de cena do codificador H.263.

Formato de Cena	Pixels (Y)	Linhas (Y)	Pixels (C)	Linhas (C)
Sub-QCIF	128	96	64	48
QCIF	176	144	88	72
CIF	352	288	176	144
4CIF	704	576	352	288
16CIF	1408	1152	704	576

2.1.2 Organização das camadas do quadro de vídeo

A componente de luminância da cena é amostrada nas resoluções de quadro vistas na subseção anterior, enquanto que as componentes de crominância, Cb e Cr, são subamostradas dividindo a resolução por dois, tanto na direção horizontal como na vertical. A seqüência de vídeo é composta de inúmeras cenas que são divididas em grupos de cenas. Cada cena é dividida em fatias ou tiras. Cada tira é dividida em macroblocos, consistindo de quatro blocos de luminância de 8 pixels x 8 linhas seguido por um bloco Cb e um bloco Cr, cada um consistindo também de 8 pixels x 8 linhas. Por exemplo, um quadro com resolução QCIF (176x144) é dividido em 99 macroblocos (de 16x16) como mostrado na Figura 2.2 a seguir. Em outras resoluções são padronizadas segmentações de macroblocos similares a essa.

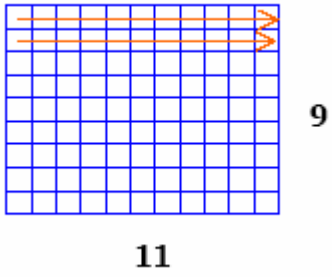


Figura 2.2 – Subdivisão de um quadro QCIF em 99 macroblocos de 16x16.

2.1.3 Estimador e compensador de movimento

É freqüente o caso em que quadros sucessivos ou próximos no tempo tenham imagens similares. Conseqüentemente, é útil para a redução da largura de banda fazer uso, tanto quanto possível, da informação apresentada em um quadro codificado previamente. Ou seja, realiza-se a subtração entre o quadro anterior e o quadro corrente, e somente a diferença ou resíduo é codificada e transmitida. Isto significa que áreas que não mudam do quadro (por exemplo, o fundo) não precisam ser codificadas novamente. Neste caso, o quadro anterior é usado como uma estimativa do quadro atual.

Uma maneira mais sofisticada de aumentar a eficiência da codificação é trabalhar em nível de macrobloco no quadro, em vez de processar o quadro todo de uma vez como descrito acima. O processo é chamado de estimativa de movimento, ou mais precisamente, prognóstico compensado de movimento, e baseia-se no fato de que a maioria do movimento que o macrobloco sofre é um movimento de translação. O movimento apresentado por um macrobloco é representado por um vetor movimento que tem dois componentes, o primeiro indicando o deslocamento horizontal, e o segundo indicando o deslocamento vertical.

2.1.4 Transformada Discreta do Coseno (DCT)

O DCT 8x8 transforma um grupo de valores de pixels (ou valores residuais) em um conjunto de coeficientes de freqüência espaciais. Isto é análogo à transformação de um sinal no domínio do tempo em um sinal no domínio da freqüência usando a Transformada Rápida de Fourier. Ou em outras palavras a DCT é usada para facilitar a identificação de componentes irrelevantes do sinal, para o processo de quantização. A DCT em conjunto com a quantização opera sobre um bloco de pixels bidimensional compactando os valores em um pequeno número de coeficientes. Isto significa que somente uns poucos coeficientes são necessários para recriar uma cópia do bloco de pixels original.

2.1.5 Quantização

A quantização é uma importante técnica para a compressão de dados. O módulo quantizador reduz a precisão de cada coeficiente ajustando aqueles próximos de zero

para zero. Portanto, somente permanecem uns poucos coeficientes significativos diferentes de zero. Isto só é possível devido à baixa sensibilidade do olho humano para reconstruir diferenças relacionadas a sinais de alta freqüência espacial. Isto significa que mudanças em componentes de alta freqüência de imagens não são percebidas e, conseqüentemente, podem ser descartadas, o que já não acontece com mudanças lentas em intensidade ou cor. Então, a idéia básica da quantização é eliminar a maioria dos coeficientes DCT correspondentes a componentes de alta freqüência.

2.1.6 Codificação por entropia

Um codificador por entropia (tal como o codificador de Huffman) codifica símbolos de alta probabilidade de ocorrência com códigos binários curtos e códigos binários longos são usados para símbolos de baixa probabilidade de ocorrência. A codificação por entropia no H.263 baseia-se nessa técnica e é usada para comprimir os coeficientes DCT quantizados. O resultado é uma seqüência de códigos binários de comprimento variável.

2.1.7 Armazenador de quadros

O quadro corrente deve ser armazenado de modo que ele possa ser usado como referência quando o próximo quadro for codificado. Em vez de simplesmente copiar o quadro atual diretamente no armazenador, os coeficientes quantizados são re-escalados usando a DCT inversa e adicionados ao vetor de referência do compensador de movimento. Isto garante que no próximo quadro codificado o codificador de movimento use o conteúdo deste quadro armazenado para determinar a melhor área para compensação de movimento.

2.1.8 Controle de codificação

O bloco controle de codificação é usado para chavear, em nível de macrobloco, os modos de intracodificação e inter codificação. O critério usado para decidir o modo de codificação é um algoritmo chamado de Soma de Diferenças Absolutas (SAD - *Sum of Absolute Differences*). Se o macrobloco não muda significativamente com relação à cena de referência, o codificador pode então saltar aquele macrobloco e o decodificador simplesmente repetirá o macrobloco locado na cena de referência.

2.2 SINTAXE DO TREM DE PULSOS H.263

A Figura 2.3 ilustra como se particiona o fluxo de dados H.263 em várias camadas. Cada camada será explanada nas subseções a seguir.

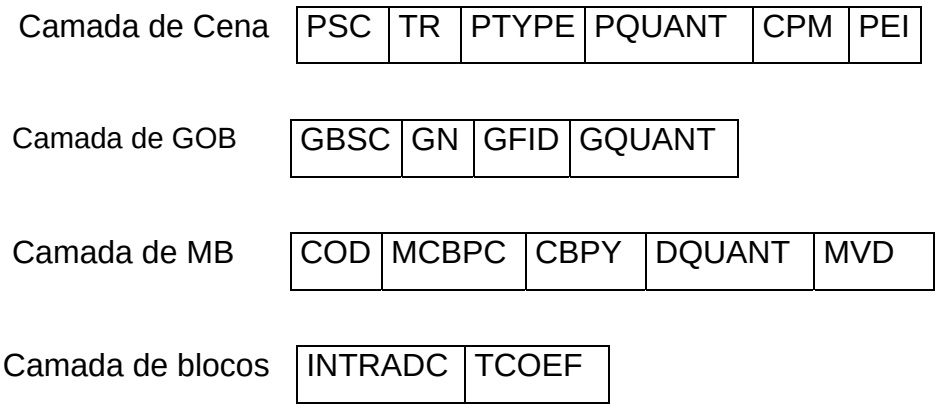


Figura 2.3 – Sintaxe do trem de pulsos H.263.

2.2.1 Camada de cena

Está relacionada aos quadros da seqüência de imagens. Contém o código de início de cena (PSC – *Picture Start Code*), tem um comprimento de 22 bits e serve para identificar o início de cada quadro. A referência temporal (TR – *Temporal Reference*) de 8-bits forma-se incrementando por um seu valor no cabeçalho de cena previamente transmitido mais o número de cenas não transmitidas. O PTYPE (*Picture Type Information*) de 13-bits contém informações a respeito da cena (indicador de separação de cena, indicador de dados de câmera, liberação de imagem congelada, formato de fonte, tipo de codificação de cena) e o uso dos modos de codificação opcionais. Existem quatro modos de codificação opcionais

1. Vetor de movimento irrestrito;
2. Codificação aritmética baseada em sintaxe;
3. Previsão avançada;
4. PB-quadros.

Uma palavra código de comprimento fixo (PQUANT) de 5 bits indica o quantizador a ser utilizado na cena, a menos que atualizado por qualquer GQUANT ou DQUANT

subseqüente. O valor PQUANT representa a metade do passo do quantizador a ser utilizado. A palavra código CPM (*Continuous Presence Multipoint*) de 1-bit sinaliza o uso do modo de presença multiponto opcional. A informação de inserção extra de bit (PEI – *Picture Extra Insertion*) é mudada para 1 quando existe um campo de dados nos bits informação de espera (PSPARE).

2.2.2 Camada de Grupo de Blocos

A camada seguinte é a de grupo de blocos (GOBs). Cada quadro é dividido em $k \times 16$ linhas, conhecidas como um GOB. O valor de k depende do formato e é 1 para sub-QCIF, QCIF e CIF, 2 para 4CIF e 4 para 16CIF. Conseqüentemente, cada cena em QCIF tem 144/16 = 9 GOBs.

Esta camada consiste de um cabeçalho seguido de dados para os macroblocos. Uma exceção é o primeiro GOB de cada cena (com número 0) que não deve ser acompanhado de cabeçalho. O código de início de grupo de blocos GBSC (*Group of Block Start Code*) é uma palavra código de 17-bits e pode ser *byte*-alinhada. O número de GOBs é indicado por uma palavra código GN (*Group Number*) de 5-bits. O ID do quadro GOB (GFID) é uma palavra código de 2-bits tendo o mesmo valor em todos os cabeçalhos de uma dada cena. O quantizador usado nas partes restantes da cena pode ser modificado ajustando-se a palavra código GQUANT de 5-bits até que seja atualizada por qualquer GQUANT ou DQUANT subseqüente. Como no caso de PQUANT, o valor de GQUANT representa metade do tamanho do quantizador a ser usado.

2.2.3 Camada de Macroblocos

Cada GOB é dividido em macroblocos (MB). Um macrobloco compreende um bloco de luminância (Y) de 16×16 pixels e dois blocos de crominância (C_B e C_R) espacial correspondentes de 8×8 pixels. Conseqüentemente, consiste de quatro blocos Y, um bloco C_B e um C_R na ordem mostrada na Figura 2.4.

Os dados para cada macrobloco consistem de um cabeçalho seguido de dados para os blocos. Se o bit de indicação de macroblocos codificados (COD) é colocado em ‘0’, o macrobloco é codificado. Se estiver em ‘1’, não há informação a ser transmitida para este macrobloco. O MCBPC é um codificador de palavra de comprimento variável que

informa o tipo de macrobloco e o bloco codificado padrão para crominância. As palavras código variáveis são pré-definidas em tabelas para quadros I e P. Similarmente, CBPY é uma palavra código variável para luminância.

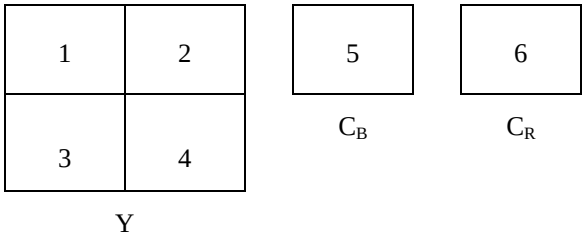


Figura 2.4 – Arranjo de blocos dentro de um macrobloco.

A informação do quantizador pode ser atualizada por uma palavra código DQUANT de 2-bits que define uma mudança no valor do quantizador. O dado do vetor de movimento (MVD) é codificado em comprimento variável, além de estar presente em todos os inter macroblocos.

No modo interquadro, os macroblocos podem ser intra ou intercodificados dependendo da figura de mérito conhecida como a soma das diferenças absolutas [13], que mede se houve muitas mudanças entre o macrobloco corrente e o anterior.

2.2.4 Camada de Blocos

É nesta camada que se aplica a DCT. Depois de aplicar em um bloco, usa-se uma palavra código (INTRADC) para codificar o coeficiente DC dos intrablocos. Os outros coeficientes transformados são codificados usando palavras código de comprimento variável (TCOEF) representando um EVENT. Um EVENT é uma combinação do último coeficiente não zero (LAST), o número de zeros sucessivos precedendo o coeficiente codificado (RUN) e o valor não zero do coeficiente codificado (LEVEL) ao longo de um caminho em zig-zag conforme a Figura 2.5.

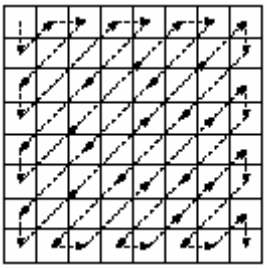


Figura 2.5 – Caminho em zig-zag na transformação dos coeficientes.

1.4 ERRO RECUPERÁVEL ADICIONADO AO PADRÃO DE CODIFICAÇÃO DE VÍDEO

Esta seção descreve brevemente os recursos que foram acrescentados aos padrões de codificação de vídeo (H.261 e H.263) para combater erros de canal. Essas adições vão desde codificação de canal e sincronismo nos algoritmos de codificação de vídeo a modificações nos níveis de transporte e multiplexadores.

1.2.1 H.261

O padrão H.261 foi projetado para transmissão de p x 64 kbps por meio de canais tais como linhas ISDN. As taxas de erro em tais linhas são da ordem de 10^{-6}. O H.261 tem um código de correção de erro (código FEC – *Forward Error Correction*) baseado no Código BCH (511, 493). Este código é bem simples e protege 493 bits de fonte com 511 – 493 = 18 bits de paridade. Esses 18 bits de paridade detectarão até dois erros dentro dos 493 bits de fonte. Este código FEC pode ser apropriado para canais com fio, mas limitado quando aplicado a canais com desvanecimento propensos a erros.

1.2.2 H263 Versão 2

Existem vários anexos dentro da norma H.263 que fornecem algum tipo de medida de proteção de erro e erro recuperável. O anexo H da norma H.263 também contém o mesmo código BCH como descrito antes. Em 1998, o H.263 foi revisado para a versão

2 [14], que ficou formalmente conhecida como H.263+. A versão 9 inclui mais nove modos, tais como intra codificação avançada (AIC), filtro de desbloqueio (DF) e quantização modificada (MQ). Recursos adicionais incluem também, entre outros, a habilidade de congelar e liberar a cena de vídeo. Estas novas características são documentadas nos anexos I a T. Aquelas que têm impacto na recuperação de erros do H.263 são descritas brevemente a seguir.

1.4.2.1 Anexo K – Modo Estruturado de Faixas (ou Tiras)

O Modo Estruturado de Faixas permite um agrupamento funcional de um número de macroblocos na cena, melhorando a recuperação dos erros, aumento de transporte sobre redes de pacotes, e reduzido atraso. Na seção 2.2.2, foi discutido o uso dos cabeçalhos GOB para manter a sincronização. Contudo, isto é limitado quando os cabeçalhos GOB

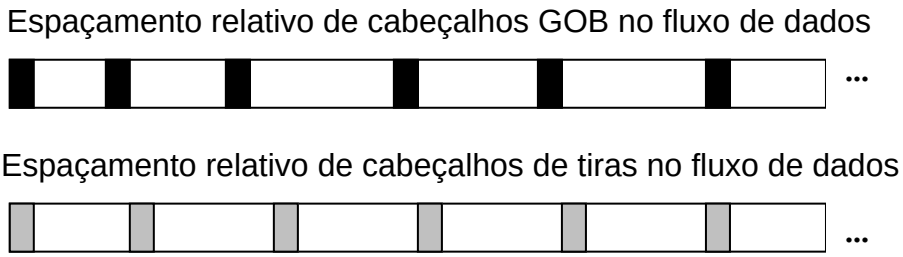


Figura 2.6 – Ilustrações de diferenças posicionais entre usar os cabeçalhos GOB do H.263 padrão e o H.263 Versão 2, anexo K (modo estruturado de faixas) para melhorar o erro recuperável.

não aparecem em intervalos variáveis dentro do trem de pulsos do H.263. Com a implementação do modo estruturado de faixas, os cabeçalhos das faixas são regularmente interespaçados ao longo dos trens de pulsos, como mostrado na Figura 2.6. Este espaçamento regular auxilia a recuperação de erros com a maior facilidade de sincronização, bem como limita os erros a uns poucos macroblocos em vez de todo GOB.

1.4.2.2 Anexo N – Modo de Seleção de Cena de Referência

Este modo aumenta a recuperação de erros, permitindo selecionar uma cena de referência anterior, que não é a cena decodificada mais recente, que pode ser sintaticamente referenciada. Isto funciona quando se detecta um erro no decodificador,

um canal adjacente anterior é utilizado para sinalizar o codificador a mudar o quadro de referência usado para prever o próximo quadro codificado. Esse método atua rapidamente para cancelar erros que de outra maneira se propagariam até o próximo quadro intracodificado. Complexidade adicional de sinalização e aumento da memória usada no decodificador são as modificações requeridas de maneira a que este modo funcione.

1.4.2.3 Anexo O – Modo de Escalabilidade

Dá suporte à escalabilidade temporal, SNR e espacial, e pode ser usado em conjunto com esquemas de controle de erro. Esse modo fornece tipos de cenas adicionais que são usadas para gerar camadas a mais, suplementares às camadas de base. A idéia é fornecer um nível de proteção maior às camadas de base do que as camadas adicionais.

1.4.2.4 Anexo R – Modo Independente de Decodificação de Segmento

Isto aumenta a recuperação de erro, assegurando que dados corrompidos de alguma região da cena não possam causar a propagação de erro para outras regiões. Com este modo os vetores de movimento não apontam para macroblocos pertencentes a outros segmentos. Isto ajuda a localizar qualquer erro causado por vetores de movimento corrompidos no corrente segmento.

3. CODIFICAÇÃO DE CANAL

Embora esta dissertação esteja centrada em apresentar uma modificação no sistema de sincronismo da codificação de vídeo para melhorar a qualidade de apresentação da seqüência de cenas, assunto que será tratado no Capítulo 4, o sinal precisa ser transmitido e, portanto, torna-se vulnerável a perturbações no caminho da transmissão ou canal. O ruído que inevitavelmente está presente em um canal pode causar a transmissão de informação corrompida. No Capítulo 1 a discussão centrou-se na remoção da redundância de fonte pela codificação de fonte ou compressão. Uma vez que os dados tenham sido comprimidos, tornam-se suscetíveis a ruído. A codificação de canal, também conhecida como codificação de controle de erro ou correção de erro no sentido da transmissão (FEC – *Forward Error Correction*), é um método no qual se insere uma redundância controlada nos dados, permitindo ao receptor detectar e/ou corrigir um número determinado de erros introduzidos pelo canal, como mostrado na Figura 3.1.

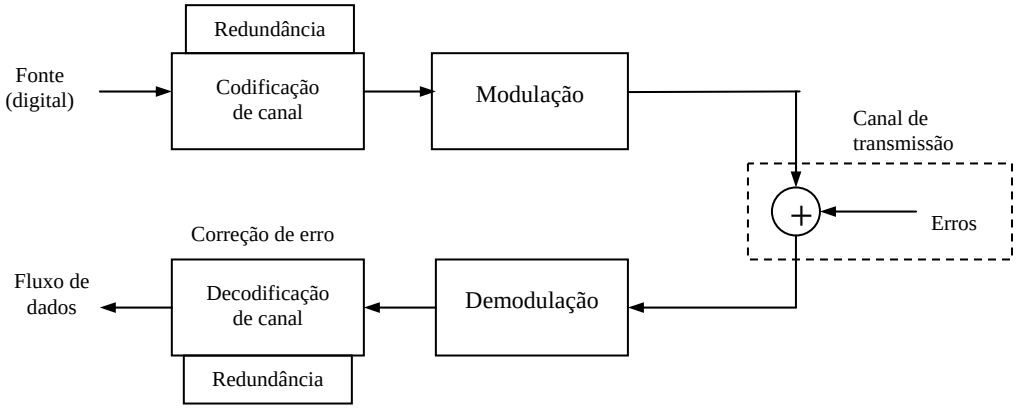


Figura 3.1 – Transmissão com codificação de canal e proteção de erro.

Em geral a adição de redundância implica que a largura de banda requerida para transmissão tem que ser aumentada, crescendo também com isso a complexidade do sistema de comunicação. Se a largura de banda não pode ser aumentada devido a limitações físicas, então algumas vezes isto implica que a taxa de fonte tem que ser diminuída. Existe ainda a questão da comercialização que tem que ser considerada

quando do projeto ou escolha de um sistema de codificação FEC. O segundo teorema de Shannon [1], também conhecido como teorema da codificação de canal na presença de ruído, prova que para todo canal com capacidade C existem códigos FEC tais que a informação pode ser transmitida por meio do canal em taxas menores que C com uma taxa de erro arbitrariamente baixa. A prova deste teorema é encontrada em [1, 15].

Embora existam muitos tipos de códigos FEC disponíveis para uso em um sistema de comunicação, eles podem ser divididos em duas principais categorias – códigos de blocos e códigos convolucionais. Códigos de blocos são formados tomando uma palavra de comprimento *k* bits e gerando uma palavra código de comprimento *n* bits, com $n > k$. A diferença gerada ($n - k$) é relacionada aos *k* bits originais por uma fórmula algébrica. Códigos convolucionais, por outro lado, são códigos que são produzidos quando os bits da informação são convoluídos com a resposta ao impulso de um codificador. Essa operação de convolução implica que cada bit de saída codificado é dependente do comprimento da resposta impulsiva do codificador [29].

Este capítulo introduz algumas das técnicas de codificação de canal disponíveis. Dentre elas as mais utilizadas: Código RS (*Reed-Solomon*), de bloco cíclico, e Código RCPC (*Rate Compatible Punctured Convolutional*), convolucional.

3.1 Código Linear de Bloco

Código linear de bloco é um dos códigos de controle de erro mais amplamente utilizados atualmente devido a sua facilidade de implementação. O termo linear significa que uma combinação linear de duas palavras código resulta em uma terceira palavra código. O método utilizado para codificar os blocos de informação é um processo de mapeamento no qual os bits são concatenados um a um gerando *n-k* conhecidos como bits de paridade e transmitidos com os *k* bits da mensagem original. Esta concatenação dos bits de paridade com a mensagem original é conhecida como codificação sistemática. A Figura 3.2 ilustra o processo sistemático de codificação de bloco.

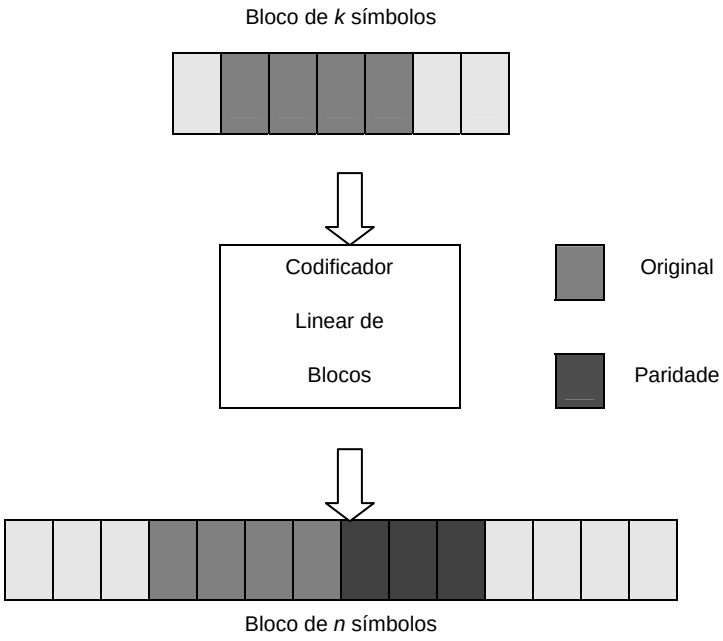


Figura 3.2 – Codificação de bloco sistemática *k/n*, em que *k* = 4 e *n* = 7.

Um exemplo de código linear de blocos é o Código de Hamming, útil para aplicações de correção de um erro por palavra, ou seja, pode-se ter problemas no uso em canais em que erros múltiplos ocorram na mesma palavra código.

3.2 Códigos Cíclicos

Existe uma outra classe de código de blocos conhecida como códigos cíclicos. O termo cíclico provém do fato de que um deslocamento cíclico em uma palavra código resulta em uma palavra código. Por exemplo, seja $\mathbf{c}_1 = (c_0, c_1, ..., c_{n-1})$, então uma palavra código é obtida fazendo um deslocamento cíclico para a direita, por exemplo, $\mathbf{c}_2 = (c_{n-1}, c_0, ..., c_{n-2})$. A segunda propriedade que esse código possui é a da linearidade, em que a soma de duas palavras código resulta em uma palavra código. Por causa dessas duas propriedades, eles têm as seguintes vantagens em relação aos outros tipos de códigos:

- A aplicação computacional pode ser facilmente implementada via registradores de deslocamentos rápidos;

- Eles têm uma estrutura algébrica para a qual podem ser encontrados métodos práticos de decodificação.

3.2.1 Códigos BCH

Códigos BCH (*Bose-Chaudhuri-Hocquenghem*) [17 – 19] formam uma classe de códigos cíclicos de correção de erros aleatórios. Eles derivam de uma generalização de código de Hamming e são aplicados na correção de erros múltiplos. A forma não binária de códigos BCH forma os códigos *Reed-Solomon* [16].

3.2.2 Códigos *Reed-Solomon*

Códigos *Reed-Solomon* (RS) são uma importante subclasse de códigos BCH usados para combater erros em surto que é uma das razões pelos quais eles são usados em aplicações de CD-ROM. A vantagem entre códigos binários BCH e códigos RS é que códigos RS protegem melhor sobre um alfabeto com 2^m símbolos. Os recursos computacionais usados para códigos RS são os mesmos de códigos BCH. Assim, um codificador RS (n, k) expande um bloco de k símbolos para um bloco de n símbolos pela introdução de símbolos redundantes $n - k$. Os parâmetros para um código RS corretor de t erros são dados na Tabela 3.1.

Tabela 3.1 – Propriedades de Códigos RS.

Comprimento do bloco	:	$n = 2^m - 1$ símbolos
Comprimento da mensagem	:	k símbolos
Dígitos de verificação de paridade	:	$n - k = 2t$ símbolos
Distância mínima	:	$d_{min} = 2t + 1$ símbolos
Capacidade de correção de erro	:	$t = \left\lfloor \frac{n-k}{2} \right\rfloor$ símbolos

Os códigos RS fornecem uma larga faixa de palavras código que podem ser usadas para diferentes comprimentos de mensagens. Devido à relação entre códigos RS e BCH, a facilidade de decodificação de códigos BCH bem como os algoritmos de decodificação associados podem ser estendidas aos códigos RS.

3.3 Códigos Convolucionais

Códigos convolucionais, introduzidos pela primeira vez por Elias [29], usam um registrador de deslocamento no processo de codificação. Não há separação dentro de predeterminados segmentos do trem de pulsos, a informação é espalhada sobre várias saídas de dados e multiplexadas numa única linha de transmissão de dados. Usando um codificador convolucional relativamente simples, como mostrado na Figura 3.3, para cada bit entrante no registrador de deslocamento dois bits são gerados como símbolos de saída. O número de linhas de entrada é $m = 1$ bit, o número de linhas de saída é $n = 2$ bits, e a taxa de código é $R = m/n = ½$.

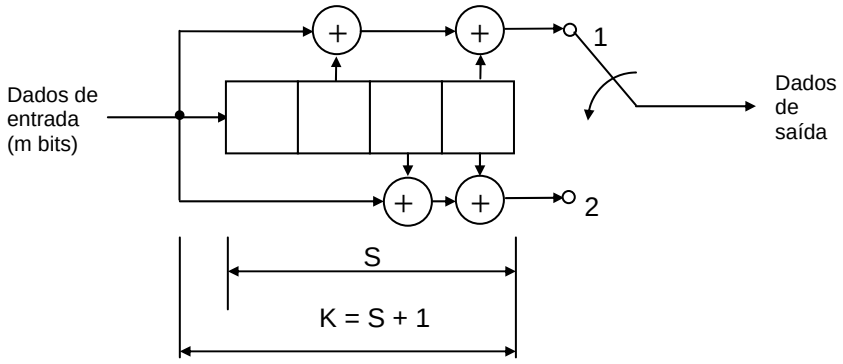


Figura 3.3 – Modelo de codificador convolucional para explicar os termos básicos.

A memória, isto é, a profundidade de armazenamento do codificador, é definida como o número de bits precedentes que contribuem para a codificação do bit corrente. No exemplo $S \cdot m = 4$, com S sendo igual ao comprimento do registrador de deslocamento. O comprimento de restrição K descreve o número total de bits que contribuem para o processo de codificação. Neste exemplo $K = (S + 1) \cdot m = 5$. O codificador convolucional tem $2^{S \cdot m} = 16$ possíveis estados e, finalmente, é caracterizado pelo número e posições de derivações no registrador de deslocamento, os quais são indicados pelos coeficientes do gerador polinomial g. Os parâmetros do codificador convolucional neste exemplo podem ser resumidos como segue:

Número de linhas de entrada $m = 1$

Número de linhas de saída $n = 2$

Taxa do código (m/n)	R	= 1/2
Memória	$S \cdot m$	= 4
Estados possíveis	$2^{S \cdot m}$	= 16
Comprimento de restrição	K	= 5
Gerador polinomial 1	g_1	$= 1 + D^2 + D^4$
Gerador polinomial 2	g_2	$= 1 + D^3 + D^4$

3.3.1 Códigos RCPC

Uma das propriedades de códigos convolucionais é que códigos de taxas baixas podem ser puncionados (perfurados) para produzir códigos de taxas maiores [20]. Isto é atrativo para aplicações nas quais se requer a geração de taxas variáveis de código de canal.

A punção (perfuração) é definida como o ato de apagar um ou mais bits da saída do codificador. O apagamento de bits resulta no aumento da taxa de codificação, que por sua vez torna menor o nível de proteção de canal disponível. Por causa da taxa de código ajustável, estes códigos são conhecidos como *Rate Compatible Punctured Convolutional Codes* (RCPC) [21]. Como um exemplo, considere uma palavra código **c** = (11, 01, 11, 10, 01) de taxa igual a ½. A taxa original ½ é conhecida como a taxa de código mãe. Se em cada quatro bits o terceiro bit for apagado, a saída resultante vem a ser

$$\mathbf{c} = (11, 0E, 11, 1E, 01,...), \tag{3.1}$$

em que *E* é um símbolo que foi apagado. Os bits que foram apagados não são transmitidos por meio do canal. Examinando (3.1) pode-se ver que a cada dois bits de entrada resulta em três bits de saída, e que significa que o código puncionado tem uma nova taxa igual a 2/3.

Além da taxa variável, uma outra vantagem é que um decodificador projetado para decodificar o código mãe pode também ser usado para decodificar códigos de taxas maiores, desde que os bits dentro dos códigos de taxas maiores estejam embutidos no

código gerado a partir do código mãe. Estes códigos RCPC também oferecem níveis de proteção de erro comparáveis aos códigos convolucionais fixos em taxas similares [22]. Os códigos RCPC são então úteis para codificação fonte-canal combinada, em que os níveis de proteção de erro dos bits da fonte são puncionados pela significância fornecida pelo codificador de fonte. Este assunto será tratado detalhadamente no capítulo 4.

Na implementação a seguir, um código mãe de taxa ¼ [23] é usado para proteger os bits da fonte na menor taxa. Os bits de saída são então puncionados baseados no valor de significância da fonte. Estes bits puncionados não são transmitidos pelo canal. Os geradores polinomiais usados para produzir os códigos e os valores puncionados são mostrados na Tabela 3.2 e o diagrama de estado do codificador na Figura 3.4. A implementação correspondente é mostrada na Figura 3.5, na qual cada 0 na matriz de punção representa um bit que é puncionado.

Tabela 3.2 – Gerador Polinomial e as matrizes de puncionamento para códigos RCPC com código mãe à taxa de ¼.

Gerador Polinomial	Matriz de Puncionamento		
	Taxa 1/3	Taxa 1/2	Taxa 2/3
$g^{(0)} = 1 + D^3 + D^4$	11111111	11111111	11111111
$g^{(1)} = 1 + D + D^2 + D^4$	11111111	11111111	10101010
$g^{(2)} = 1 + D^2 + D^3 + D^4$	11111111	00000000	00000000
$g^{(3)} = 1 + D + D^3 + D^4$	00000000	00000000	00000000

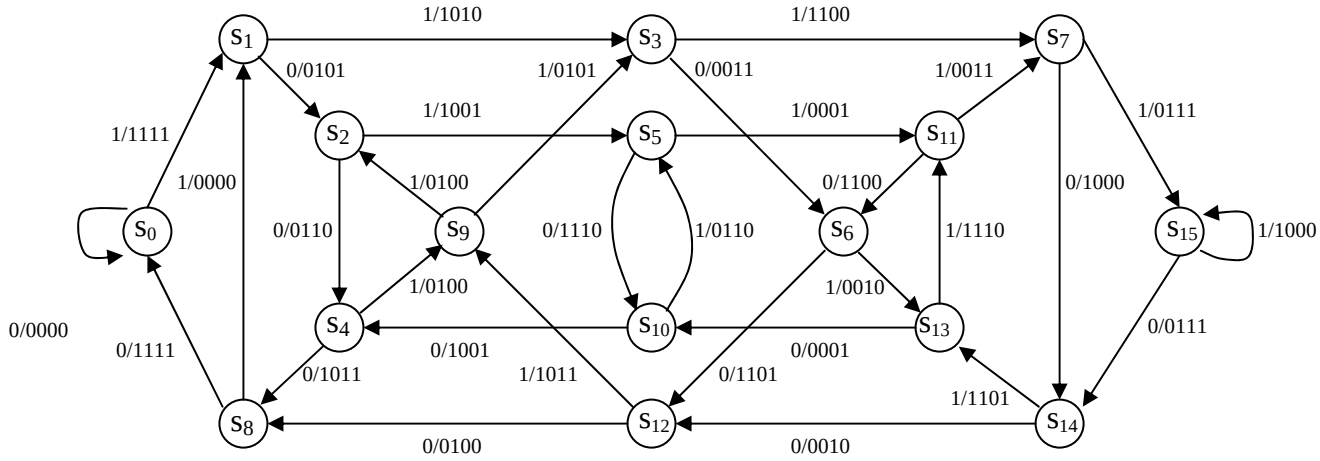


Figura 3.4 – Diagrama de estado do código RCPC dado na Tabela 3.2

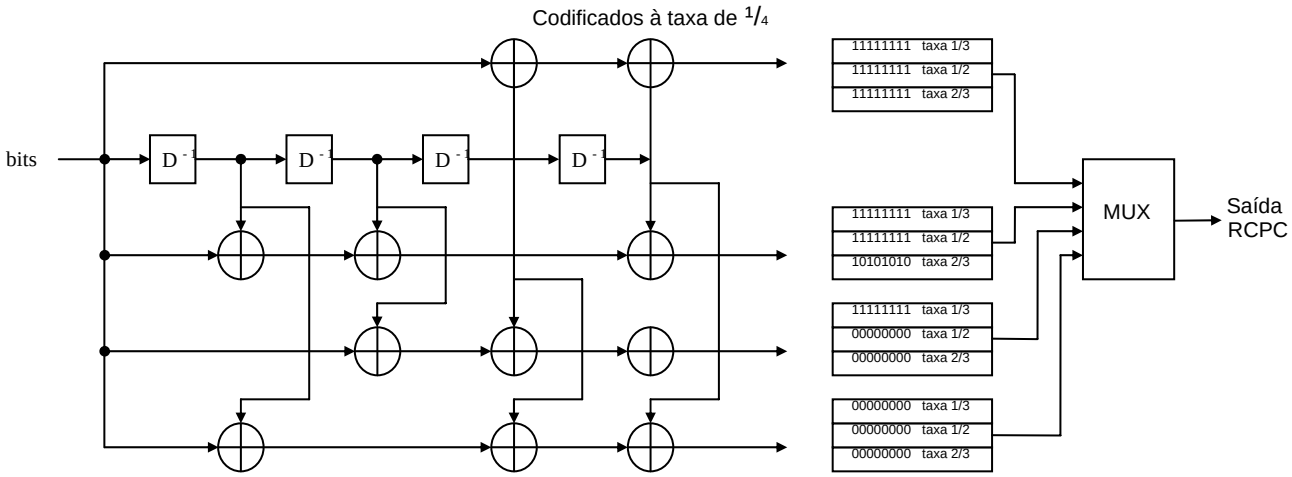


Figura 3.5 – Codificador RCPC com código mãe à taxa de ¼.

Esse código de taxa ¼ pode ser puncionado para obter os seguintes códigos de taxas mais elevadas: 4/15, 2/7, 4/13, 1/3, 4/11, 2/5, 4/9, ½, , 4/7, 2/3, 4/5 e 8/9. A distância livre (d_{free}) correspondente, associado com as taxas ponderadas assume os valores 13, 13, 11, 11, 9, 8, 7, 7, 5, 4, 3 e 2 respectivamente. Das matrizes de ponderação, observa-se que a punção ocorre a cada período de 8 bits. Depois que a punção ocorreu, os bits são então multiplexados e transmitidos. Um estudo de códigos RCPC pode ser encontrado em [24].

4. O CODIFICADOR DE VÍDEO MODIFICADO

Este capítulo introduz um método de gerar seqüências de vídeo com erro recuperável baseado em codificação combinada de canal/fonte e técnicas de cancelamento de erro e sincronização. O algoritmo de codificação de fonte de vídeo é o H.263 versão 2 com algumas modificações para permitir o codificador trabalhar com codificação de canal/fonte combinada, bem como fornecer cancelamento de erro no decodificador. Estas modificações são explanadas mais tarde neste capítulo.

Por causa da facilidade pela qual a sincronização pode ser perdida dentro do trem de pulsos codificado pelo H.263 (devido ao comprimento variável das palavras código produzidas), propõe-se nesta dissertação a aplicação de um sincronismo suplementar aos dados por meio de técnicas de EREC (*Error Resilient Entropy Coding*) preconizadas por Redmill e Kingsbury [25]. Os experimentos são conduzidos forçando uma taxa de transmissão de 1.0 bits/pixel, incorporando ao decodificador H.263 o cancelamento de erro nos domínios temporal e espacial. As simulações são executadas com várias seqüências, usando suplementarmente taxas fixas de códigos corretores de surtos de erro de *Reed-Solomon* bem como taxas variáveis de códigos RCPC (*Rate Compatible Punctured Convolutional Codes*).

4.1 CARACTERÍSTICAS DA SEQÜÊNCIA DE VÍDEO

Os vídeos usados neste estudo consistem de várias seqüências de imagens que contêm desde pequenos a grandes movimentos. Todos eles são movimentos de cabeça e ombro. Cada um deles foi formatado com o quadro padrão QCIF com 144 linhas e 176 pixels/linha em cores. As seqüências de VICTINHO somente contêm movimento de cabeça com um fundo estático. A seqüência de LUCAS contém ambos os movimentos de cabeça e tronco, além de movimento do fundo forçado pelo deslocamento da câmera.

Em termos de grau de dificuldade de medida da codificação pelo número de bits de saída pelo codificador H.263, para uma determinada taxa de quadros, a seqüência VICTINHO é o de menor grau devido a haver um mínimo de mudanças de um quadro para outro. A seqüência LUCAS é a de maior dificuldade, ou seja, requer o maior número de bits para qualquer taxa de quadros.

4.2 SIGNIFICÂNCIA DA INFORMAÇÃO DE FONTE

Esta seção descreve a determinação da significância dos bits da fonte codificada e sua contribuição para a PSNR do sinal decodificado. Já que o H.263 é um codificador de vídeo que usa tanto compressão espacial como temporal, o método de obtenção da informação significativa é descrito a seguir:

Se a taxa total do canal (em bits/pixel) é R_T, então

$$R_T = R_F + R_C \tag{4.1}$$

em que R_F e R_C são as taxas de saída nos codificadores de fonte e canal respectivamente. R_F determina a qualidade da imagem decodificada, enquanto R_C determina a quantidade de proteção de erro disponível para os bits codificados de fonte. Se R_T é restringido pelas limitações de largura de banda, então para um canal suscetível a erro terá de haver um ponto de equilíbrio entre R_F e R_C de modo a permitir a recepção de imagens com uma determinada qualidade e livre de erros. Se as condições do canal são deterioradas, R_C deve ser aumentado em detrimento de R_F (qualidade da imagem) e vice-versa.

Nesta seção é feita a análise do trem de pulso e sintaxe do H.263 para vídeo e suas conseqüências caso haja uma perturbação nestes bits (por exemplo, perda de sincronismo). Em cada parâmetro é numerado entre parênteses a quantidade de bit usada. Somente os argumentos relevantes são descritos. O detalhamento completo da sintaxe do H.263 está no Capítulo 2.

- **Código de Início de Cena (22 bits)**

Este é o cabeçalho de Código de Início de Cena (PSC – *Picture Start Code*), que define o início de todo quadro codificado. Os primeiros 17 bits formam o código de início de grupo de blocos (GBSC – *Group of Blocks Start Code*) e os últimos 5 o número de grupo (GN). O GN é 00000 para o PSC e 11111 para o código de final de seqüência (EOS). Um erro nestes 22 bits acarreta a perda de sincronismo e o decodificador não fica apto a decodificar a seqüência da cena.

- **Referência de Tempo (8bits)**

Estes bits representam a referência de tempo e informam quantos quadros foram pulados mais um durante a codificação temporal da seqüência. Um erro nessa palavra causa a inserção de um quadro extra na cena corrente ou reduz o número de quadros armazenados na seqüência de imagem decodificada. Isso acarreta a falta de sincronismo entre voz e imagem até que a sincronização seja restabelecida.

- **Tipo de Informação (13 bits)**

Os bits de tipo de informação (PTYPE) informam a respeito da cena completa. Dentro do PTYPE os bits de 1 até 5 não são usados e são ignorados pelo decodificador. Bits de 6 até 8 representam o tamanho da imagem e para os experimentos, assume-se ser QCIF ou 011. O bit 9 é importante já que ele informa ao decodificador o tipo de codificação que está sendo usada (inter ou intra modo). A escolha de codificação inter ou intra também afeta a presença ou ausência de bit de indicação de macrobloco codificado (a ser discutido em 4.2.7). Bits 10 até 13 são indicam o uso de qualquer dos modos opcionais e conseqüentemente são ignorados pelo codificador.

- **Quantizador de Informação (5 bits)**

Estes 5 bits (PQUANT) indicam o tamanho do passo (de 2 a 62 em incrementos de 2) do quantizador. O bit mais à esquerda ou primeiro bit é o bit mais significativo. Isto ocorre porque um erro no primeiro bit resulta na maior mudança no tamanho do passo do quantizador levando à deterioração da PSNR.

- **Presença Contínua Multiponto (1 bit)**

Um erro no bit CPM (*Continuous Presence Multipoint*) comuta a opção de presença multiponto. O decodificador ignora este bit.

- **Inserção de Informação extra (1 bit)**

Um erro neste bit altera os próximos 9 bits a serem considerados como informação válida. A norma H.263 especifica que os decodificadores não devem usar este bit, que terá futuramente sua compatibilidade adicionada.

- **Indicação de Macrobloco Codificado (1bit)**

Um erro neste bit (COD – COD*ed macroblock indication*) resulta no tratamento dos dados seguintes ao COD como dados de codificação de macroblocos (uma mudança de 0 para 1) ou a indicação para o decodificador que não existem dados de macroblocos em seguida ao bit de COD (mudança de 0 para 1). Em ambos os casos a omissão ou inclusão de dados causa a propagação de erros pelo resto do trem de pulsos.

- **MNBPC (1 – 9 bits)**

Este é um código de comprimento variável dependente do modo de codificação de macrobloco usado. Um erro nessa palavra resulta no uso de padrões errados quando da decodificação dos padrões de crominância U e V.

- **Padrão de Bloco Codificado para Luminância (2 – 6 bits)**

Similar ao MNBPC, esta palavra, abreviada para CBPY, também varia em comprimento e um erro nesta palavra resulta no uso de padrões errados quando da decodificação dos blocos de luminância Y. Esses erros são visualizados como artefatos verdes ou magenta na imagem decodificada.

- **DQUANT (2 bits)**

Um erro neste bloco causa o uso pelo decodificador de valores diferenciais errados quando da mudança de passo pelo quantizador usado. O tamanho do passo varia de {-2, -1, 1, 2} e é adicionado ao QUANT. O QUANT é limitado para a faixa entre 1 e 31.

- **Dados de Vetor de Movimento (1 – 13 bits)**

Um erro nos Dados de Vetor de Movimento pode causar alterações catastróficas no que diz respeito a direção do movimento do objeto na cena na decodificação. Em termos práticos isto mudaria a posição dos blocos na cena resultante. Sendo uma palavra código de comprimento variável, o erro poderia se propagar pelo trem de pulsos, levando a significativos artefatos visuais. Esta palavra está presente somente no modo inter quadro.

- **INTRADC (8 bits)**

Esta é uma palavra código de comprimento fixo que representa o nível a ser usado na geração do intracoeficiente DC decodificado na transformada DCT inversa. Um erro no INTRADC afeta o valor médio do bloco e, portanto, o valor médio de luminância do bloco.

- **TCOEF (3 – 22 bits)**

Estes códigos de comprimento variável representam os coeficientes AC da operação DCT. Dependendo da posição do erro, as componentes AC podem ser deslocadas dentro do bloco causando artefatos de alta freqüência que aparecem na imagem decodificada como pequenos quadrinhos fora de foco.

4.2.1 Propagação de Erro no fluxo de dados do Codificador H.263

Como o fluxo de dados do H.263 é composto de códigos dos blocos de comprimentos fixos e variáveis, a corrupção de uma palavra código de comprimento variável pode resultar na melhor das hipóteses em informação decodificada totalmente falsa. Na pior das hipóteses, na propagação de erros decodificados pelo trem de pulsos acarretando uma informação de saída desastrosa do decodificador H.263.

Por exemplo, se um erro residual permanecer no valor de DQUANT (ver Tabela 4.1), uma das palavras código de comprimento fixo utilizada pelo codificador H.263, o erro que resulta faz com que o valor diferencial utilizado para atualizar o quantizador seja 1 em vez de 2 original, como mostrado na Figura 4.1.

Tabela 4.1 – Códigos de comprimento fixo para DQUANT.

Índice	Valor Diferencial	DQUANT
0	-1	00
1	-2	01
2	1	10
3	2	11

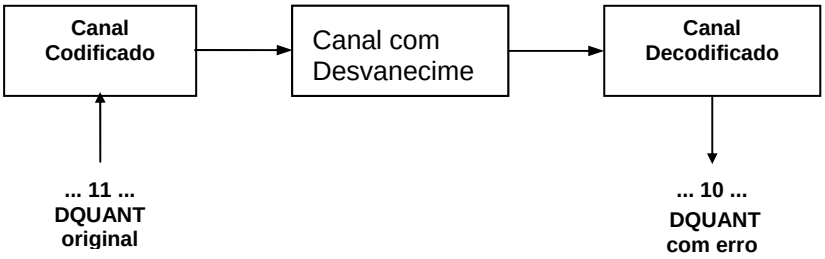


Figura 4.1 – Erro residual em palavras código DQUANT de comprimento fixo.

No caso de palavras código de comprimento variável, o bit de erro residual pode mudar uma palavra código e representar outra, ou uma palavra código que não se encontra na tabela. Isto pode então resultar na decodificação de palavras código finalizando em pontos errados, ou de o codificador procurar continuamente por bits que fazem uma palavra código não existente. Um exemplo deste tipo de propagação de erro é dado na Figura 4.2.

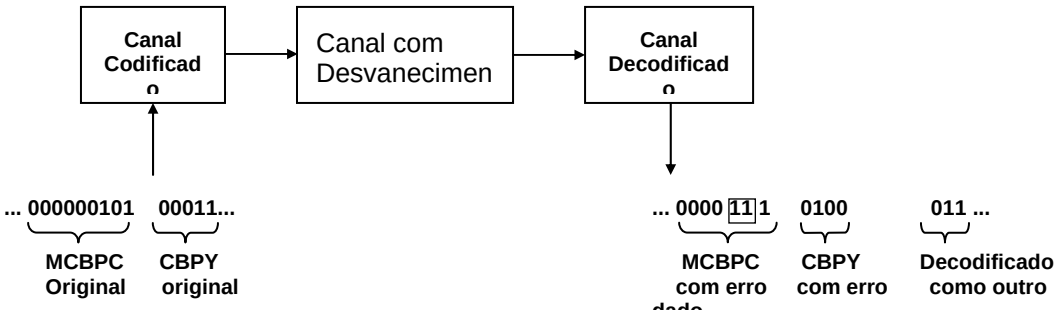


Figura 4.2 – Erro residual em palavras código MCBPC de comprimento variável causando erro de decodificação de fonte de MCBPC e a subseqüente palavra código CBPY. Pode ocorrer a propagação do erro ao restante do trem de pulsos.

As palavras código MCBPC e CBPY transmitidas são 0000 0010 1 e 0001 1, respectivamente. Estas palavras código correspondem àquelas de índice 7 na Tabela 4.2 para MCBPC, e índice 4 da Tabela 4.3 para CBPY (inter).

Tabela 4.2 – Códigos de comprimento variável para MCBPC (inter).

Índice	Tipo de MB	CBPC (56)	Número de bits	Código
0	0	00	1	1
1	0	01	4	0011
2	0	10	4	0010
3	0	11	6	0001 01
4	1	00	3	011
5	1	01	7	0000 111
6	1	10	7	0000 110
7	1	11	9	0000 0010 1
8	2	00	3	010
9	2	01	7	0000 101
10	2	10	7	0000 100
11	2	11	8	0000 0101
12	3	00	5	0001 1
13	3	01	8	0000 0100
14	3	10	8	0000 0011
15	3	11	7	0000 011
16	4	00	6	0001 00
17	4	01	9	0000 0010 0
18	4	10	9	0000 0001 1
19	4	11	9	0000 0001 0
20	*Stuffing*	-	9	0000 0000 1

Tabela 4.3 – Códigos de comprimento variável para CBPY.

Índice	CBPY (intra) (1234)	CBPY (inter) (1234)	Número de bits	Código
0	0000	1111	4	0011
1	0001	1110	5	0010 1
2	0010	1101	5	0010 0
3	0011	1100	4	1001
4	0100	1011	5	0001 1
5	0101	1010	4	0111
6	0110	1001	6	0000 10
7	0111	1000	4	1011
8	1000	0111	5	0001 0
9	1001	0110	6	0000 11
10	1010	0101	4	0101
11	1011	0100	4	1010
12	1100	0011	4	0100
13	1101	0010	4	1000
14	1110	0001	4	0110
15	1111	0000	2	11

Depois do canal decodificado, dois bits de erro residual permanecem como indicado pelo quadrado sobre o trem de pulsos de canal decodificado na Figura 4.2. Supondo que não há bit de erros prévios para o decodificador de fonte deste MCBPC, o decodificador H.263 procede a decodificação do trem de pulsos como mostrado, produzindo um MCBPC completamente diferente. O MCBPC de fonte decodificado, representado por 0000111, agora corresponde ao índice 5 da Tabela 4.1. Além do mais, vale salientar que sendo a palavra código CBPY, que segue MCBPC, de comprimento variável, ela também é decodificada errada como 0100, ou índice 12 da Tabela 4.3 para o CBPY (inter).

Palavras código de comprimento fixo ou variável subseqüentes a esse erro estão sujeitas a um efeito dominó que pode resultar em falha catastrófica do decodificador ao reproduzir a seqüência de vídeo transmitida.

4.3 PROTEÇÃO DE ERRO NÃO-UNIFORME

Esta seção mostra um exemplo de codificação de canal de modo a realizar a proteção de erro não uniforme ao trem de pulsos. Mas, devido o codificador fazer uso da informação de significância de fonte do trem de pulsos do H.263, primeiramente se apresenta essa classificação de significância.

4.3.1 Classificação da Significância

A atribuição de classes de significância aos parâmetros do codificador H.263 baseia-se no critério de indicar a mais alta classe de significância para os bits que causam maior degradação da PSNR. As classes são representadas por um valor numérico em que o maior número é a mais alta classe de significância.

- **Classe 2**

O conjunto de bits inicial de cena (cabeçalho) é considerado o dado mais importante dentro do trem de pulsos codificado pelo H.263, consistindo do código de início de cena, referência temporal, informação de tipo, informação do quantizador, o bit multiponto de presença contínua e a informação de inserção extra. Este conjunto de 50

bits faz com que o cabeçalho seja considerado a classe mais alta. O grupo de blocos que está presente no início de cada camada GOB também pode ser considerado da mesma classe, já que compartilham a mesma função. A única diferença é que o cabeçalho GOB serve para identificar cada GOB em vez do início de um quadro de cena. Em uma seção mais adiante é visto como remover efetivamente o cabeçalho de GOB produzindo uma economia em termos do número de bits requeridos para codificação de fonte.

Da mesma forma que o PSC, os bits pertencentes ao TR, PTYPE, PQUANT, COM e PEI são tratados também como classe 2. Nos quadros inter-codificados, o bit de COD que está presente no início de cada camada de macrobloco também é tratado como um bit classe 2 por indicar se os bits seguintes pertencem a um macrobloco codificado (COD = 0) ou se não há informação adicional a ser transmitida para aquele macrobloco (COD = 1).

- **Classe 1**

Esta classe de significância aplica-se principalmente ao restante dos parâmetros da camada de macroblocos e são os bits de MCBPC, CBPY, DQUANT e MVD. Na camada de blocos, também se considera o coeficiente do INTRADC como classe 1. Estes bits controlam a informação de luminância e crominância, mudando o valor do quantizador, bem como qualquer vetor de movimento durante o modo de codificação interquadro.

- **Classe 0**

Finalmente, reserva-se a última classe de significância, classe 0, para o TCOEF codificado de comprimento variável, que compõe a massa de dados do H.263.
A codificação de canal usada é baseada no codificador RCPC descrito no Capítulo 3. Dadas as três classes de significância, como mostrado na figura 4.3, a ordem de importância é dada por

$$\text{Classe 2} > \text{Classe 1} > \text{Classe 0}. \qquad (4.2)$$

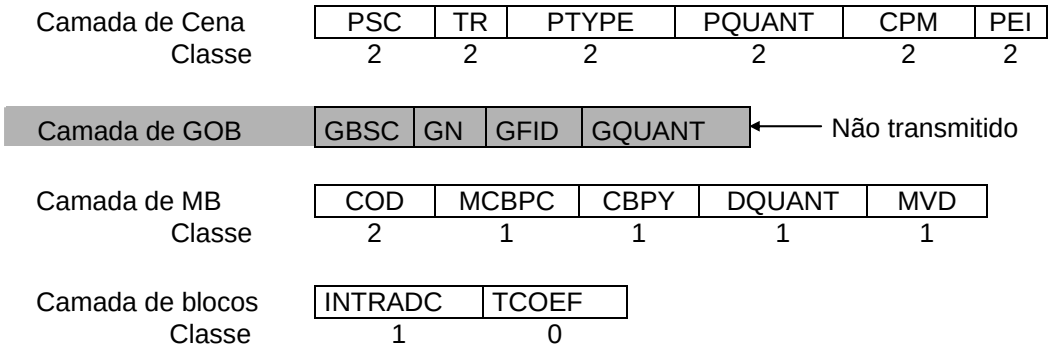


Figura 4.3 – Classificação do trem de pulsos do H.263 dentro das várias classes de significância. Quanto maior o número, maior a significância.

As matrizes de punção escolhidas serão aquelas que correspondam às taxas 1/3, 1/2 e 2/3 para as classes 2, 1 e 0 respectivamente. Estas matrizes são dadas na Tabela 4.4.

Tabela 4.4 – Gerador Polinomial e as matrizes de puncionamento para códigos RCPC às taxas de 1/3, 1/2 e 2/3.

Gerador Polinomial	Matriz de Puncionamento		
	Taxa 1/3	Taxa 1/2	Taxa 2/3
$g^{(0)} = 1 + D^3 + D^4$	11111111	11111111	11111111
$g^{(1)} = 1 + D + D^2 + D^4$	11111111	11111111	10101010
$g^{(2)} = 1 + D^2 + D^3 + D^4$	11111111	00000000	00000000
$g^{(3)} = 1 + D + D^3 + D^4$	00000000	00000000	00000000

Deve-se lembrar que cada taxa tem sua própria distância livre (d_{free}) a qual define os erros de t que cada taxa pode corrigir. O parâmetro d_{free} é igual a 11, 7 e 4 para as taxas 1/3, 1/2 e 2/3 respectivamente. O codificador de canal RCPC e os mecanismos de punção associados usados neste codificador são mostrados na Figura 4.4.

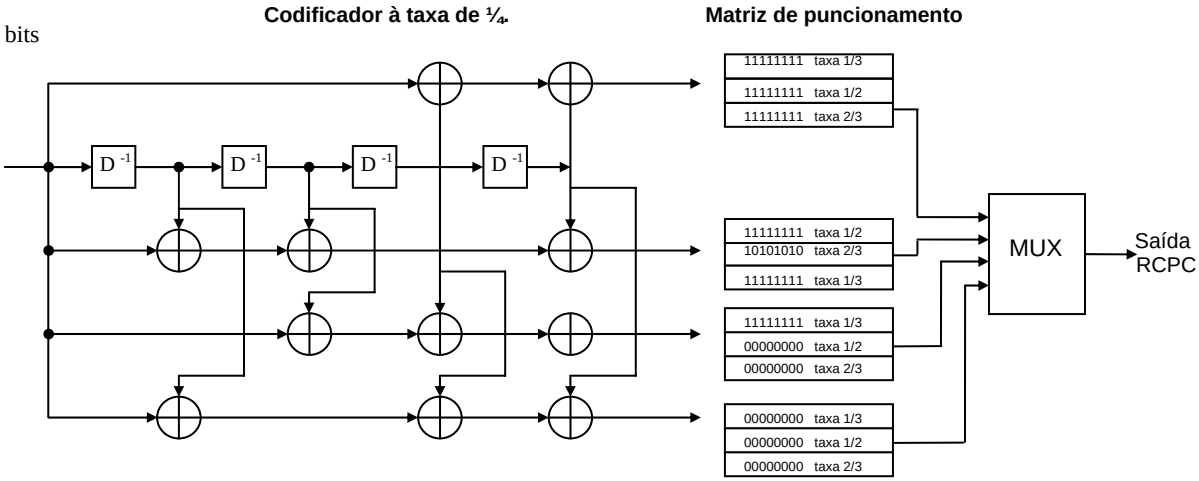


Figura 4.4 – Codificador RCPC com taxa de código mãe de 1/4 e matrizes puncionada às taxas de 1/3, 1/2 e 2/3.

4.4 SINCRONIZAÇÃO

A transmissão de dados de vídeo tem melhores resultados em termos de PSNR com proteção de erro não-uniforme sobre um canal com desvanecimento que os mesmos dados protegidos somente com um código RS com correção de erro fixo [26]. O método [26] consiste do sistema representado pelo diagrama em blocos mostrado na Figura 4.5.

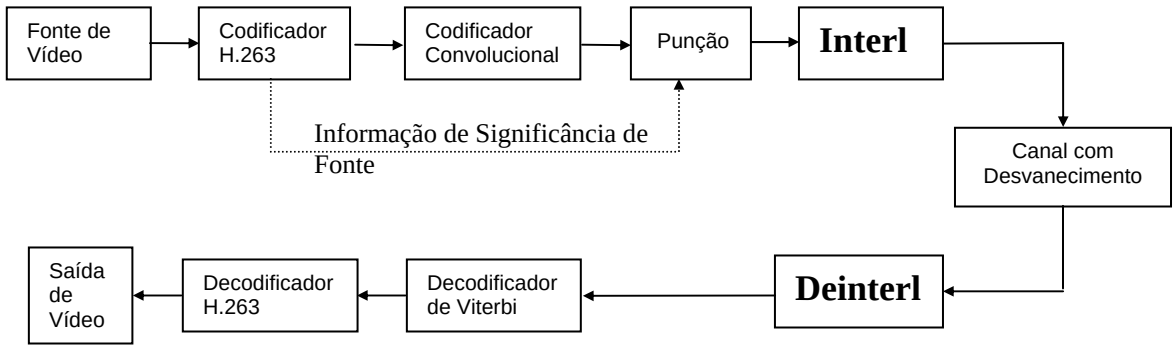


Figura 4.5 – Diagrama em blocos de sistema usado na transmissão de vídeo H.263 protegido contra erro não uniforme sobre canais de rádio móveis.

O diagrama em blocos mostra que o sinal de vídeo codificado é imediatamente codificado em canal usando a técnica de proteção de erro não-uniforme e transmitido através de um canal com desvanecimento. Devido à natureza variável (do comprimento) das palavras código do H.263, cada macrobloco é também de comprimento variável. A transmissão serial de tais dados por um canal propenso a erros causa a propagação dos erros ao longo do trem de pulsos. Isto por sua vez acarreta a perda de sincronismo do decodificador de fonte, resultando assim na decodificação de quadros de vídeo espacial e temporalmente corrompidos.

A proposta, então, para manter o sincronismo é inserir palavras código de sincronismo periodicamente no trem de pulsos, como por exemplo, os cabeçalhos inseridos em cada GOB. Já que a palavra código variável mais longa do H.263 é de 13 bits, as palavras de sincronismo utilizadas devem exceder esse comprimento e serem únicas e robustas o suficiente para serem detectadas na presença de bits de erro. O problema de adicionar essas palavras de sincronismo é o aumento do número de bits a serem transmitidos. Isto leva a um aumento da taxa de bits, que por sua vez reduz a taxa de quadros em um canal com uma largura de banda já bem restrita.

Um método que preserva o sincronismo bem como previne a propagação de erros no trem de pulsos, e que ainda aumenta muito pouco a taxa de bits é o método EREC [25].

4.4.1 O Algoritmo EREC

O algoritmo de codificação por entropia com erros recuperáveis (EREC) é um algoritmo que foi projetado para, quando aplicado a um trem de pulsos que consiste de códigos de comprimentos variáveis, aumentar prontamente a recuperação de erros aleatórios e em surto, bem como exibir uma gradual degradação destes erros na presença de canais nas piores condições. O EREC foi baseado em um trabalho anterior de Chen e Kingsbury [27], conhecido como ERPC (*Error Resilient Positional Code*) e usado para codificar informação posicional ou dados esparsos.

O algoritmo EREC aumenta a recuperação de erro e fornece gradual degradação pela reordenação dos blocos de comprimento variável em *slots* de dados de comprimento fixo, sendo então codificados e transmitidos. A Figura 4.6 mostra um exemplo gráfico de como a reordenação EREC funciona. Os blocos fonte são transferidos aos *slots*

EREC e deslocados para baixo por um. Cada bloco fonte residual completará os slots vazios subseqüentes até preenchê-los totalmente.

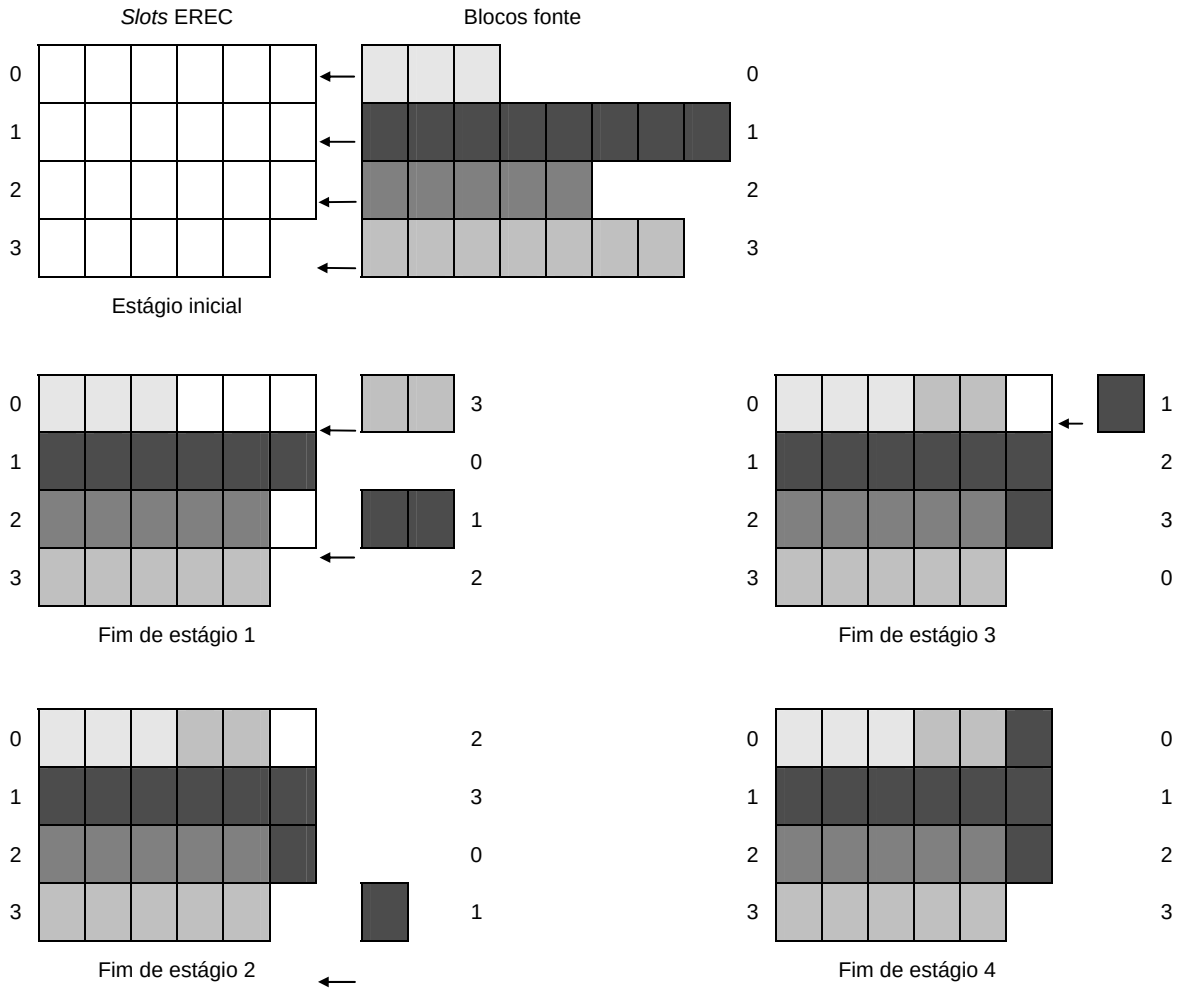


Figura 4.6 – Reordenação EREC. No final de cada estágio todas as fontes de blocos são deslocadas para baixo por um.

O algoritmo EREC é mais bem explicado usando um codificador de fonte imaginário cuja saída tem N_b blocos de dados de comprimento variável, sendo o *i*-ésimo bloco de comprimento b_i bits. Seja N_s o número de slots usados pelo codificador EREC, em que o *j*-ésimo *slot* é de comprimento s_j bits. Seja o número total de bits de saída (T_s) do codificador EREC igual a

$$T_s = \sum_{j=0}^{N_s - 1} s_j \qquad (4.3)$$

e o número total de bits provenientes da saída (blocos de comprimento variável) do codificador de fonte igual a

$$T_b = \sum_{i=0}^{N_b - 1} b_i \ . \tag{4.4}$$

Nesta implementação, é simples a escolha de fazer o número de slots igual ao número de blocos de saída, isto é, $N_s = N_b$. Similarmente, o número total de bits de saída do codificador EREC é ajustado para ser igual ao número total de bits de saída do codificador de fonte, ou $T_s = T_b$. Isto significa que não há aumento no número de bits de fonte codificada a não ser os do cabeçalho EREC, o que será explicado mais tarde.

A intenção agora é encontrar a dimensão, s_j, para cada *slot* de comprimento fixo usado pelo EREC. Isto é feito usando a seguinte fórmula:

$$s_j = \begin{cases} \bar{s} + 1, & 0 \le j \le \text{restante}\left(\dfrac{T_s}{N_s}\right) - 1 \\ \bar{s}, & \text{restante}\left(\dfrac{T_s}{N_s}\right) - 1 < j \le N_s - 1 \end{cases} \tag{4.5}$$

em que

$$\bar{s} = \left\lfloor \frac{T_s}{N_s} \right\rfloor \ . \tag{4.6}$$

Conseqüentemente, se o codificador de fonte imaginário produz $T_b = 23$ e $N_b = 4$ com os seguintes comprimentos de bloco $b_i = \{3, 8, 5, 7\}$, de (4.4) e (4.5) e fazendo $T_s = T_b$, resulta em $\bar{s} = 5$ e

$$s_j = \begin{cases} 6, & 0 \le j \le 2 \\ 5, & 2 < j \le 3 \end{cases}$$

Com referência à Figura 4.6, a reordenação dos bits dos blocos de comprimento variável dentro dos *slots* de comprimento fixo ocorre com o seguinte algoritmo

- **Algoritmo de reordenação de bits EREC**

begin

assign slot s_j *to block* b_i

for *each block* b_i ***do***

  if *block* b_i *has bits left to transfer*

    if *slot* s_j *is not full*

      transfer as many bits as possible from b_i *to* s_j

    else

      assign slot $s_{j=i+\phi_n}$ (mod N_s) *to block* b_i

end

No algoritmo transcrito, ϕ_n é uma seqüência predefinida que é usada pelo EREC no estágio n para designar o bloco i para o slot $j = i + \phi_n$ (mod N_s). A seqüência ϕ_n pode estar em uma ordem incremental, por exemplo, $\phi_n = \{1, 2, 3, 4\}$, como mostrado na Figura. 4.6. Contudo, uma ordem pseudo-aleatória de *slots* é preferível, pois resulta em codificadores mais rápidos devido à sua natureza não correlacionada [27]. Conseqüentemente, ϕ_n é escolhido numa seqüência pseudo-aleatória de 1 até N_s, por exemplo, $\phi_n = \{3, 1, 4, 2\}$.

Na implementação realizada, N_s e N_b são fixados iguais ao número de macroblocos em um quadro QCIF, em outras palavras, $N_s = N_b = 99$. A única informação requerida no decodificador EREC é T_s, desde que N_s e ϕ_n já sejam conhecidos.

O decodificador EREC funciona de modo inverso para decodificar os dados dos blocos de comprimento variável provenientes dos *slots* EREC de comprimento fixo. Durante a decodificação o bloco i é designado para o *slot* $j = i + \phi_n$ (mod N_s) para reverter o processo. A única complexidade na implementação do decodificador é que ele deve estar habilitado a reconhecer quando cada macrobloco tenha sido completamente decodificado e marcado como tal. Isto acarreta a inclusão de parte do decodificador

H.263 dentro do programa de decodificação EREC. O processo de decodificação é ilustrado na Figura. 4.7.

Desde que o *start* de cada *slot* EREC é o início de um macrobloco, o decodificador EREC pode ser modificado para inserir um cabeçalho GOB depois dos 11 macroblocos decodificados. Isto deixa óbvia a necessidade de transmitir o cabeçalho GOB para sincronização, resultando na economia de 176 bits por quadro de fonte codificado. Esta economia de bits pode ser usada para transmitir um valor de T_s altamente protegido por meio do canal.

A saída do codificador EREC segue a seguinte ordem, T_s, o cabeçalho H.263, e os *slots* de comprimento fixo FS1 até FS99, como mostrado na Figura 4.8. T_s são marcados na classe de alta significância (classe 2), de modo que o codificador de canal o codifica no mais alto nível de proteção disponível.

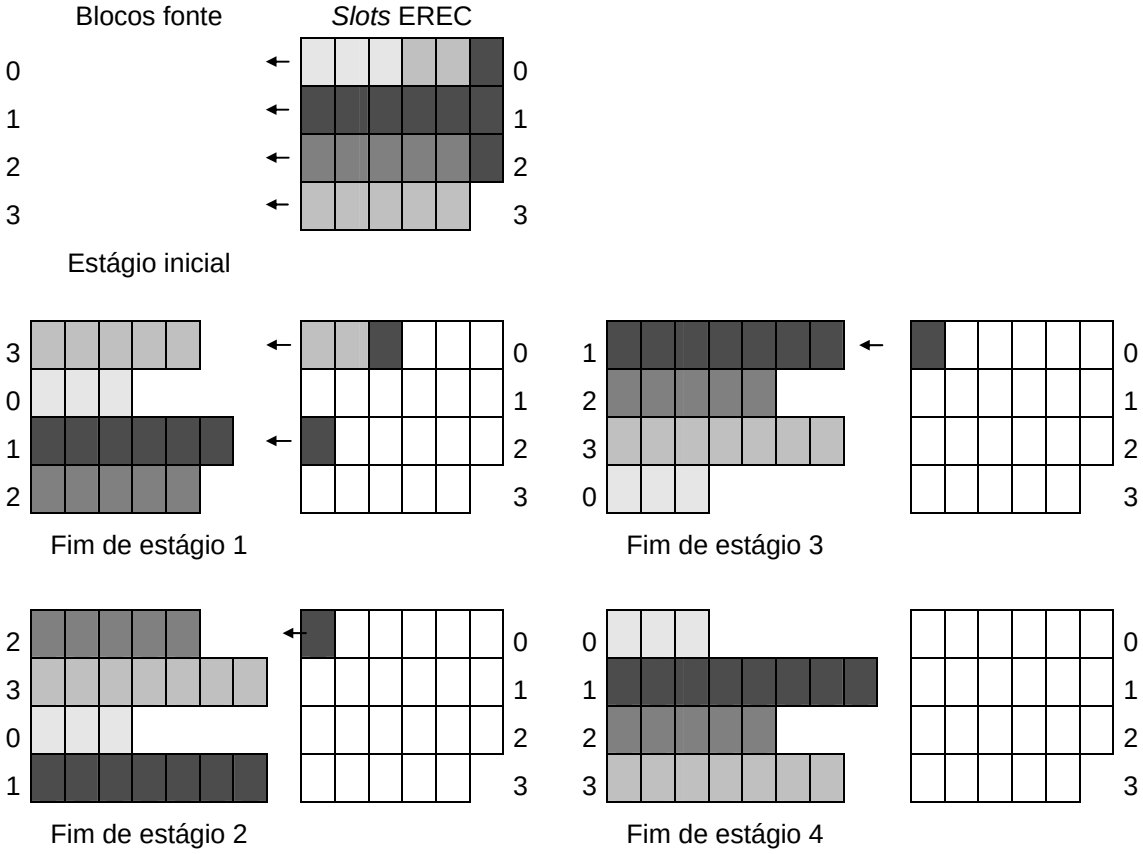


Figura 4.7 – Decodificação EREC. Ao final de cada estágio os blocos fonte são deslocados para baixo de uma unidade.

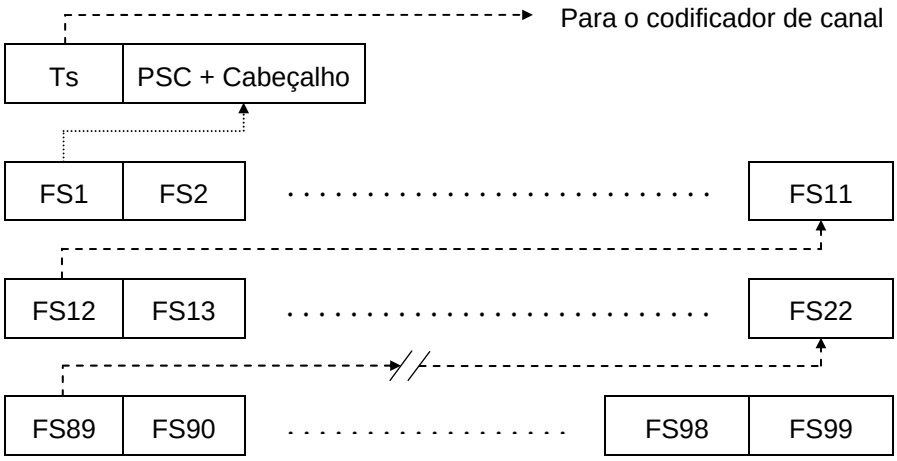
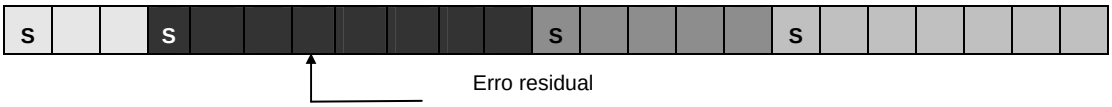


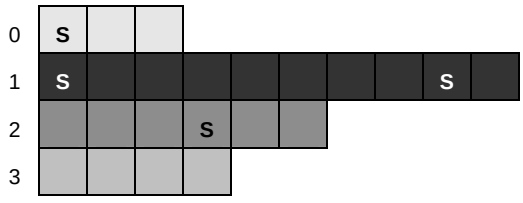
Figura 4.8 – Saída do codificador EREC.

A maneira como o EREC previne a propagação de erros é importante nos esquemas de codificação que usam particionamento de dados para produzir blocos de dados de comprimento variável contendo informações mais importantes no início do que no fim dos dados. Isto é particularmente verdade para os macroblocos produzidos pelo codificador H.263. Uma demonstração de como o EREC preserva o sincronismo de blocos e previne a propagação de erros será descrita a seguir. Considere um trem de pulsos construído de blocos de comprimento variável sem EREC, como mostrado na Figura 4.9. Estes blocos são os mesmos que foram usados na Figura 4.6.



(a)

(b)



(c)

Figura 4.9 – Exemplo de propagação de erro sem EREC. O **S** representa o ponto de partida original de cada bloco de comprimento variável. (a) Erro residual remanescente no segundo bloco de comprimento variável. (b) O erro de decodificação propaga-se do segundo bloco para o terceiro e quarto blocos. (c) Os blocos rearranjados são diferentes daqueles blocos originais da Figura 4.6 (com os blocos 2 e 3 de partida errados).

Na Figura 4.9a, um erro residual no bloco 1 resulta em dados de comprimento variável serem interpretados erroneamente como mostrado na Figura 4.9b. O que aconteceu é que o bloco 1 foi considerado de comprimento maior que o real, acarretando com isso o deslocamento do início dos blocos 2 e 3. Quando o dado é rearranjado de volta a sua forma, como mostrado na Figura 4.9c, fica claro o erro no sincronismo do ponto de partida dos blocos 2 e 3. Ao mesmo tempo, são perdidas importantes informações particionadas nesses blocos.

Agora considere um erro residual remanescente nos mesmos blocos de comprimento variável que tenha usado o EREC, como mostrado na Figura 4.10a. No estágio inicial de decodificação dos slots EREC supõe-se que o erro tenha feito o bloco 1 terminar prematuramente. Isto significa que o restante dos dados do bloco 1, bloqueados nos *slots* 0 e 2, não serão decodificados, como mostrado na Figura 4.10b. Seguindo o processo de decodificação como pode ser visto no final do estágio 1, a decodificação está completa mesmo que o decodificador EREC não tenha ido até o final de todos os estágios. No final do estágio 4 pode ser observado que o ponto de partida de todos os blocos foi preservado, e que existem dois dados restantes pertencentes ao bloco 1 ainda nos *slots* EREC. Pode também ser visto que o bloco 1 foi encurtado para dois dados. Contudo, os blocos perdidos estão próximos do final do bloco 1 e com o tipo de particionamento de dados usado, os dados próximos do final contêm menos informação importante.

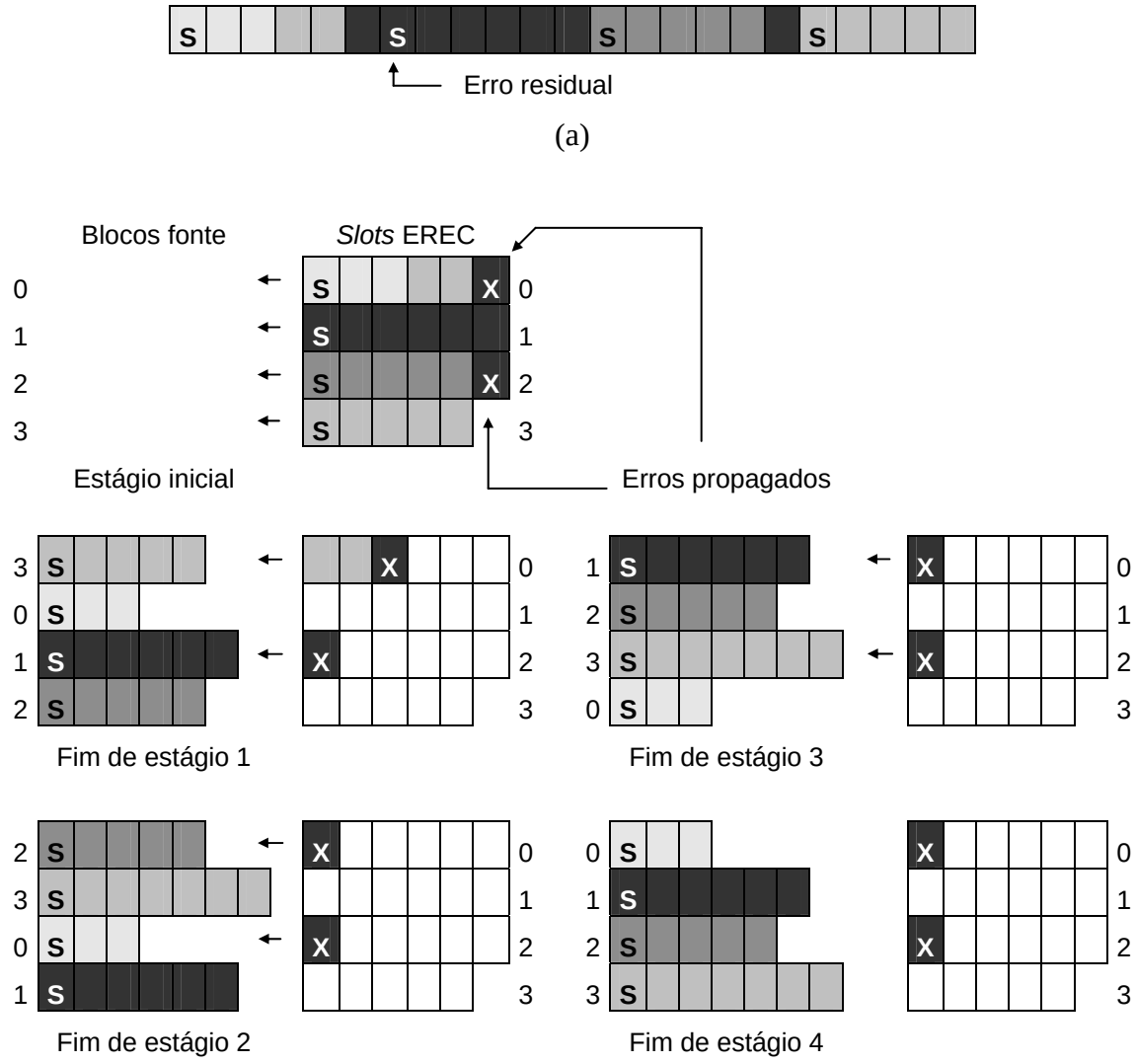


Figura 4.10 – Exemplo de propagação de erro com EREC. S e X representam, respectivamente, o ponto de partida original de cada bloco de comprimento variável e os blocos errados. (a) Erro residual remanescente no ponto de partida do segundo bloco de comprimento variável. (b) Devido à maneira que o EREC funciona, o ponto de partida de cada bloco é preservado mesmo que o erro residual tenha causado o término prematuro do bloco 1 e deixando com isso dois blocos restantes nos *slots* EREC.

4.5 CANCELAMENTO DE ERRO

Como mencionado nos capítulos anteriores, o cancelamento de erro é implementado para remover artefatos visuais remanescentes durante a decodificação de fonte. Para codificação de imagens congeladas usando algoritmo SPIHT (*Set Partitioning in Hierarchical Trees*) de Said e Pearlman [28], faz sentido concentrar no cancelamento de erros que causam grande quantidade de distorção, como o erro em $n_{máx}$, levando à saturação do nível médio de luminância da foto. Neste sistema de codificação de vídeo, o foco está concentrado sobre o cancelamento de erros no domínio da freqüência e no domínio do tempo, já que cada domínio produz artefatos visuais que resultam em diferentes tipos de degradação de imagem.

4.5.1 Cancelamento de erro no domínio da freqüência

No domínio da freqüência, a maior degradação visual, mas de erro cancelável é um erro fora de faixa no valor produzido pelos coeficientes da transformada multiplicada pelo nível de quantização, que é utilizado na transformada discreta inversa do coseno (IDCT). A Figura 4.11a mostra o artefato de alta-frequência produzido por este tipo de erro.

(a) (b)

Figura 4.11 – (a) Quadro de uma seqüência mostrando o artefato visual, que lembra um padrão xadrez. (b) O mesmo quadro com aplicação do cancelamento de erro.

Pelo uso das seguintes fórmulas, os valores de reconstrução são obtidos pela DCT inversa,

$$|REC| = \begin{cases} QUANT \cdot (2 \cdot |LEVEL| + 1|) \ , & (QUANT \mod 2) = 1 \\ \\ QUANT \cdot (2 \cdot |LEVEL| + 1|) - 1 \ , & (QUANT \mod 2) = 1 \end{cases} \quad (4.7)$$

Dado que o maior valor de |LEVEL| é igual a 127 e é listado na Tabela 4.5, o maior valor de (2 · |LEVEL| + 1|) é 255. Devido a erros residuais de bits nos coeficientes da transformada, o valor de (2 · |LEVEL| + 1|) poderia possivelmente exceder 255, a qual resulta no aparecimento destes artefatos de alta-frequência na imagem decodificada.

Tabela 4.5 – Códigos de comprimento fixo para LEVELS

Índice	Nível	Código
-	-128	Proibido
0	-127	1000 0001
⋮	⋮	⋮
125	-2	1111 1110
126	-1	1111 1111
-	0	Proibido
127	1	0000 0001
128	2	0000 0010
⋮	⋮	⋮
253	127	0111 1111

A maneira mais simples de cancelar estes artefatos é testar a condição de (2 · |LEVEL| + 1|) fora da faixa antes de aplicar a DCT inversa. Se uma condição fora de especificação é detectada, isto é (2 · |LEVEL| + 1|) > 255, coloca-se este coeficiente em zero. Ao fazer isso, elimina-se o artefato de alta-frequência. Este método de cancelamento de erro pode ser visto na Figura 4.11b.

4.5.2 Cancelamento de erro no domínio do tempo

Implementa-se o cancelamento de erro no domínio temporal detectando dois tipos de erros. O primeiro é resultado da decodificação errônea de palavras código de comprimento variável encontradas nas tabelas de decodificação H.263. Erros de bits residuais levam o decodificador H.263 a manter uma procura por uma palavra código não-existente entre aquelas listadas nas tabelas de decodificação H.263. Se não é detectada, o decodificador H.263 procura em vão até o fim do trem de pulsos ou trava quando o buffer de entrada sinalizar a sobrecarga de bits de entrada.

O método de cancelamento para este tipo de erro é recolocar o grupo de blocos no quadro *n* com o GOB do quadro previamente decodificado, *n* – 1. A maneira do algoritmo de recolocação funcionar é procurar o *buffer* de trem de pulsos entrante para um grupo válido de GBSC e de GN. Isto é possível porque, como mencionado anteriormente, o algoritmo EREC insere um GBSC e um GN válido para cada 11 macroblocos. Isto permite ao decodificador H.263 localizar com sucesso o próximo GBSC e GN. O GN encontrado é usado como um guia para substituir o corrente GOB com erro pelo GOB copiado do quadro decodificado anterior. Se o algoritmo de cancelamento detecta o código de início de imagem (a qual é no caso um cabeçalho GOB com um GBSC e um GN de 00000) durante a procura, todo o grupo de blocos daquele com erro até o final do quadro é substituído.

5. SIMULAÇÃO E RESULTADOS

A simulação para verificar a efetividade da codificação de vídeo com erro recuperável é apresentada com o uso de um *software*, **UB-Live-263v2-PC,** que é um codec H.263 que permite a codificação e decodificação de seqüências e arquivos de vídeo de acordo com diferentes parâmetros e opções especificadas pelo codificador objeto desta dissertação. São usadas duas seqüências QCIF, VICTINHO e LUCAS, para representar os dois extremos da codificação de vídeo, o primeiro com o mínimo de movimentos, e o segundo com bastante movimento. Os parâmetros para cada seqüência são mostrados na Tabela 5.1.

Tabela 5.1 – Parâmetros de simulação para o sistema de vídeo com erro recuperável.

	VICTINHO		LUCAS	
Formato da cena	QCIF 176 x 144			
Número de quadros	200			
Taxa de bits (kbps)	9,6		30	
Taxa de quadros (quadros/s)	5,0			
Nível de quantização	16			
Cancelamento de erro	Não	Sim	Não	Sim
Cabeçalho GOB	Sim	Não	Sim	Não
Tamanho em *bytes*	44416	48448	119104	121920

Devido ao alto conteúdo de movimento da seqüência LUCAS a taxa alvo de bits do codificador H.263 é 30 kbps, enquanto que a seqüência VICTINHO, de baixo conteúdo de movimento, é codificada a 9,6 kbps. Os resultados de PSNR da componente de luminância (Y), e de crominância (U) e (V) são usados como medida de qualidade. As seqüências são comparadas a uma seqüência de vídeo original.

Os resultados também são apresentados para as mesmas seqüências com e sem o benefício do algoritmo EREC para mostrar as melhorias obtidas com ele. Aquelas seqüências que foram simuladas sem o algoritmo EREC requereram a inclusão de cabeçalhos GOB a cada 11 macroblocos de acordo com a norma H.263.

Todas as seqüências são decodificadas usando um decodificador H.263 implementado com cancelamento de erro temporal, pois de outra maneira resultaria na falha da decodificação. Aquelas seqüências sem o benefício do algoritmo EREC também não têm implementado o cancelamento de erro em freqüência no decodificador H.263 e são rotuladas “Sem EREC”. Aquelas seqüências com o algoritmo EREC incluem a implementação de cancelamento de erro em freqüência no decodificador H.263, e são rotuladas “Com EREC”. Isto é feito para que possam ser visualizadas as vantagens do uso de cancelamento de erro em frequência dentro do decodificador H.263.

5.1. RESULTADOS

5.1.1. Comparação quantitativa

O primeiro conjunto de resultados consiste de gráficos de PSNR da componente de luminância (Y). O esquema de codificação de vídeo de erro recuperável consistindo do codificador de fonte H.263, algoritmo EREC e o decodificador H.263 com cancelamento de erro implementado é comparado contra os resultados obtidos sem o EREC implementado. A Figura 5.1 mostra gráficos esses valores de PSNR.

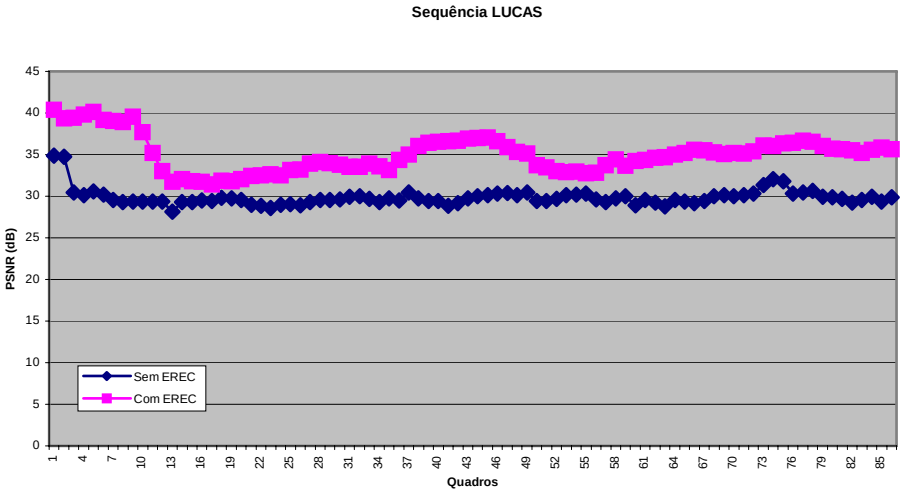


Figura 5.1a – PSNR da seqüência LUCAS.

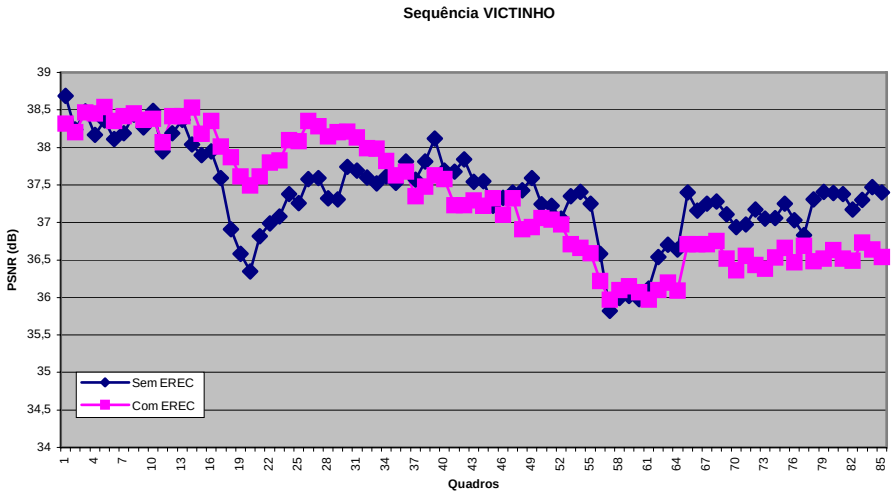


Figura 5.1b – PSNR da seqüência VICTINHO.

Os resultados obtidos para a seqüência LUCAS com considerável movimento demonstram a eficiência do decodificador verificando-se uma sensível melhora na PSNR. Para a seqüência VICTINHO quase sem movimento, quase não há diferença porque se for observado praticamente só há movimento labial.

5.1.2. Comparação qualitativa

Foi mencionado no início que as curvas de PSNR fornecem apenas resultados um tanto subjetivos, principalmente quando tenha sido implementado cancelamento de erro em freqüência e nível de movimento da seqüência for mínimo. Conseqüentemente é melhor escolher vários quadros de cada seqüência para ilustrar as diferenças visuais obtidas em relação à versão original do vídeo.

Figura 5.2a – Seqüência VICTINHO original Quadro 54.

Figura 5.2b – Seqüência VICTINHO decodificada sem EREC.

Figura 5.2c – Seqüência VICTINHO decodificada com EREC.

Figura 5.2d – Seqüência LUCAS .original Quadro 02.

Figura 5.2e – Seqüência LUCAS decodificada sem EREC.

Figura 5.2f – Seqüência LUCAS decodificada com EREC.

As Figuras 5.2a e 5.2d mostram as imagens originais, quadros 54 e 02, para as seqüências VICTINHO e LUCAS, respectivamente. As seqüências originais são codificadas e decodificadas sem o benefício do EREC e mostradas nas Figuras 5.2b e 5.2e (VICTINHO e LUCAS, respectivamente). A seguir são codificadas e decodificadas com o benefício do EREC e mostradas nas Figuras 5.2c e 5.2f (VICTINHO e LUCAS, respectivamente). Todas estas imagens estão no tamanho natural do formato QCIF.

5.2. CONCLUSÃO

Este capítulo apresentou um método de codificação de vídeo com erro recuperável. Esse método usa várias técnicas discutidas em capítulos anteriores tais como o algoritmo de codificação de vídeo H.263 e a proteção a erro não uniforme baseado em códigos RCPC (que tem sua aplicação na codificação de canal). Primeiro, o vídeo é codificado em fonte usando o algoritmo de codificação de vídeo H.263. Então em vez de usar cabeçalhos GOB para manter a sincronização, estes são removidos e o algoritmo EREC é usado para fornecer uma sincronização mais efetiva e, no caso de transmissão em canais ruidosos, uma degradação suave de erro. O estudo da significância de fonte apresentado sobre o fluxo de dados H.263 pode ser usado aplicando códigos RCPC para o caso da transmissão sobre canais com desvanecimento.

Depois da decodificação de canal, aplica-se uma versão modificada do algoritmo EREC, no qual são reinseridos os cabeçalhos GOB no trem de pulsos H.263 a cada 11

macroblocos. A seguir, métodos de cancelamento de erro espacial e temporal baseados nos cabeçalhos GOB reintroduzidos são combinados com o decodificador H.263 para fornecer um sistema de decodificação de vídeo de erro recuperável.

Os resultados das simulações usando o **UB-Live-263v2-PC,** mostram efetividade deste sistema por fornecer imagens de melhor qualidade à saída do decodificador. A questão da transmissão, como por exemplo a simulação em canais com desvanecimento, poderá ser um estudo para dar prosseguimento a este trabalho.

Na questão da implementação prática deste sistema, atualmente já existem estudos que são a evolução do padrão H.263 e chamado de padrão H.264 ou H.26L que preconizam algo semelhante ao que foi descrito aqui (utilizam o algoritmo CABAC em vez do EREC, ou então, padrões de multimídia MPEG-4). Há inclusive alguns fabricantes que fornecem plataformas de desenvolvimento em *firmware* para teste destes *codec* de vídeo.

Para finalizar pode-se destacar que no transcorrer deste trabalho foram necessários praticamente todos os conhecimentos adquiridos no curso de mestrado, inclusive alguns com aprofundamento maior e outros que nem estavam na grade curricular. Esta dissertação é concluída na certeza que houve uma exigência de esforço grande até a conclusão deste trabalho, pois culminou inclusive com propostas de patente do sistema.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] SHANNON, C. E. *The Mathematical Theory of Communication*, Bell Systems Technical Journal, vol. 27, pág. 379-423, 1948.

[2] WALLACE,G. K. *The JPEG still picture compression standard*, Communication of the ACM, vol. 34, nº 4, pág. 30-45, Abril 1991.

[3] HUFFMAN, D. A. *A method for the construction of minimum redundancy codes*, Proceedings IRE, vol. 40, pág. 1098-1101, 1962.

[4] ITU-T, *H.261: Video Codec for Audio Visual Services at p x 64 kbits/s*, março 1993.

[5] ITU-T, *H.263: Video Coding for Low Bitrate Communication*, março 1996.

[6] ISO/IEC/JTC1/SC29, *Generic Coding for Moving Pictures and Associated Audio Information: Video*, março 1995.

[7] SERFERIDIS, V. e GHANBARI, M. *Generalized block matching motion estimation*, on Proceeding of SPIE: Visual Communication and Image Processing, Junho 1992, pág. 80-91.

[8] JAIN, J. R. e JAIN, A. K. *Displacement measurement and its applications in interframe coding*, IEEE Transactions on Communications, vol. 29, nº 4, pág. 1799-1808, dezembro 1981.

[9] CHEN, L. G., JEHNG, Y. S. e CHIUEH, T. D., *An efficient parallel motion estimation algorithm for digital image processing*, IEEE Transactions on Circuits and Systems for Video Technology, vol. 1, nº 4, pág. 378-385, dezembro 1991.

[10] BIERLING, M. *Displacement estimation by hierarchical blockmatching*, SPIE Vol. 1001 Visual Communication and Image Processing, pág. 942-951, maio 1988.

[11] ITU-T, H.263: *Video Coding for Low Bitrate Communication*, maio 1996.

[12] ITU-T, *H.263: Video codec for audio visual services at p x 64 kbits/s*, março 1993.

[13] ITU-T SG-15, *Video Codec Test Model, TMN5*, janeiro 1995.

[14] ITU-T, *H.263 Version 2: Video Coding for Low Bitrate Communication*, janeiro 1998.

[15] GALLAGER, R. G. *Information Theory and Reliable Communication*, John Wiley & Sons, NY, 1968.

[16] REED, I. S. e SOLOMON,G. *Polinomial codes over certain finite fields*, SIAM Journal on Applied Mathematics, vol. 8, pág. 300-304, 60.

[17] BOSE, R. C. e RAY-CHAUDHURI, D. K. *On a class of error correcting binary group codes*, Information and Control, vol. 3, pág. 68-79, março 1960.

[18] BOSE, R. C. e RAY-CHAUDHURI, D. K. *Further results on error correcting binary group codes*, Information and Control, vol. 3, pág. 279-290, setembro 1960.

[19] HOCQUENGHEM, A. *Codes correcteurs d'erreurs*, Chiffres, vol. 2, pág. 147-156, 1959.

[20] CAIN, J. B., CLARK, G. C. e J. M. GEIST, *Punctured convolutional codes of rate (n – 1)/n and simplified maximum likelihood decoding*, IEEE Transactions on Information Theory, vol. IT-25, pág. 97-100, janeiro 1979.

[21] HAGENAUER, J. *Rate Compatible puctured convolutional codes (RCPC codes) and their applicatons*, IEEE Transactions on Communication, vol. 36, nº 4, pág. 389-400, abril 1988.

[22] WICKER, S. B. *Error Control Systems for Digital Communication and Storage*, Prentice Hall, NJ, 1995.

[23] J. Hagenauer, “The performance of rate-compatible punctured convolutional codes for digital móbile radio”, IEEE Transactions on Communications, vol. 38, nº 7, pág. 966-980, julho 1990.

[24] HAGENAUER, J., SESHADRI, N. e SUNDBERG, C.W. *The performance of rate-compatible punctured convolutional codes for digital mobile radio*, IEEE Transactions on Communications, vol. 38, nº 7, pág. 966-980, julho 1990.

[25] D. W. Redmill e N. Kingsbury, *The EREC: An error resilient technique for coding variable-length blocks of data*, IEEE Transactions on Image Processing, vol. 5, nº 4, pág. 565-574, abril 1996.

[26] YAP, C. W., NGAN, K. N. e LIYANAPATHIRANA, R. *Error protection scheme for the transmission of H.263: coded video over mobile radio channels*, Proceedings of the SPIE: Visual Communications and Image Processing, fevereiro 1997, pág. 1241-1249.

[27] CHENG, N. T. e KINGSBURY, N. *The ERPC: An efficient error-resilient technique for encoding positional information or sparse data*, IEEE Transactions on Communications, vol. 40, nº 1, pág. 140-148, Janeiro 1992.

[28] SAID, A. e PEARLMAN, W. A. *A new fast and efficient image codec based on set partitioning in hierarchical trees*, IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, nº 3, pág. 243-250, junho 1996.

[29] ELIAS, P. *Coding for noise channels*, IRE Convention Record, pág. 37-47, 1955.