

Recomendação Para Normalização de Sistemas Espaciais de Dados

COMPRESSÃO DE DADOS DE IMAGENS

CCSDS 122.0-B-1

Livro Azul

Novembro 2005

Nota:
Esta edição atual inclui todas as atualizações contidas no Corrigendum Técnico 1 ("Technical Corrigendum 1"), de **Julho de 2006**.

Nota:
Esta edição atual inclui todas as atualizações contidas no Corrigendum Técnico 2 ("Technical Corrigendum 2"), de **Julho de 2008**.

N.T. – Este documento, em sua versão inicial, originou-se da tradução do documento original CCSDS 122.0-B-1, de Novembro de 2005, escrito em língua Inglesa. O conteúdo e o formato do documento original foram preservados, sempre que possível. Todo o trabalho de tradução e de edição deste documento foi realizado pela Comissão de Estudo CE 08:001.06, dedicada a Sistemas Espaciais de Transferência de Dados e de Informações, pela ABNT – Associação Brasileira de Normas Técnicas. © CE 08:001.06/ABNT, INPE/MCT, São José dos Campos, SP, Brasil - Janeiro 2009.

COLABORADORES DESTA EDIÇÃO

Por ocasião em que foi realizada a editoração deste documento interno participaram deste trabalho:

1) Os seguintes MEMBROS da Comissão de Estudo em Sistemas Espaciais de Transferência de Dados e de Informação – ABNT/CB08/SC01/CE06:

Eduardo W. Bergamini	Coordenador Editorial da Comissão	INPE/MCT
Elvis Paulo da Silva	Membro	EMBRAER
Gerald J. F. Banon	Membro	INPE/MCT
José Bastos Mollica	Membro	ANATEL
Kátia Pádua Thomaz	Membro	ECI/UFMG
Marília Vidigal da Costa Souza	Membro	EMBRAER
Marco Antonio Grivet M. Maia	Membro	CETUC/PUC-RIO
Reginaldo Palazzo Júnior	Membro	DT/FEEC/UNICAMP
Sérgio Costa	Membro	AEL/IAE/CTA
Valéria Cristina M. N. Leite	Membro	AEL/IAE/CTA

2) As seguintes SECRETÁRIAS da coordenação desta Comissão de Estudo, que contribuíram de forma decisiva no extenso trabalho editorial e de correspondência associada a este documento:

Síntique Rodrigues dos Santos	Secretária	SME/TEC/INPE
Karen Fernanda dos Santos Pinho	Secretária	SME/TEC/INPE

São José dos Campos, Abril de 2008.

AUTORIDADE

Edição:	Norma Recomendada
Data:	Julho, 2006
Local:	Washington D.C, USA

Este documento foi aprovado para publicação pelo Conselho de Gerenciamento do Comitê Consultivo para Sistemas Espaciais de Dados (CCSDS) e representa o consenso de entendimento dos especialistas dos painéis técnicos das Agências Membro do CCSDS. O procedimento para revisão e autorização dos Relatórios CCSDS está detalhado no *Manual de Procedimentos do Comitê Consultivo para Sistemas Espaciais de Dados*.

Este documento é publicado e mantido por:

Secretariado CCSDS
Escritório de Comunicações Espaciais (Código M-3)
Administração Nacional de Aeronáutica e Espaço
Washington, DC 20546, USA

NOTA-DO-TRADUTOR

Este documento representa o resultado da tradução de um documento original, escrito em língua Inglesa. O conteúdo e o formato do documento original foram preservados neste documento, sempre que possível. O conteúdo residual em língua Inglesa que este documento possui é resultante do texto do documento original. Esse resíduo em língua Inglesa foi mantido por ter sido esse recurso considerado pertinente, para os seus usuários. Este documento é também considerado de utilidade para pessoas não familiarizadas o suficiente com a língua Inglesa e que necessitam: consultar, conhecer e ou utilizar o documento original, escrito em língua Inglesa, a partir do qual foi realizada a tradução que resultou neste documento, presente. Primariamente, de forma equivalente ou correspondente, este documento constitui fonte de informação para o preparo de Normas ou de Documentos Normativos (Anexos, etc.) para o Brasil. A Comissão de Estudos CE 08:001.06, dedicada a Sistemas Espaciais de Dados e Informação, da Associação Brasileira de Normas Técnicas (ABNT), realizou o trabalho de tradução e de edição deste documento. Este trabalho faz parte do esforço de normalização espacial no Brasil, a cargo da Comissão CE 08:001.06/ABNT, dentro do seu escopo de especialização. O documento original, em língua Inglesa, cuja tradução deu origem a esta Norma, possui o título: "Image Data Compression", sob código CCSDS 122.0-B-1, 'Blue Book', de Novembro de 2005, produzido pelo 'Consultative Committee for Space Data Systems (CCSDS)'.

© ABNT/CB08/SC01/CE06, INPE/MCT, São José dos Campos, SP, Brasil – Outubro de 2007

DECLARAÇÃO DE INTENÇÃO

O Comitê Consultivo para Sistemas Espaciais de Dados (CCSDS) é uma organização oficialmente estabelecida e gerenciada pelas suas Agências espaciais membro. O Comitê se reúne periodicamente para tratar de problemas de sistemas de dados que são comuns a todos os seus participantes, e para formular soluções técnicas seguras para estes problemas. Pelo fato da participação no CCSDS ser plenamente voluntária, os resultados das ações do Comitê são definidos como sendo **Recomendações para Normalização** que não são consideradas de uso obrigatório para qualquer uma de suas Agências membro.

Esta **Recomendação para Normalização** é publicada e representada como resultado de consenso dos membros do CCSDS. O endosso a esta **Recomendação** é inteiramente voluntário. Este endosso, no entanto, implica nos seguintes entendimentos:

- Sempre que um membro estabelece uma **norma** relacionada com o CCSDS, esta **norma** estará em acordo com a **Recomendação para Normalização** pertinente ao caso. O estabelecimento de uma tal **norma** não exclui outras disposições que um membro possa desenvolver.
- Sempre que um membro estabelece uma **norma** relacionada com o CCSDS, o membro deverá prover aos outros membros do CCSDS as seguintes informações:
 - A **norma** propriamente dita.
 - Com antecipação, a data da sua capacitação operacional inicial.
 - Com antecipação, a duração operacional do seu respectivo serviço.
- Acordos específicos de serviço devem ser feitos via memorando de entendimento. Nenhuma **Recomendação para Normalização** ou qualquer **norma** a ela subsequente substitui um memorando de entendimento.

Em prazo não além de cinco anos após a sua publicação esta **Recomendação para Normalização** será revista pelo CCSDS para determinar se ela deverá: (1) Continuar efetiva, sem sofrer modificações; (2) Ser modificada para refletir o impacto de novas tecnologias, novos quesitos, ou novas orientações; ou (3) Ter a sua adoção descontinuada ou cancelada.

Nos casos em que uma nova versão da **Recomendação para Normalização** for editada, padrões e implementações existentes de membros relacionados com o CCSDS não passarão a ser consideradas não compatíveis com o CCSDS, por este motivo. Cabe a cada membro a responsabilidade de determinar quando tais padrões ou implementações devem ser modificadas ou atualizadas. Cada membro é fortemente encorajado orientar o seu planejamento para adoção dos novos padrões e implementações em acordo com uma versão posterior da Recomendação para Normalização.

PREFÁCIO

Dentro de um processo de evolução normal, é esperando que a expansão, descarte, ou modificação deste documento possa ocorrer. Esta Recomendação para Normalização esta, portanto, sujeita ao gerenciamento de documentos e aos procedimentos de controle de modificação do CCSDS, tal como estão definidos no *Manual de Procedimentos do Comitê Consultivo para Sistemas Espaciais de Dados*. As versões atuais de documento CCSDS são mantidas no sítio Web do CCSDS:

<http://www.ccsds.org/>

Questões relacionadas com o conteúdo ou status deste documento devem ser endereçadas ao Secretariado do CCSDS no endereço indicado na página i.

Por ocasião desta publicação, as agências membro e observadoras ativas do CCSDS eram:

Agências Membro

- Agenzia Spaziale Italiana (ASI)/Italy.
- British National Space Centre (BNSC)/United Kingdom.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.
- Russian Federal Space Agency (FSA)/Russian Federation.

Agências Observadoras

- Austrian Space Agency (ASA)/Austria.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- Comando-Geral de Tecnologia Aeroespacial (CTA)/Brazil.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Communications Research Laboratory (CRL)/Japan.
- Danish Space Research Institute (DSRI)/Denmark.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Federal Service of Scientific, Technical & Cultural Affairs (FSST&CA)/Belgium.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space and Astronautical Science (ISAS)/Japan.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Korea Aerospace Research Institute (KARI)/Korea.
- MIKOMTEK: CSIR (CSIR)/Republic of South Africa.
- Ministry of Communications (MOC)/Israel.
- National Oceanic & Atmospheric Administration (NOAA)/USA.
- National Space Program Office (NSPO)/Taipei.
- Space & Upper Atmosphere Research Commission/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- United States Geological Survey (USGS)/USA.

CONTROLE DO DOCUMENTO

Documento	Título	Data	Situação/Observações
CCSDS 122.0-B-1 Cor. 2	Compressão de Dados de Imagens Recomendação para Normalização, Edição 1 (Image Data Compression, Recommended Standard, Issue 1)	Julho 2008	Edição original

SUMÁRIO

<u>Seção</u>	<u>Página</u>
1 INTRODUÇÃO.....	1-1
1.1 PROPÓSITO.....	1-1
1.2 ESCOPO.....	1-1
1.3 APLICABILIDADE.....	1-1
1.4 CRITÉRIO.....	1-1
1.5 CONVENÇÃO.....	1-2
1.6 DEFINIÇÕES.....	1-2
1.7 NOMENCLATURA.....	1-3
1.8 REFERÊNCIAS.....	1-3
2 VISÃO GERAL.....	2-1
2.1 GERAL.....	2-1
2.2 ENTREGA DOS DADOS.....	2-2
3 DESCRIÇÃO DA TRANSFORMADA WAVELET DISCRETA.....	3-1
3.1 VISÃO GERAL.....	3-1
3.2 QUADRO DE IMAGEM.....	3-2
3.3 DWT UNIDIMENSIONAL DE UM NÍVEL.....	3-3
3.4 DWT INVERSA.....	3-5
3.5 DWT BIDIMENSIONAL DE UM NÍVEL.....	3-7
3.6 INVERSO DA DWT BIDIMENSIONAL.....	3-8
3.7 DWT BIDIMENSIONAL MULTINÍVEL.....	3-8
3.8 DWT 2-D MULTINÍVEL INVERSA.....	3-9
3.9 PESOS DA SUBBANDA.....	3-10
4 O CODIFICADOR DE PLANO DE BITS (BPE).....	4-1
4.1 RESUMO.....	4-1
4.2 CABEÇALHO DO SEGMENTO.....	4-5
4.3 CODIFICAÇÃO INICIAL DOS COEFICIENTES DC.....	4-18
4.4 ESPECIFICANDO A PROFUNDIDADE DE BITS AC EM CADA BLOCO...	4-24
4.5 CODIFICAÇÃO EM PLANO DE BITS.....	4-25

<u>Seção</u>	<u>Página</u>
5 SEGURANÇA	5-1
5.1 INTRODUÇÃO	5-1
5.2 CONSIDERAÇÕES DE SEGURANÇA COM RELAÇÃO A ESTA RECOMENDAÇÃO DE NORMA	5-1
5.3 AMEAÇAS POTENCIAIS E CENÁRIOS DE ATAQUE	5-2
5.4 CONSEQÜÊNCIAS DO NÃO EMPREGO DE SEGURANÇA PARA A TECNOLOGIA.....	5-2
ANEXO A GLOSSÁRIO DE ACRÔNIMOS E TERMOS (Informativo)	A-1
ANEXO B REFERENCIAS INFORMATIVAS (Informativo).....	B-1
ANEXO C SÍMBOLOS USADOS NOS ESTÁGIOS DE CODIFICAÇÃO (Informativo)	C-1

Figuras

2-1 Esquema Geral do Codificador.....	2-1
3-1 DWT 2-d (Um Nível)	3-7
3-2 Decomposição DWT 2-d em Três-Níveis de uma Imagem.....	3-9
4-1 Esquema da Transformada Wavelet de uma Imagem	4-1
4-2 Visão Geral da Estrutura de um Segmento Codificado	4-4
4-3 Visão Geral da Estrutura do Cabeçalho do Segmento Quando Todas as Partes Estão Incluídas	4-5
4-4 Parte 1 do Cabeçalho do Segmento Quando Parte 1B está Inclusa	4-7
4-5 Parte 2 do Cabeçalho do Segmento	4-10
4-6 Parte 3 do Cabeçalho do Segmento	4-13
4-7 Parte 4 do Cabeçalho do Segmento	4-16
4-8 Formato dos Dados Codificado para um Conjunto (ou, <i>Gaggle</i>) Quando a Não Codificado (Opção Uncoded) é Seleccionada	4-22
4-9 Formato dos Dados Codificado para um Conjunto (ou, <i>Gaggle</i>) Quando a Opção Codificado (ou, <i>Coded</i>) é Seleccionada	4-22
4-10 Estrutura de Plano de Bits Codificada para um Segmento	4-25

Tabela

3-1 Coeficientes do Filtro de Análise para o Filtro 9/7	3-3
3-2 Coeficientes do Filtro de Síntese para o Filtro 9/7	3-5
3-3 Pesos da Subbanda Padrão para DWT de Inteiros 9/7	3-10
4-1 Coordenadas Internas da Subbanda para Coeficientes em uma Única Família.....	4-2
4-2 Subbanda de Origem para Coeficientes AC	4-2
4-3 Sumário da Funcionalidade do Cabeçalho de Segmento.....	4-6
4-4 Conteúdo da Parte 1 do Cabeçalho do Segmento	4-7
4-5 Conteúdo da Parte 2 do Cabeçalho do Segmento	4-10

SUMÁRIO (continuação)

<u>Tabela</u>	<u>Página</u>
4-6 Conteúdo da Parte 3 do Cabeçalho do Segmento	4-13
4-7 Conteúdo da Parte 4 do Cabeçalho de Segmento	4-15
4-8 Quantização dos Coeficientes DC	4-18
4-9 Identificadores da Opção de Código	4-21
4-10 Regras de Seleção da Opção de Código	4-23
4-11 Sumário de Comprimentos Máximos da Palavra e Valores Impossíveis da Palavra	4-29
4-12 Mapeamento de Inteiros para Palavras de Dois Bits	4-30
4-13 Mapeamento de Inteiros para Palavras de Três Bits.....	4-30
4-14 Mapeamento de Inteiros para Palavras de Quatro Bits.....	4-30
4-15 Opções de Código de Comprimento Variável para Palavras de Dois Bits.....	4-31
4-16 Opções de Código de Comprimento Variável para Palavras de Três Bits	4-32
4-17 Opções de Código de Comprimento Variável para Palavras de Quatro Bits	4-32
4-18 Identificando as Opções de Código de Comprimento Variável	4-33
C-1 Símbolos Usados nos Estágios de Codificação na Secção 4	C-1

1 INTRODUÇÃO

1.1 PROPÓSITO

A finalidade deste documento é para estabelecer uma Recomendação para Normalização para algoritmos de compressão de dados aplicados a dados de imagem espaciais digitais bidimensionais obtidos de instrumentos da carga útil e para especificar como estes dados comprimido serão formatados em segmentos para permitir a descompressão no receptor final.

A codificação fonte, para compressão de dados, é um método utilizado em sistemas de dados para reduzir o volume de dados digitais a fim de alcançar benefícios em áreas incluindo, mas não limitadas a,

- a) redução da largura de banda do canal de transmissão;
- b) redução do requisito de armazenamento temporário (ou, *buffering*) e memória;
- c) redução de tempo de transmissão de dados em uma dada taxa.

1.2 ESCOPO

As características dos dados são especificadas somente na medida necessária para assegurar a capacidade de apoio às multimissões. A especificação não se propõe a quantificar a redução relativa à largura de banda, aos méritos de cada enfoque discutido, ou aos requisitos de projeto para codificadores e decodificadores associados. Algumas informações de desempenho estão incluídas na referência [B1].

Esta Recomendação para Normalização trata da compressão de dados de imagem, a qual é aplicável a uma grande variedade de dados digitais espaciais, onde a especificação é para uma redução dos dados com escalabilidade, incluindo a opção para uso de compressão com perdas, a qual permite alguma perda de fidelidade no processo compressão e descompressão de dados. A referência [B1] fornece um esboço para implementação.

1.3 APLICABILIDADE

Esta Recomendação para Normalização destina-se a aplicações de compressão dos dados de imagem de missões espaciais prevendo apoio à telemetria por pacotes. Além disso, serve como um guia para o desenvolvimento de padrões compatíveis com o CCSDS neste campo, baseado na boa prática da engenharia.

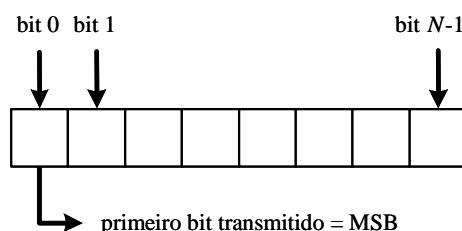
1.4 CRITÉRIO

O conceito e a base lógica para o algoritmo de Compressão de Dados de Imagem descrito aqui podem ser encontrados na referência [B1].

1.5 CONVENÇÃO

Na especificação de cabeçalhos (ou, *headers*) para compressão de dados, a seguinte convenção é usada para identificar cada unidade de informação binária ou ‘bit’ em uma palavra de N -bits. O primeiro bit da palavra a ser transmitido (isto é, o mais alinhado à esquerda quando se desenha uma figura) é definido como ‘Bit 0’; o bit seguinte é definido como ‘Bit 1’ e assim por diante até o ‘Bit $N-1$ ’. Quando a palavra é utilizada para expressar um valor binário não sinalizado (tal como em um contador), o Bit Mais Significativo (MSB, *Most Significant Bit*) corresponderá à potência de dois mais alta, isto é, 2^{N-1} .

NOTA – Esta numeração do bit aplica-se aos cabeçalhos de segmento descritos em 4.2. Uma numeração diferente é usada para indexar bits de magnitude nos coeficientes da Transformada Wavelet Discreta (DWT, ou *Discrete Wavelet Transform*), como descrito em 4.1.



De acordo com as práticas modernas de comunicação de dados, as palavras de dados de uma espaçonave são freqüentemente agrupadas em ‘palavras’ de oito bits que se ajustam a convenção acima. Ao longo desta Recomendação de Norma, a seguinte nomenclatura será utilizada para descrever este agrupamento:

Palavra de 8 bits = ‘Byte’

1.6 DEFINIÇÕES

Neste documento, para qualquer número real x , o maior inteiro n , tal que, $n \leq x$ deve ser indicado por

$$n = \lfloor x \rfloor,$$

e correspondentemente, o menor inteiro n , tal que, $n \geq x$ por

$$n = \lceil x \rceil.$$

O módulo m de um número M com relação a um divisor n é indicado por

$$m = M \bmod n.$$

A afirmação de que um valor M é codificado mod n significa que o número

$$m = M \bmod n$$

é codificado ao invés de *M*.

1.7 NOMENCLATURA

As seguintes convenções aplicam-se em toda esta Recomendação para Normalização:

- a) as palavras ‘deve(m)’ (*shall*) e ‘precisa(m)’ (*must*) implicam uma especificação verificável e consistente;
- b) a palavra ‘deveria(m)’ (*should*) implica em uma especificação opcional, mas desejável;
- c) a palavra ‘pode(m)’ (*may*) implica em uma especificação opcional;
- d) as palavras ‘é’ (*is*), ‘são’ (*are*), e ‘será’ (*will*) implicam em afirmações de fato.

Nas seções normativas deste documento (seções 3 e 4), o texto informativo (isto é, não normativo) está diferenciado do texto normativo pela seguinte convenção:

- a) notas introduzem afirmações informativas breves;
- b) os seguintes títulos introduzem um ou mais parágrafos de texto informativo:
 - Visão Geral (ou, *Overview*);
 - Critério.

1.8 REFERÊNCIAS

Esta Recomendação para Normalização não contém referências normativas. Referências informativas estão listados no anexo B.

2 VISÃO GERAL

2.1 GERAL

Esta Recomendação para Normalização define um algoritmo específico para compressão de dados de imagem de carga útil (ou, *payload*) que tem aplicabilidade difundida para vários tipos de instrumentos. Esta Recomendação para Normalização não tenta explicar a teoria base de operação do algoritmo, a qual é parcialmente descrita em [B1].

Há duas classes de métodos de compressão de dados: sem perda (*lossless*) e com perda (*lossy*). Na compressão sem perdas, os dados originais podem ser reproduzidos exatamente, enquanto que na compressão com perdas, a quantização ou outras aproximações usadas no processo de compressão resultam na incapacidade para reproduzir o conjunto de dados originais sem nenhuma distorção. O aumento da quantidade de informação dos dados sujeitos a compressão sem perdas resulta em um grande volume de dados para um certo conjunto de dados fonte.

A técnica da compressão descrita nesta Recomendação para Normalização pode ser usada para produzir ambas, compressão sem perda e com perda. Uma técnica alternativa de compressão sem perda, a qual tem a mais baixa complexidade, mas que não é especificamente adaptada para imagens, foi anteriormente adotada pelo CCSDS (ver a referência [B2]). O padrão JPEG2000 é outro compressor de imagem (ver a referência [B6]). A presente Recomendação de Norma difere do padrão JPEG2000 em vários aspectos:

- a) ela especificamente visa instrumentos de taxa alta usados a bordo de espaçonaves;
- b) um compromisso foi realizado entre o desempenho da compressão e a complexidade com ênfase específica em aplicações de espaçonaves (cf. a);
- c) a baixa complexidade mais desta recomendação auxilia na implementação rápida e de baixa potência (ou, *low-power*) do hardware;
- d) ela tem um conjunto limitado de opções, auxiliando na sua aplicação bem sucedida sem o conhecimento profundo do algoritmo.

O compressor consiste de duas partes funcionais, descritas na figura 2-1, um módulo de Transformada Wavelet Discreta que realiza a decorrelação, descrita na seção 3, e um Codificador de Plano de Bits, o qual codifica os dados decorrelacionados, descrito na seção 4.

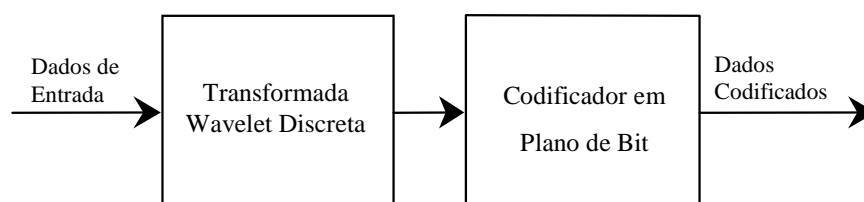


Figura 2-1: Esquema Geral do Codificador

Esta Recomendação para Normalização admite ambos os formatos de entrada baseados em quadros (ou, *frame-based*) produzidos, por exemplo, pelos agrupamentos CCD (chamado quadros da imagem) e formatos baseados em faixas (ou, *strip-based*) produzidos por sensores tipo *push-broom* (chamado faixas ou tiras (ou, *stripmaps*) da imagem). Um range dinâmico de pixel da imagem de até 16 bits é admitido. O algoritmo especificado admite uma implementação eficaz de memória para o procedimento de compressão, o qual não requer quadros intermediários grandes para armazenamento temporário (ver a referência [B1]).

2.2 ENTREGA DOS DADOS

A seqüência de bits (ou, *bitstream*), correspondente a um quadro ou faixa da imagem, consiste em um segmento único ou uma seqüência de segmentos. Cada segmento consiste em um cabeçalho seguido por um campo de dados codificado. Um segmento pode ter comprimento fixo ou comprimento variável dependendo do modo operacional selecionado. Os valores dos parâmetros do algoritmo usados na codificação podem ser especificados usando o cabeçalho definido na seção 4.

Os efeitos do erro de um único bit podem se propagar e corromper dados reconstruídos até o fim do segmento afetado (veja a referência [B1]). Portanto, medidas deveriam ser tomadas para minimizar a taxa de erros de bit em potencial no link de transmissão. O mecanismo de transporte para entrega da seqüência de bits (ou, *bitstream*) codificada deve possuir, no caso de um erro de bit, a habilidade para transferir o cabeçalho do próximo segmento.

No caso a seqüência de bits codificado a ser transmitido sobre um link espacial CCSDS, vários protocolos podem ser usados para transferir a seqüência de segmentos. Esses protocolos estão especificados nas referências:

- referência [B3], Protocolo do Pacote Espacial;
- referência [B4], Protocolo de Distribuição de Arquivo CCSDS (CFDP);
- referência [B5], serviço de pacotes ou serviço de seqüência de bits s (ou, *bitstream*) como fornecido pelo Protocolo de Link de Dados Espaciais AOS.

A interface para esses protocolos de link espacial e questões relacionadas são discutidas na referência [B1].

3 DESCRIÇÃO DA TRANSFORMADA WAVELET DISCRETA

3.1 VISÃO GERAL

Esta Recomendação para Normalização, para o módulo de descorrelação, faz uso da Transformada Wavelet Discreta (DWT) separável, bidimensional (2-d), de três níveis, com nove e sete coeficientes (ou, *taps*) para filtros passa - baixa e passa - alta, respectivamente. Tal transformada é produzida pela aplicação repetida da DWT unidimensional (1-d) descrita em 3.3, 3.5, e 3.7. Duas wavelets 1-d específicas são detalhadas nesta Recomendação para Normalização: a DWT biortogonal 9/7, referenciada como ‘DWT de ponto flutuante 9/7’ (ou, 9/7 *Float DWT*) ou simplesmente ‘DWT’, e uma aproximação não-linear inteira para essa transformada, chamada de ‘DWT de Inteiros 9/7’ (ou, 9/7 *Integer DWT*) ou simplesmente ‘DWT de Inteiros’.

A definição de inversa funcional de cada DWT é incluída em 3.4, 3.6, e 3.8.

Enquanto a DWT de ponto flutuante, geralmente, exibe eficiência de compressão superior com relação a perdas, somente a DWT de inteiros suporta perfeitamente a compressão sem perdas.

Considera-se que a imagem utiliza píxeis de R -bit para representar valores inteiros sinalizados ou não sinalizados, onde $R \leq 16$.

A saída de valores da DWT 2-d de 3-níveis (ver 3.7) é convertida para valores inteiros apropriados antes da aplicação do Codificador de Plano de Bit (BPE – *Bit Plane Encoder*— veja a seção 4). Cada inteiro é representado utilizando uma palavra binária que consiste de um bit de sinal junto com vários bits de magnitude. O tamanho máximo da palavra necessário para armazenar cada inteiro depende de R .

No caso de DWT de ponto flutuante, os valores calculados no domínio wavelet são arredondados para os respectivos inteiros mais próximo antes da aplicação do BPE. Uma palavra de comprimento correspondente a $R+5$ bits é suficiente para armazenar esses valores inteiros.

No caso de DWT de inteiros, antes de aplicar o BPE, os valores calculados no domínio wavelet são multiplicados por pesos inteiros, os quais são uniformes em cada subbanda (ver 3.9). Uma palavra de comprimento correspondente a $R+4$ bits é suficiente para armazenar tais inteiros antes que os fatores de pesos sejam aplicados. Após operação de aplicação dos pesos, palavras de comprimentos maiores podem ser usadas para armazenar os coeficientes da wavelet em algumas implementações.

Esquemas de implementação e métodos de reconstrução de imagem não são parte desta Recomendação para Normalização, embora alguns enfoques sejam descritos na referência [B1]. Implementações de compressor podem diferir consideravelmente dependendo, por exemplo, se a implementação espera pela formação de um quadro inteiro antes do início da compressão.

3.2 QUADRO DE IMAGEM

3.2.1 Cálculo de ambos DWT de ponto flutuante e DWT de inteiros depende das dimensões, i.e., comprimento da linha e dimensão da coluna, do quadro da imagem a ser transformada.

NOTA – Essa consideração formal de tamanho do quadro não implica que uma implementação tenha que fornecer espaços de memória (ou, *buffers*) para os quadros completos.

3.2.2 Largura da Imagem deve ser comunicada ao decodificador utilizando os cabeçalhos do segmento definidos em 4.2. A altura da imagem é inferida indiretamente por uma bandeira (ou, *flag*) do cabeçalho indicando o último segmento de uma imagem, como descrito em 4.2.

NOTA – Embora o compressor vá trabalhar com imagens pequenas, ele é projetado para explorar as correlações bidimensionais na imagem, e conseqüentemente, o desempenho da compressão pode ser pobre quando a altura ou a largura da imagem forem inferiores a 32.

3.2.3 A máxima largura da imagem (número de colunas) é o 2^{20} ; a largura mínima da imagem é 17. Não há restrição no número máximo de linhas que constituem um quadro da imagem; o número mínimo de linhas é 17.

3.2.4 A aplicação da DWT de ponto flutuante ou de inteiros para uma imagem requer que as dimensões da imagem sejam múltiplos inteiros de oito. Se a largura ou altura da imagem original não for um múltiplo de oito, a imagem será ‘preenchida’ (ou, *padded*) acrescentando o número mínimo de linhas e/ou colunas extra para produzir uma imagem com ambas as largura e altura divisível por oito:

- a) quando colunas são adicionadas para o preenchimento (ou, *padding*),
 - 1) colunas devem ser acrescentadas no canto direito da imagem, i.e., no canto contendo o maior índice de pixel na direção horizontal,
 - 2) valores de pixel de colunas acrescentadas devem ser cópias do valor da coluna da imagem mais a direita;
- b) quando linhas são acrescentadas para o preenchimento,
 - 1) linhas devem ser acrescentadas no canto de baixo da imagem, i.e., no canto contendo o índice de pixel mais alto na direção vertical,
 - 2) linhas acrescentadas devem ser cópias idênticas da última linha da imagem.

3.2.5 Linhas e/ou colunas de dados acrescentadas para preenchimento devem ser descartados depois da descompressão.

3.3 DWT UNIDIMENSIONAL DE UM NÍVEL

3.3.1 TRANSFORMADA DE PONTO FLUTUANTE 9/7

3.3.1.1 Coeficientes do Filtro de Análise

A DWT de ponto flutuante 9/7 usa dois conjuntos de coeficientes do filtro de análise:

$$\begin{aligned} & \{h_{-4}, h_{-3}, h_{-2}, h_{-1}, h_0, h_1, h_2, h_3, h_4\} \\ & \{g_{-3}, g_{-2}, g_{-1}, g_0, g_1, g_2, g_3\} \end{aligned} \quad (1)$$

Os valores numéricos dos coeficientes do filtro de análise estão especificados na tabela 3-1.

Tabela 3-1: Coeficientes do Filtro de Análise para o Filtro 9/7

Coeficientes do Filtro de Análise		
i	Filtro Passa - baixa h_i	Filtro Passa - alta g_i
0	0.852698679009	-0.788485616406
± 1	0.377402855613	0.418092273222
± 2	-0.110624404418	0.040689417609
± 3	-0.023849465020	-0.064538882629
± 4	0.037828455507	

NOTAS

- Os coeficientes do filtro são obtidos como aproximações reais de números irracionais. A precisão aritmética usada no cálculo da DWT é uma questão de implementação e não parte desta Recomendação para Normalização. O impacto da precisão aritmética reduzida dos coeficientes do filtro no desempenho da compressão é mitigado pelo fato de que os coeficientes da wavelet são aproximados para o inteiro mais próximo antes da codificação. Veja a referência [B1].
- A DWT 1-d é definida para um número par de amostras de entrada. O procedimento de expansão da imagem descrito em 3.2 assegura que o número de amostras, ao qual a DWT 1-d é aplicada é par em todos níveis.

3.3.1.2 Operações do Filtro de Análise

Para $N > 2$,

$$\{x_0, x_1, x_2, \dots, x_{2N-1}\} \quad (2)$$

denota um sinal unidimensional consistido de $2N$ amostras. A DWT unidimensional de ponto flutuante é definida pelo seguinte par de operações do filtro de *análise*:

$$C_j = \sum_{n=-4}^4 h_n x_{2j+n}; \quad D_j = \sum_{n=-3}^3 g_n x_{2j+1+n}; \quad j = 0, 1, \dots, N-1 \quad (3)$$

Os valores C_j, D_j são chamados de *coeficientes* da transformada wavelet.

NOTA – Na equação 3 os coeficientes (ou, *taps*) do filtro são escolhidos, tal que, C_j e D_j representem saídas passa-baixa e passa-alta, respectivamente. Isso é C_j é uma versão aproximada (passa-baixa) do sinal original, enquanto que D_j contém informações de alta frequência.

No início (limite esquerdo) do sinal, a equação 3 requer valores de sinal com índices negativos, e o final (limite direito) do sinal, requer valores de sinal com índices que excedam $2N-1$. Esses valores são obtidos a partir da seguinte extensão simétrica espelhada do sinal:

$$x_m = x_{-m} \text{ for } m < 0; \quad x_{2N-1+m} = x_{2N-1-m} \text{ for } m > 0 \quad (4)$$

3.3.2 TRANSFORMADA DE INTEIROS 9/7

NOTA – Uma transformada alternativa nesta Recomendação para Normalização é uma aproximação não-linear da DWT 9/7. A não-linearidade é introduzida como o resultado de operações de arredondamento (ou, *round-off*) usadas para gerar saídas inteiras na etapa de decorrelação. A transformação não-linear pode ser usada para obter compressão sem perdas.

Como a DWT de ponto flutuante definida na equação 3, a DWT de inteiros 1-d, um nível, mapeia um vetor de sinal (equação 2) para dois conjuntos de coeficientes da wavelet, um conjunto passa-alta, D_j , e um conjunto passa-baixa, C_j , de acordo com as equações 5 e 5. Filtros de borda (ou, *boundary filters*) especiais são necessários em cada extremidade do sinal, e levam às fórmulas apresentadas nas equações 5 e 6 para $j=0, j=N-2$, e $j=N-1$.

Equações 5 **Erro! Fonte de referência não encontrada.** e 5 definem a transformada de inteiros que deve ser usada com esta Recomendação de Norma. Dados os valores de entrada de x_i , os valores de D_j , na equação 5, devem ser calculados primeiro e usados, posteriormente, para calcular os valores de C_j na equação 5.

$$\begin{aligned} D_0 &= x_1 - \left[\frac{9}{16}(x_0 + x_2) - \frac{1}{16}(x_2 + x_4) + \frac{1}{2} \right] \\ D_j &= x_{2j+1} - \left[\frac{9}{16}(x_{2j} + x_{2j+2}) - \frac{1}{16}(x_{2j-2} + x_{2j+4}) + \frac{1}{2} \right] \quad \text{for } j = 1, \dots, N-3 \\ D_{N-2} &= x_{2N-3} - \left[\frac{9}{16}(x_{2N-4} + x_{2N-2}) - \frac{1}{16}(x_{2N-6} + x_{2N-2}) + \frac{1}{2} \right] \\ D_{N-1} &= x_{2N-1} - \left[\frac{9}{8}x_{2N-2} - \frac{1}{8}x_{2N-4} + \frac{1}{2} \right] \end{aligned} \quad (5)$$

$$C_0 = x_0 - \left[-\frac{D_0}{2} + \frac{1}{2} \right]$$

$$C_j = x_{2j} - \left[-\frac{D_{j-1} + D_j}{4} + \frac{1}{2} \right] \quad \text{for } j = 1, \dots, N-1$$
(5)

NOTA – Devido ao procedimento de arredondamento, a transformação recomendada é, a rigor, não-linear e assim não é realmente uma DWT. No entanto, uma transformação inversa (não-linear) perfeita existe (ver 3.4.2).

3.4 DWT INVERSA

3.4.1 TRANSFORMADA DE PONTO FLUTUANTE 9/7 INVERSA

3.4.1.1 Coeficientes do Filtro de Síntese

A DWT de ponto flutuante 9/7 inversa usa dois conjuntos de coeficientes de filtro de síntese:

$$\{q_{-3}, q_{-2}, q_{-1}, q_0, q_1, q_2, q_3\}$$

$$\{p_{-4}, p_{-3}, p_{-2}, p_{-1}, p_0, p_1, p_2, p_3, p_4\}$$
(6)

Os valores numéricos dos coeficientes do filtro de síntese são especificados na tabela 3-2.

Tabela 3-2: Coeficientes do Filtro de Síntese para o Filtro 9/7

Coeficientes do Filtro de Síntese		
<i>i</i>	Filtro Passa-baixa q_i	Filtro Passa-alta p_i
0	0.788485616406	-0.852698679009
±1	0.418092273222	0.377402855613
±2	-0.040689417609	0.110624404418
±3	-0.064538882629	-0.023849465020
±4		-0.037828455507

3.4.1.2 Operações do Filtro de Síntese

A DWT unidimensional deve ser invertida por meio de um par de operações de filtragem de *síntese*:

$$\left. \begin{aligned} x_{2j} &= \sum_{n=-1}^1 q_{2n} C_{j+n} + \sum_{n=-2}^1 p_{2n+1} D_{j+n} \\ x_{2j+1} &= \sum_{n=-1}^2 q_{2n-1} C_{j+n} + \sum_{n=-2}^2 p_{2n} D_{j+n} \end{aligned} \right\} \quad (j = 0, 1, \dots, N-1) \quad (7)$$

Para estimação correta da equação 7 nas bordas (ou, *boundaries*), os sinais dos coeficientes da wavelet C_j, D_j devem ser estendidos como segue:

$$\begin{aligned} D_m &= D_{-m-1} \quad \text{for } m < 0; & D_{N-1+m} &= D_{N-1-m} \quad \text{for } m > 0 \\ C_m &= C_{-m} \quad \text{for } m < 0; & C_{N-1+m} &= C_{N-m} \quad \text{for } m > 0 \end{aligned} \quad (8)$$

3.4.2 TRANSFORMADA DE INTEIROS INVERSA

Como a DWT de ponto flutuante inversa definida na equação 7, a DWT de inteiros inversa mapeia dois conjuntos de coeficientes wavelet, um conjunto passa-baixa, C_j , e um conjunto passa-alta, D_j , e retorna um vetor x_j de acordo com as equações 9 e 10. Filtros de borda (ou, *boundary filter*) especiais são necessários em cada extremidade dos conjuntos de dados, levando às fórmulas apresentadas nas equações 10 e 11 para $j=0, j=1, j=2N-3$, e $j=2N-1$.

$$\begin{aligned} x_0 &= C_0 + \left\lfloor -\frac{D_0}{2} + \frac{1}{2} \right\rfloor \\ x_{2j} &= C_j + \left\lfloor -\frac{D_{j-1} + D_j}{4} + \frac{1}{2} \right\rfloor \quad \text{for } j = 1, \dots, N-1 \end{aligned} \quad (9)$$

$$\begin{aligned} x_1 &= D_0 + \left\lfloor \frac{9}{16}(x_0 + x_2) - \frac{1}{16}(x_2 + x_4) + \frac{1}{2} \right\rfloor \\ x_{2j+1} &= D_j + \left\lfloor \frac{9}{16}(x_{2j} + x_{2j+2}) - \frac{1}{16}(x_{2j-2} + x_{2j+4}) + \frac{1}{2} \right\rfloor \quad \text{for } j = 1, \dots, N-3 \\ x_{2N-3} &= D_{N-2} + \left\lfloor \frac{9}{16}(x_{2N-4} + x_{2N-2}) - \frac{1}{16}(x_{2N-6} + x_{2N-2}) + \frac{1}{2} \right\rfloor \\ x_{2N-1} &= D_{N-1} + \left\lfloor \frac{9}{8}x_{2N-2} - \frac{1}{8}x_{2N-4} + \frac{1}{2} \right\rfloor \end{aligned} \quad (10)$$

Na equação 9, os valores de índice par do sinal $x_0, x_2, \dots, x_{2N-2}$ devem ser calculados primeiro, a partir dos coeficientes da DWT, via equação 10. Esses valores reconstruídos devem então ser utilizados para calcular os valores de índice ímpar do sinal $x_1, x_3, \dots, x_{2N-1}$ via equação 11.

3.5 DWT BIDIMENSIONAL DE UM NÍVEL

Descorrelação da imagem é efetuada usando uma DWT bidimensional, a qual é executada pela aplicação repetida da DWT unidimensional. Considerando a imagem como uma matriz de dados composta por linhas e colunas de vetores de sinal, uma DWT 2-d de um nível deve ser implementada na imagem através dos dois passos seguintes na seguinte ordem:

- a DWT 1-d deve ser calculada em cada linha da imagem, produzindo uma matriz (ou, *array*) intermediária de dados filtrados horizontalmente passa-baixa e horizontalmente passa-alta, cada metade tão larga quanto a matriz da imagem original, como ilustrado na figura 3-1(b);
- a DWT 1-d deve ser aplicada a cada coluna de ambas as matrizes intermediárias para produzir quatro subbandas, como mostrado na figura 3-1(c).

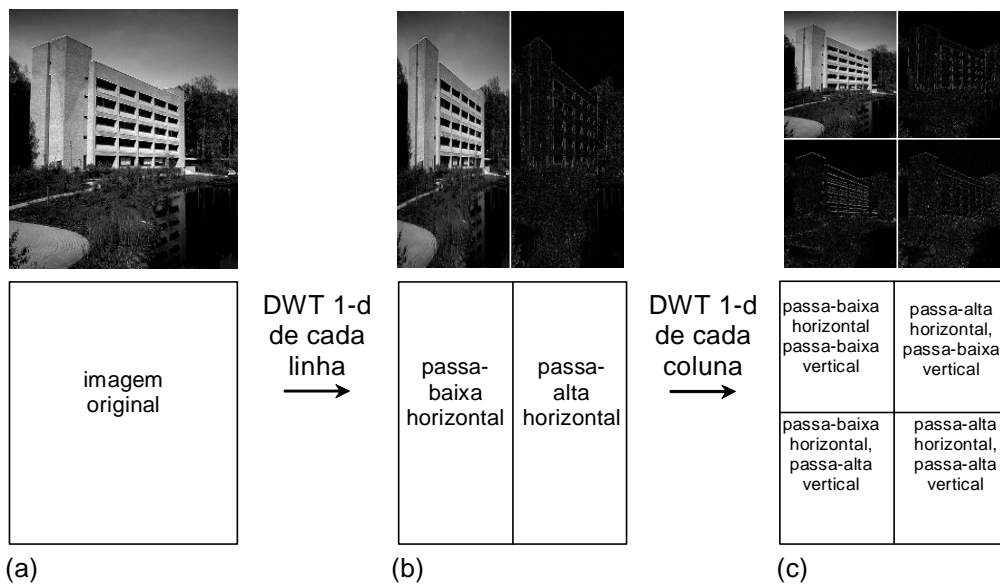


Figura 3-1: DWT 2-d (Um Nível)

Cada uma das quatro matrizes de subbanda obtidas possui metade da largura e metade da altura da matriz da imagem original. Em exemplos, essas subbandas são frequentemente mostradas organizadas como uma matriz (ou, *array*), a qual possui o mesmo tamanho da matriz da imagem original (ver o figura 3-1(C)). Começando na esquerda superior e prosseguindo em sentido horário na figura 3-1(C), as quatro subbandas são denominadas LL, HL, HH, LH.

3.6 INVERSO DA DWT BIDIMENSIONAL

A DWT 2-d de um-nível deve ser invertida pela aplicação repetida da transformada 1-d inversa nas colunas e nas linhas da matriz de dados transformada na ordem inversa da que as transformadas 1-d foram aplicadas:

- a) cada coluna deve ser invertida para produzir um conjunto de dados transformado intermediário:
 - 1) a DWT 1-d inversa deve ser aplicada às colunas das subbandas LL e LH para se obter a matriz passa-baixa horizontal intermediária da figura 3-1(b),
 - 2) a DWT 1-d inversa deve ser aplicada às colunas das subbandas HL e HH para obter a matriz passa-alta horizontal intermediária da figura 3-1(b);
- b) a DWT 1-d inversa deve ser aplicada às linhas das matrizes intermediária horizontal passa-baixa e horizontal alta-passa para recuperar a imagem original.

Este procedimento inverso deve ser usado para ambas as descompressões com e sem perdas.

3.7 DWT BIDIMENSIONAL MULTINÍVEL

Para aumentar a eficiência da compressão, a correlação restante na subbanda LL, depois da decomposição DWT 2-d, é explorada aplicando níveis adicionais de decomposição DWT para produzir uma DWT 2-d multinível. Isso produz a decomposição piramidal descrita na referência [B7].

Esta Recomendação para Normalização especifica três níveis de decomposição.

Em cada nível, a DWT 2-d descrita em 3.5 deve ser aplicada à subbanda LL produzida pelo nível de decomposição anterior.

NOTA – Figura 3-2 ilustra uma decomposição DWT 2-d de três níveis. Em cada nível de decomposição, a subbanda LL do nível anterior é decomposta usando uma DWT 2-d, e é substituída por quatro novas subbandas. Cada nova subbanda tem metade da largura e metade da altura da subbanda LL, a partir da qual foi calculada. Cada nível de decomposição adicional, assim, aumenta o número de subbandas em três, mas deixa inalterado o número total de coeficientes DWT usados para representar os dados de imagem. Após n níveis de decomposição DWT 2-d, o número total de subbandas é, portanto, $3n+1$. Seguindo a recomendação de uma decomposição em três níveis, dez subbandas são geradas. As subbandas são, tipicamente, mostradas organizadas para formar uma matriz com as mesmas dimensões da imagem original, como é feito na figura 3-2. Índices são adicionados a LL, HL, HH, e LH para indicar o nível de decomposição.

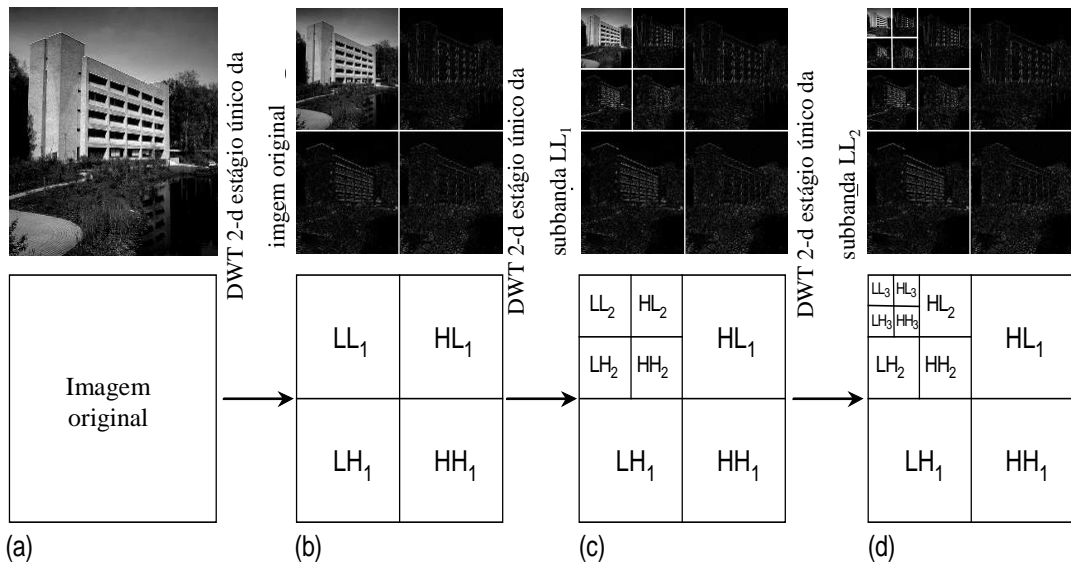


Figura 3-2: Decomposição DWT 2-d em três níveis de uma imagem

3.8 DWT 2-D MULTINÍVEL INVERSA

O processo de inversão da DWT multinível deve ser o seguinte:

- as quatro subbandas de nível mais alto, LL_3 , LH_3 , HL_3 , HH_3 , devem ser invertidas usando uma DWT 2-d de um nível (ou, *single level*) inversa para produzir a subbanda LL_2 , a qual, então, substitui as subbandas de nível mais alto na matriz de dados da transformação;
- as quatro subbandas LL_2 , LH_2 , HL_2 , HH_2 devem ser invertidas para produzir a subbanda LL_1 , a qual, outra vez, substitui as subbandas de nível mais alto na matriz dos dados da transformação;
- uma DWT inversa 2-d de um nível final deve ser aplicada às subbandas LL_1 , LH_1 , HL_1 , HH_1 para reproduzir a imagem original.

3.9 PESOS DA SUBBANDA

3.9.1 O Codificador de Plano de Bits (BPE) descrito na seção 4 deve ser usado para codificar as subbandas produzidas pela decomposição DWT 2-d.

3.9.2 Para transformação de inteiros, todos os coeficientes da subbanda devem ser multiplicados por seus respectivos fatores de peso antes da codificação. Esta multiplicação não é feita para transformação de ponto flutuante.

3.9.3 Pesos padrão para uso na multiplicação dos coeficientes da subbanda da DWT de inteiros são apresentados na tabela 3-3.

3.9.4 Um conjunto de pesos definidos pelo usuário pode ser adotado se o usuário deseja enfatizar os dados em algumas subbandas com relação a outras. Esta opção é sinalizada (ou, *flagged*) na Parte 4 do Cabeçalho do Segmento (4.2.5).

Tabela 3-3: Pesos da Subbanda Padrão para DWT de Inteiros 9/7

Subbanda	HH ₁	HL ₁ , LH ₁	HH ₂	HL ₂ , LH ₂	HH ₃	HL ₃ , LH ₃	LL ₃
Fator de peso	2 ⁰	2 ¹	2 ¹	2 ²	2 ²	2 ³	2 ³

NOTA – Os fatores de peso são escolhidos para serem potências de dois a fim de permitir que o escalonamento seja realizado usando operações de deslocamento de bit (ou, *bit-shift*). Por exemplo, a tabela 3-3 indica que depois de aplicar uma DWT 2-d de três níveis usando transformação de inteiros, cada um dos coeficientes resultantes da DWT na subbanda LH₂ deveria ser multiplicado por um fator de 2², o que pode ser realizado por duas operações de deslocamento de bit à esquerda.

4 O CODIFICADOR DE PLANO DE BITS (BPE)

4.1 RESUMO

Após a DWT, os coeficientes da wavelet ou são aproximados para o inteiro mais próximo (quando a transformada de ponto-flutuante é usada), ou escalonados usando os fatores de peso descrito em 3.9 (quando a transformada de inteiros é usada). Quando o escalonamento é realizado, um ou mais dos bits menos significativos, em muitas subbandas, são necessariamente zerados. O processo de BPE leva isso em conta, isto é, os bits que devem ser todos zero, devido ao processo de escalonamento, não são codificados no BPE. Na descrição nessa seção, os valores de coeficiente mencionados referem-se aos valores após qualquer operação de escalamento ou de aproximação. A notação $\text{BitShift}(\Gamma)$ é usada para indicar o número de bits menos significativos em cada coeficiente da subbanda Γ que são necessariamente zero como resultado da operação de escalamento da subbanda. Quando a DWT 9/7 de ponto flutuante é usada, as subbandas não são escalonadas, assim o BitShift igual-se a zero para cada subbanda.

O Codificador de Plano de Bits (BPE) processa coeficientes wavelet em grupos de 64 coeficientes chamados de *blocos*. Um exemplo de um bloco é ilustrado na figura 4-1 com píxeis sombreados. Um bloco corresponde a uma região localizada na imagem original.

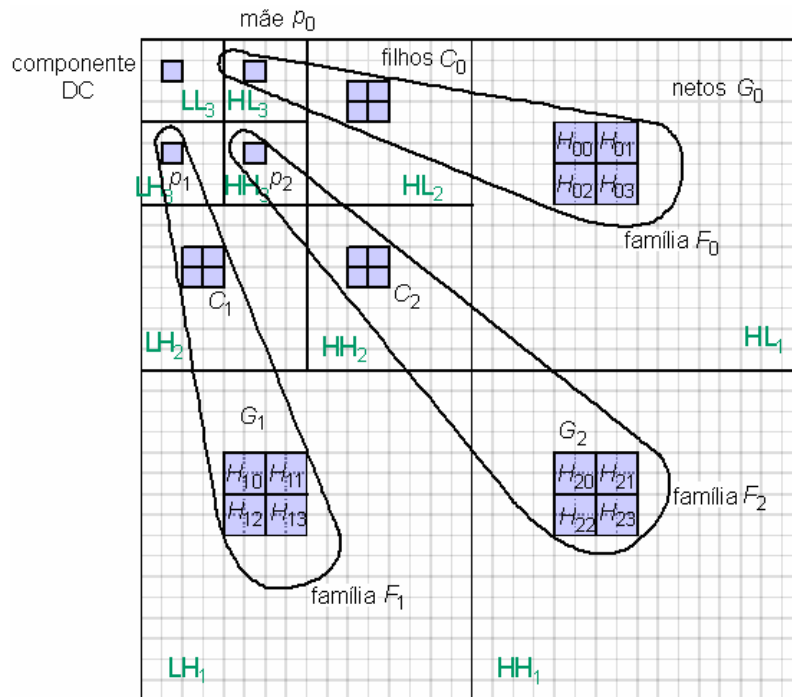


Figura 4-1: Esquema da Transformada Wavelet de uma Imagem

Informações relativas a um bloco de coeficientes são codificadas pelo BPE em conjunto. Um bloco consiste de um único coeficiente da subbanda LL_3 , chamado *coeficiente DC*, e 63 *coeficientes AC*. Os coeficientes AC em um bloco são arranjados em três *famílias*, F_0 , F_1 e F_2 . Figura 4-1 ilustra um único bloco de coeficientes e a estrutura da família.

Cada família F_i no bloco tem um coeficiente *mãe* (ou, *parent*), p_i , um conjunto C_i de quatro coeficientes *filhos*, e um conjunto G_i de dezesseis coeficientes *netos*. Os netos na família F_i são ainda divididos em grupos numerados $j=0,1,2,3$, denominado H_{ij} , como ilustrado na figura 4-1. Essa estrutura é usada para codificar em conjunto informações relativas aos grupos de coeficientes no bloco, como descrito em 4.5.

Um coeficiente wavelet é identificado por suas coordenadas dentro de sua subbanda. Assim, coordenadas (r, c) indicam o coeficiente wavelet na linha r , coluna c dentro da subbanda, com o pixel na esquerda superior da subbanda tendo coordenadas $(0,0)$.

O coeficiente DC para cada bloco é um coeficiente da subbanda LL_3 . As coordenadas para os outros coeficientes no bloco podem ser determinadas a partir das coordenadas do coeficiente DC. Para um bloco com coeficiente DC com coordenadas (r, c) dentro da subbanda LL_3 , a tabela 4-1 lista as coordenadas para os coeficientes AC, dentro de suas respectivas subbandas de origem. Tabela 4-2 mostra a subbanda de origem para cada tipo de coeficiente, determinado pela família e tipo de membro da família. A relação entre todos os coeficientes em um bloco é também ilustrada graficamente na figura 4-1. Blocos devem ser processados pelo Codificador de Plano de Bits consecutivamente na varredura tipo raster (ou, *raster scan*) na ordem em que seus coeficientes DC, correspondentes, ocorrerem em LL_3 : linha por linha, cada linha sendo processada da esquerda para direita.

Tabela 4-1: Coordenadas Internas da Subbanda para Coeficientes em uma única Família

Grupo de Coeficiente na Família i	Coordenadas
Mãe, p_i	(r, c)
Grupo dos Filhos, C_i	$(2r, 2c), (2r, 2c+1), (2r+1, 2c), (2r+1, 2c+1)$
Grupo dos Netos, H_{i0}	$(4r, 4c), (4r, 4c+1), (4r+1, 4c), (4r+1, 4c+1)$
Grupo dos Netos, H_{i1}	$(4r, 4c+2), (4r, 4c+3), (4r+1, 4c+2), (4r+1, 4c+3)$
Grupo dos Netos, H_{i2}	$(4r+2, 4c), (4r+2, 4c+1), (4r+3, 4c), (4r+3, 4c+1)$
Grupo dos Netos, H_{i3}	$(4r+2, 4c+2), (4r+2, 4c+3), (4r+3, 4c+2), (4r+3, 4c+3)$

Tabela 4-2: Subbanda de Origem para Coeficientes AC

	Família 0	Família 1	Família 2
Mãe	HL_3	LH_3	HH_3
Filhos	HL_2	LH_2	HH_2
Netos	HL_1	LH_1	HH_1

Um *segmento* é definido como um grupo de S blocos consecutivos. A codificação dos coeficientes DWT continua segmento-por-segmento e cada segmento é codificado independentemente dos outros. S pode ser atribuído qualquer valor entre $16 \leq S \leq 2^{20}$, exceto para o último segmento de uma imagem, para o qual S pode ter qualquer valor no intervalo de $1 \leq S \leq 2^{20}$. O valor pode ser escolhido com base na memória disponível para armazenar o segmento. Quando múltiplos quadros de imagem são transmitidos, a codificação de cada novo quadro começa com um novo segmento, ou seja, um único segmento não precisa conter dados codificados de dois quadros separados.

Um segmento de blocos é ainda dividido em *conjuntos* (ou, *gaggles*). Cada conjunto (ou, *gaggle*) consiste em 16 blocos, exceto, possivelmente, para o último conjunto de segmento, o qual contém $S \bmod 16$, quando S não é um múltiplo de 16.

Coefficientes DC são representados usando representação em complemento de dois. Seja c_m o m -ésimo coeficiente DC em um segmento, ou seja, o coeficiente DC do m -ésimo bloco em um segmento. O número de bits necessários para representar c_m em complemento de dois é dado na equação 11:

$$\begin{aligned} &1 + \lceil \log_2 |c_m| \rceil, & \text{se } c_m < 0 \\ &1 + \lceil \log_2 (1 + c_m) \rceil, & \text{se } c_m \geq 0 \end{aligned} \quad (11)$$

Dentro de um segmento, BitDepthDC é definido como o máximo desse valor sobre todos os coeficientes DC (ou seja, todos os valores de m) no segmento. Cada coeficiente DC no segmento é representado usando BitDepthDC bits, em complemento de dois.

Um coeficiente AC é representado usando a representação binária da magnitude do coeficiente, mais um bit indicando o sinal, quando o coeficiente não é zero. BitDepthAC_Block $_m$ na equação 12 denota o número máximo de bits necessários para especificar a magnitude de qualquer coeficiente AC no m -ésimo bloco.

$$\text{BitDepthAC_Block}_m = \lceil \log_2 (1 + \max_x (|x|)) \rceil \quad (12)$$

onde a maximização é aplicada sobre todos coeficientes AC x no bloco.

Para cada segmento, o BPE calcula o BitDepthAC, o qual denota o máximo valor de BitDepthAC_Bloco $_m$ para o segmento:

$$\text{BitDepthAC} = \max_{m=0, \dots, S-1} \text{BitDepthAC_Block}_m \quad (13)$$

O BPE, sucessivamente, codifica os planos de bit das magnitudes do coeficiente em um segmento, introduzindo valores de sinal do coeficiente AC em pontos apropriados no fluxo (ou, *stream*) de dados codificados. O *plano de bits* b consiste no b -ésimo bit da representação de inteiros do complemento de dois de cada coeficiente DC, e o b -ésimo bit da representação de inteiros binária da magnitude de cada coeficiente AC. Aqui, o índice do plano de bits $b=0$ corresponde ao bit significativo. O BPE continua a partir do bit mais significativo para o bit menos significativo, assim b desce de um plano de bits para o próximo, começando com $b=\text{BitDepthAC}-1$, e terminando com $b=0$. A resolução do coeficiente da DWT melhora eficazmente de um fator de 2, à medida que a codificação

continua de um bit para o próximo. O processo de codificação do plano de bits é descrito em 4.5.

Figura 4-2(a) mostra um resumo da estrutura de um segmento codificado qualquer. Dentro de um segmento, as informações do cabeçalho são codificadas. Então, os coeficientes DC quantizados provenientes dos blocos são codificados. Logo, as resoluções ou profundidade de bit (ou, *bit depths*) AC são codificadas. Portanto, os blocos de coeficiente da DWT são codificados, um plano de bits por vez, seguindo do mais significativo para o plano de bits menos significativo. A codificação de um simples plano de bits é realizada em vários estágios, e a ordem resultante do código é ilustrada na figura 4-2(b). Por exemplo, coeficientes mães são codificados no estágio 1 para todos os blocos do segmento antes da codificação dos coeficientes filho no estágio 2. A seqüência de bits resultante constitui um formato integrado (ou, *embedded*) de dados que fornece uma transmissão progressiva.

(a) Com todos os Planos de Bits

Segmento Cabeçalho (ver 4.2)
Codificação inicial dos coeficientes DC (ver 4.3)
Resoluções do coeficiente AC codificado (ver 4.4)
Plano de bits codificado $b = \text{BitDepthAC}-1$ (ver 4.5)
Plano de bits codificado $b = \text{BitDepthAC}-2$ (ver 4.5)
⋮
Plano de bits codificado $b = 0$ (ver 4.5)

(b) Dentro de um Plano de Bits

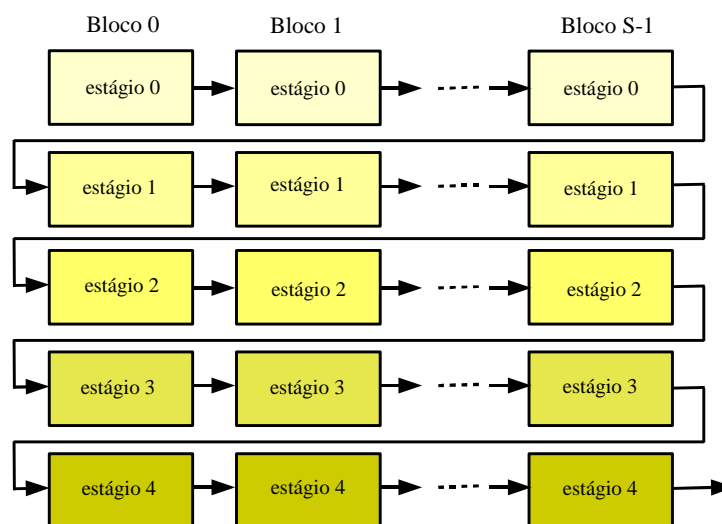


Figura 4-2: Visão Geral da Estrutura de um Segmento Codificado

A relação entre a qualidade da imagem reconstruída e o volume de dados comprimidos para cada segmento é controlada especificando-se o número máximo de bytes em cada segmento comprimido, *SegByteLimit*, e um limite de 'qualidade'. O limite da qualidade restringe a quantidade de coeficientes da DWT a serem codificados, e é especificado como um índice do

plano de bits e um ponto de parada dentro daquele plano de bits, como descrito em 4.2. A saída comprimida, para um segmento, é produzida até o limite do byte ou até que limite da qualidade é alcançado, o que vier primeiro.

A seqüência de bits codificada para um segmento pode ainda ser truncado (ou, de forma equivalente, a codificação pode ser terminada precocemente) em qualquer ponto para favorecer a redução da taxa de dados, ao custo de reduzir qualidade da imagem para o segmento correspondente.

O restante desta seção descreve cada componente do segmento codificado. A subseção 4.2 descreve o cabeçalho do segmento. Subseção 4.3 descreve a codificação a inicial dos coeficientes DC. Subseção 4.4 descreve a seqüência codificada de valores $BitDepthAC_Block_m$. Subseção 4.5 descreve o processo de codificação para um plano de bits do segmento.

4.2 CABEÇALHO DO SEGMENTO

4.2.1 GERAL

4.2.1.1 Cada segmento comprimido deve começar com um cabeçalho composto das seguintes partes, na ordem que se segue:

- a) Parte 1 (três ou quatro bytes, mandatário—ver 4.2.2);
- b) Parte 2 (cinco bytes, opcional—ver 4.2.3);
- c) Parte 3 (três bytes, opcional—ver 4.2.4);
- d) Parte 4 (oito bytes, opcional—ver 4.2.5).

4.2.1.2 Por convenção do CCSDS, todos os bits denominados ‘reservado’, em cada parte do cabeçalho, deverão ser fixados em ‘zero’.

4.2.1.3 Partes opcionais do cabeçalho podem ser omitidas quando os parâmetros descritos em uma parte puderem ser determinados sem aquela parte, p.ex., quando aqueles parâmetros estão ajustados para valores fixos conhecidos para uma missão completa.

NOTA – Figura 4-3 apresenta uma visão geral da estrutura do cabeçalho e a tabela 4-3 fornece uma descrição de alto nível da funcionalidade de cada parte do cabeçalho.

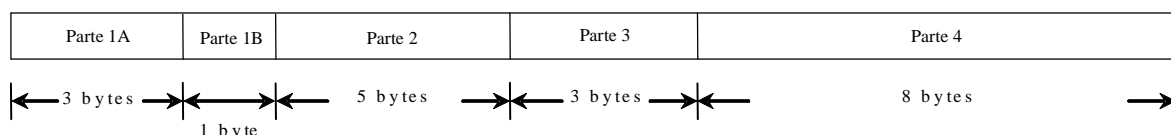


Figura 4-3: Visão Geral da Estrutura do Cabeçalho do Segmento Quando Todas as Partes Estão Incluídas

Tabela 4-3: Sumário de Funcionalidade do Cabeçalho do Segmento

Parte do Cabeçalho	Comprimento (bytes)	Estado	Descrição de Conteúdos	Referência
Parte 1A	3	Mandatário	Indica o primeiro e o último segmentos da imagem; indica quais partes cabeçalho opcional estão incluídas; codifica informações que tipicamente mudam de segmento-para-segmento.	4.2.2
Parte 1B	1	Mandatário ao último segmento de imagem, não incluído de outra forma	Especifica o número de linhas 'preenchidas' a serem eliminadas depois de reconstrução da imagem.	4.2.2
Parte 2	5	Opcional	Especifica limites na compressão bytes por segmento e qualidade da imagem.	4.2.3
Parte 3	3	Opcional	Opções de codificação incluindo número de blocos por segmento.	4.2.4
Parte 4	8	Opcional	Parâmetros da imagem e da compressão que devem ser fixados para uma imagem.	4.2.5

NOTA – A primeira parte mandatária do cabeçalho inclui informações do segmento que mudam de segmento-para-segmento. A segunda parte opcional do cabeçalho especifica limites no número de bytes comprimidos em um segmento e os limites na qualidade com a qual os coeficientes da DWT são codificados. Essa parte pode ser incluída no início de uma imagem ou na sessão de aplicação, ou no início de cada segmento codificado para controle da taxa de saída variável. A terceira parte opcional do cabeçalho especifica informações que são tipicamente fixadas para cada imagem ou sessão, mas é permitida alteração em cada segmento. Em uma aplicação típica, essa parte pode ser incluída no início de cada imagem, mas não é incluída para cada segmento. A quarta parte do cabeçalho especifica parâmetros que devem ser fixados para uma imagem inteira.

4.2.2 PARTE 1 DO CABEÇALHO DO SEGMENTO

4.2.2.1 Geral

A Parte 1 do Cabeçalho do Segmento consiste de duas subpartes:

- a) Parte 1A (3 bytes, mandatário para todos segmentos);
- b) Parte 1B (1 byte, mandatário para o último segmento, de outra maneira não é usado).

NOTA – Tabela 4.4 resume os conteúdos da Parte 1 do Cabeçalho do Segmento. A figura 4-4 ilustra a estrutura da Parte 1 do Cabeçalho do Segmento.

Tabela 4-4: Conteúdo da Parte 1 do Cabeçalho do Segmento

Parte	Campo	Largura (bits)	Descrição	Detalhes
Parte 1A	StartImgFlag	1	Sinaliza o segmento inicial em uma imagem	1: primeiro segmento na imagem 0: segmento de continuação na imagem
	EndImgFlag	1	Sinaliza o segmento final em uma imagem	1: último segmento na imagem 0: outro
	SegmentCount	8	Valor do Contador do segmento (ou, <i>segment counter value</i>)	Contador do segmento (mod 256), codificado como um inteiro binário não sinalizado (ou, <i>unsigned</i>)
	BitDepthDC	5	Número de bits necessários para representar os coeficientes em representação de complemento de 2	Valor do BitDepthDC codificado como um binário inteiro não sinalizado (ou, <i>unsigned</i>)
	BitDepthAC	5	Número de bits necessários para representar o valor absoluto dos coeficientes AC em representação de inteiros não sinalizados	Valor de BitDepthAC codificado como um inteiro binário não sinalizado (ou, <i>unsigned</i>)
	Reservado	1	Reservado para uso futuro	0
	Part2Flag	1	Indica a presença da Parte 2 do cabeçalho	1: presença da Parte 2 0: ausência da Parte 2
	Part3Flag	1	Indica a presença da Parte 3 do cabeçalho	1: presença da Parte 3 0: ausência da Parte 3
	Part4Flag	1	Indica a presença de da Parte 4 do cabeçalho	1: presença da Parte 4 0: ausência da Parte 4
Parte 1B	padrões	3	Número de linhas 'preenchidas' para eliminação depois da DWT inversa	Valor codificado como binário inteiro não sinalizado (ou, <i>unsigned</i>)
	Reservado	5	Reservado para uso futuro	00000

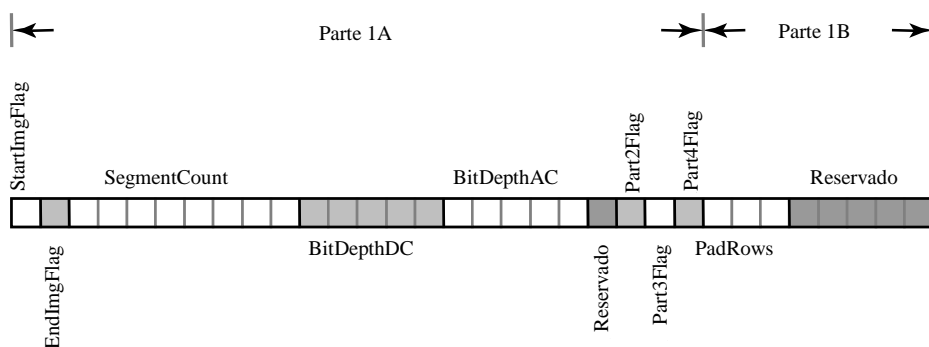


Figura 4-4: Parte 1 do cabeçalho do segmento quando a Parte 1B está incluída

4.2.2.2 Parte 1A do Cabeçalho do Segmento

4.2.2.2.1 O bit 0 (zero) da Parte 1A do Cabeçalho do Segmento deve conter a indicação (ou, *flag*) do Início da Imagem (StartImgFlag). A indicação (ou, *flag*) do Início da Imagem deve ser usada para indicar se o segmento corresponde ao início de uma imagem:

- ‘1’ deve indicar que o segmento é o primeiro segmento da imagem;
- ‘0’ deve indicar que o segmento não é o primeiro segmento.

4.2.2.2.2 Bit 1 da Parte 1A do Cabeçalho do Segmento deve conter a sinalização (ou, *flag*) de fim da imagem (EndImgFlag). A sinalização (ou, *flag*) de fim da imagem deve ser usada para indicar se o segmento é o último segmento da imagem:

- ‘1’ deve indicar que o segmento é o último segmento da imagem, nesse caso, a Parte 1B do Cabeçalho do Segmento deve ser incluída;
- ‘0’ deve indicar que o segmento não é o último segmento, nesse caso, a Parte 1B do Cabeçalho do Segmento não será incluída.

4.2.2.2.3 Bits de 2 a 9 da Parte 1A do Cabeçalho do Segmento devem fornecer o contador do segmento (ou, *segment counter*) (SegmentCount):

- a) o contador de segmento de oito-bits (ou, *eight-bit segment counter*) deve ser codificado como um binário inteiro não sinalizada;
- b) o contador segmento deve ser igual à zero para o primeiro segmento de uma imagem e deve ser incrementado de um a cada segmento subsequente;
- c) o contador do segmento deve voltar a zero e retomar a contagem depois de alcançar um valor de 255.

NOTA – Esse contador fornece um índice de segmentos, módulo 256, dentro da imagem. Contagem absoluta do segmento pode ser implementada usando o Pacote de Contagem de Sequência do CCSDS como descrito na referência [B3].

4.2.2.2.4 Bits de 10 a 14 devem conter o campo BitDepthDC. O valor do campo BitDepthDC de cinco-bits será codificado como um binário inteiro sem sinal e será igual ao número de bits necessários para representar os coeficientes DC para a representação do segmento em complemento de 2.

4.2.2.2.5 Bits de 15 a 19 devem conter o campo BitDepthAC. O valor de cinco-bits da variável BitDepthAC será codificado como um inteiro binário não sinalizado e será igual ao número de bits necessários para representar o valor absoluto dos coeficientes AC na representação de inteiros não sinalizado (ver 4.4).

4.2.2.2.6 O bit 20 está reservado para uso futuro do CCSDS e será fixado em ‘0’.

4.2.2.2.7 O bit 21 da Parte 1A do Cabeçalho do Segmento deve conter a sinalização (ou, *flag*) da Parte 2 (Part2Flag) e deve indicar a presença ou ausência da parte 2, opcional, do cabeçalho do segmento:

- ‘1’ indicará que a parte 2 do cabeçalho do segmento está presente;
- ‘0’ indicará que a parte 2 do cabeçalho do segmento está ausente.

4.2.2.2.8 O bit 22 da Parte 1A do Cabeçalho do Segmento deve conter a sinalização (ou, *flag*) da Parte 3 e deve indicar a presença ou ausência da parte 3 opcional do cabeçalho do segmento:

- ‘1’ indicará que a parte 3 do cabeçalho do segmento está presente;
- ‘0’ indicará que a parte 3 do cabeçalho do segmento está ausente.

4.2.2.2.9 O bit 23 da Parte 1A do Cabeçalho do Segmento deve conter a sinalização (ou, *flag*) da Parte 4 e deve indicar a presença ou ausência da parte 4 opcional do cabeçalho do segmento:

- ‘1’ indicará que a parte 4 do cabeçalho do segmento está o presente;
- ‘0’ indicará que a parte 4 do cabeçalho do segmento está ausente.

4.2.2.3 Parte 1B do cabeçalho do segmento

4.2.2.3.1 Quando o EndImgFlag (4.2.2.2.2) na parte 1A do cabeçalho do segmento é colocado em ‘1’, indicando que o segmento é o último segmento da imagem, a parte 1B do cabeçalho do segmento deve ser incluída.

4.2.2.3.2 Bits de 0 a 2 da parte 1B do cabeçalho do segmento devem conter o campo PadRows. O valor do campo de três-bits PadRows deve ser codificado como um binário inteiro sem sinal e deve ser igual ao número de linhas de ‘preenchimento’ (ou, *padding*) a serem eliminadas depois da DWT inversa ser executada (ver 3.2).

4.2.2.3.3 Bits de 3 a 7 da parte 1B do cabeçalho do segmento devem ser reservados para uso futuro do CCSDS serão colocados em ‘0’.

4.2.3 PARTE 2 DO CABEÇALHO DO SEGMENTO

4.2.3.1 Geral

Se usada, a parte 2 (opcional) do cabeçalho do segmento deve especificar opções de saída que controlam a relação (ou, *tradeoff*) entre o volume de dados comprimidos e a qualidade da imagem reconstruída para um segmento.

NOTA – Tabela 4-5 resume os conteúdos a parte 2 do cabeçalho do segmento. Figura 4-5 ilustra a estrutura da parte 2 do cabeçalho do segmento.

Tabela 4-5: Conteúdo da parte 2 do cabeçalho do segmento

Campo	Largura (bits)	Descrição	Detalhes
SegByteLimit	27	Número máximo de bytes comprimidos em um segmento.	valor (2 do mod ²⁷) codificado como um inteiro não sinalizado
DCStop	1	Indica se a saída comprimida pára depois da codificação dos coeficientes DC quantizados (4.3).	1: Pára a codificação depois da informação de coeficiente DC quantizado (ver 4.3.2) e planos de bit adicionais (ver 4.3.3) 0: Parada na codificação determinada pelos valores BitPlaneStop e StageStop
BitPlaneStop	5	Não usado quando DCStop = 1. Quando DCStop = 0 indica o limite na codificação dos planos de bits do coeficiente da DWT. Quando BitPlaneStop = <i>b</i> e StageStop = <i>s</i> , saída comprimida pára, uma vez que o estágio <i>s</i> do plano de bits <i>b</i> tenha sido completado (ver 4.5), a menos que a codificação pare antes no segmento por causa do limite de byte do segmento(SegByteLimit).	Valor índice do plano de bits BitPlaneStop codificado como um inteiro não sinalizado
StageStop	2		00: estágio 1 (ver 4.5.3) 01: estágio 2 (ver 4.5.3) 10: estágio 3 (ver 4.5.3) 11: estágio 4 (ver 4.5.4)
UseFill	1	Especifica se bits de enchimento (ou, <i>fill bits</i>) serão usados para produzir SegByteLimit-bytes em cada segmento.	1: bits de enchimento (ou, <i>fill bits</i>) usados se necessário para produzir SegByteLimit-bytes em cada segmento 0: bits de enchimento (ou, <i>fill bits</i>) não permitido
Reservado	4	Reservado para uso futuro.	0000

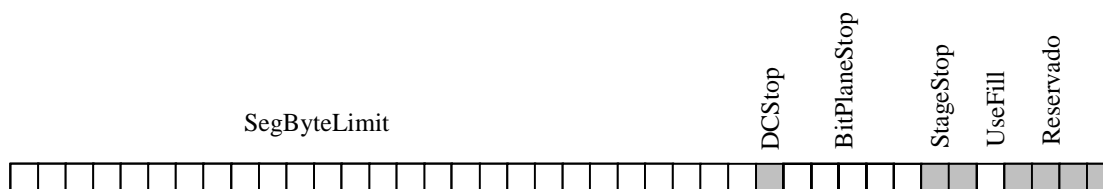


Figura 4-5: Parte 2 do cabeçalho do segmento

4.2.3.2 Conteúdo da Parte 2 do Cabeçalho do Segmento

4.2.3.2.1 Bits de 0 a 26 da parte 2 do cabeçalho do segmento devem conter o campo SegByteLimit:

- a) o valor do campo SegByteLimit será codificado como um binário inteiro sem sinal e será igual ao número máximo de bytes que podem ser usados em um segmento, incluindo bytes usados para o cabeçalho;

NOTA – Um valor SegByteLimit menor do que 9 antes do último segmento e 10 para o último segmento não seriam permitidos por causa do número combinado de bytes nas Partes 1 e 2 do cabeçalho. O valor de SegByteLimit precisa ser um múltiplo inteiro do tamanho da palavra especificado pelo CodeWordLength em 4.2.5.2.7.

- b) o valor deve ser aplicado ao segmento atual e todos os segmentos subseqüentes até que um novo valor de Segbytelimit seja codificado na Parte 2 do cabeçalho do segmento de um cabeçalho do segmento posterior.

4.2.3.2.2 O bit 27 da parte 2 do cabeçalho do segmento deve conter a sinalização (ou, *flag*) DCStop. A sinalização (ou, *flag*) DCStop irá indicar o fim da saída comprimida:

- a) ‘1’ deve indicar que a saída comprimida termina, uma vez que, o valor quantizado do coeficiente DC (ver 4.3.2) e os planos de bit adicionais de coeficientes DC (ver 4.3.3) foram codificados;
- b) ‘0’ indicará que fim da saída comprimida será determinado por um índice do plano de bits (BitPlaneStop) e por um estágio de codificação dentro da codificação daquele plano de bits (StageStop).

NOTA – Dados do segmento codificado param no ponto indicado no processo de codificação do BPE, ou quando o limite de byte do segmento é alcançado, o que vier primeiro. A saída de dados comprimida sempre inclui, pelo menos, o valor dos coeficientes DC quantizados e quaisquer planos de bit DC adicionais dos coeficientes DC (4.3.2 e 4.3.3), a menos que o valor de SegByteLimit seja muito pequeno para permitir isso.

4.2.3.2.3 Os bits de 28 a 32 da parte 2 do cabeçalho do segmento devem conter o campo BitPlaneStop:

- a) o bit 5 do campo BitPlaneStop será codificado como um binário inteiro não sinalizado e indicará o limite na codificação dos planos de bit do coeficiente da DWT;
- b) quando DCStop é ‘0’, o valor BitPlaneStop será igual ao índice do plano de bits no qual a codificação pára, a menos que a codificação pare antes, porque o SegByteLimit foi alcançado primeiro;
- c) quando DCStop é ‘1’ o valor BitPlaneStop e o valor StageStop serão ignorados.

NOTA – Quando o valor *BitPlaneStop* fornecido pelo usuário é maior que *BitDepthAC-1*, a compressão terminará depois de completar a codificação dos valores quantizados do coeficiente DC e de quaisquer planos de bit DC adicionais (4.3.2 e 4.3.3), a menos que o *SegByteLimit* seja muito pequeno para permitir isso.

4.2.3.2.4 Os bits de 33 a 34 da parte 2 do cabeçalho do segmento devem conter o campo *StageStop*. O campo de dois-bits *StageStop* indicará o estágio no qual a codificação pára no plano de bits indicado por *BitPlaneStop*:

- ‘00’ indicará estágio 1 (ver 4.5.3);
- ‘01’ indicará estágio 2 (ver 4.5.3);
- ‘10’ indicará estágio 3 (ver 4.5.3);
- ‘11’ indicará estágio 4 (ver 4.5.4).

NOTA – Compressão é totalmente limitada pelo limite de bytes de segmento quando *DCStop* é o 0, *BitPlaneStop* é 0, e *StageStop* é ‘11’ (i.e., estágio 4). Nesse caso, compressão sem perdas é alcançada para um segmento quando a DWT de Inteiros é usada e, os dados comprimidos para o segmento requererem menos do que *SegByteLimit* bytes.

4.2.3.2.5 O bit 35 da parte 2 do cabeçalho do segmento deve conter o campo *UseFill*, o qual deve indicar se os bits de enchimento (ou, *fill bits*) serão usados para produzir *SegByteLimit* bytes em cada segmento:

- a) se o valor do campo *UseFill* é ‘1’, os bits de enchimento (ou, *fill bits*) são usados sempre que for preciso produzir *SegByteLimit* bytes em cada segmento;
- b) se o valor do campo *UseFill* é ‘0’:
 - 1) bits de enchimento (ou, *fill bits*) não são usados para produzir *SegByteLimit* bytes, e o número de bytes comprimidos em um segmento pode ser menor do que *SegByteLimit* bytes;
 - 2) os bits de enchimento (ou, *fill bits*) são usados para preencher o próximo limite (ou, *boundary*) da palavra, quando o ponto de parada (segundo os valores especificados de *DCStop*, *BitPlaneStop*, *StageStop*) é alcançado antes do limite de byte especificado em *SegByteLimit*;

NOTA – Uma palavra corresponde à unidade na qual o compressor produz uma saída, que pode ser um único byte, dois bytes, ou quatro bytes. Por exemplo, um compressor que produz 4 bytes de saída de uma vez poderia incluir 31 os bits de enchimento (ou, *fill bits*) na última palavra do segmento comprimido. O comprimento da palavra do compressor é indicado pelo campo *CodeWordLength* na parte 4 do cabeçalho do segmento (ver 4.2.5).

- c) os bits de enchimento (ou, *fill bits*) devem ser colocados em ‘tudo zero’ (ou, *all zero*).

4.2.3.2.6 Bits de 36 a 39 da Parte 2 do cabeçalho do segmento são reservados para uso futuro do CCSDS e devem ser colocados em ‘ tudo zero’ (ou, *all zero*).

4.2.4 PARTE 3 DO CABEÇALHO DO SEGMENTO

4.2.4.1 Geral

4.2.4.1.1 Se usada, a Parte 3 do Cabeçalho do Segmento deverá especificar o tamanho do segmento em função do número de blocos, S , definido em 4.1 e indicar se a abordagem heurística de seleção de parâmetro (4.3.2) é usada na codificação dos coeficientes DC quantizados (ver 4.3) e/ou na profundidade de bits (ou, *bit depths*) do coeficiente AC (ver 4.4).

4.2.4.1.2 A informação codificada nessa parte do cabeçalho é permitida mudar com cada segmento.

NOTA – Tabela 4-6 resume o conteúdo da Parte 3 do Cabeçalho do Segmento. Figura 4-6 ilustra a estrutura da Parte 3 do Cabeçalho do Segmento.

Tabela 4-6: Conteúdo da Parte 3 do Cabeçalho do Segmento

Parâmetro	Largura (bits)	Descrição	Detalhes
S	20	tamanho do segmento em blocos	Valor codificado, mod 2^{20} , como um binário inteiro não sinalizado
OptDCSelect	1	Especifica se o método ótimo ou heurístico é usado para selecionar valor do parâmetro k para codificar os valores do coeficiente DC quantizado (ver 4.3.2)	1: seleção ótima de k 0: seleção heurística de k
OptACSelect	1	Especifica se método ótimo ou heurístico é usado para selecionar o valor do parâmetro k para codificar o BitDepthAC (ver 4.4)	1: seleção ótima de k 0: seleção heurística de k
Reservado	2	Reservado	00

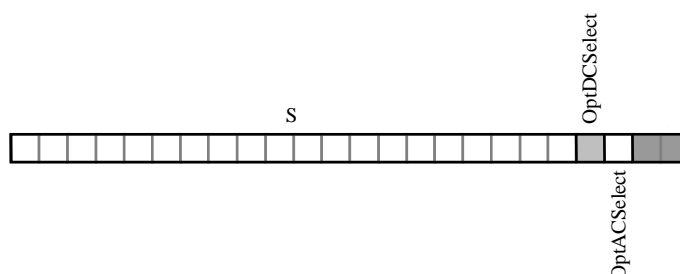


Figura 4-6: Parte 3 do Cabeçalho do Segmento

4.2.4.2 Conteúdo da Parte 3 do Cabeçalho do Segmento

4.2.4.2.1 Bits de 0 a 19 da Parte 3 do Cabeçalho do Segmento devem conter o campo ‘Tamanho do Segmento’ (ou, *Segment Size*). O campo de 20-bits ‘Tamanho do Segmento’ será codificado como um binário inteiro não sinalizado e será igual a $S \cdot 2$ do mod²⁰; S será o número de blocos que compõem o segmento.

4.2.4.2.2 Bit 20 da Parte 3 do Cabeçalho do Segmento deve conter o campo OptDCSelect. O campo OptDCSelect indicará o método usado para selecionar o valor do parâmetro k para codificar os valores quantizados dos coeficientes DC (ver 4.3.2.11):

- ‘1’ indicará seleção ótima de k ;
- ‘0’ indicará a seleção heurística de k .

4.2.4.2.3 Bit 21 da Parte 3 do Cabeçalho do Segmento deve conter o campo OptACSelect. O campo OptACSelect indicará o método usado para selecionar o valor do parâmetro k para codificar a profundidade dos bits (ou, *bit depths*) do coeficiente AC (ver 4.4):

- ‘1’ indicará seleção ótima de k ;
- ‘0’ indicará seleção heurística de k .

4.2.4.2.4 Bits de 22 a 23 da Parte 3 do Cabeçalho do Segmento 3 são reservados para uso futuro pelo CCSDS e serão colocados em ‘tudo zero’ (ou, *all zero*).

4.2.5 PARTE 4 DO CABEÇALHO DO SEGMENTO

4.2.5.1 Geral

Se usada, a Parte 4 do Cabeçalho do Segmento (opcional) fornecerá informações que não mudam dentro de uma imagem, e que poderiam muitas vezes ser fixadas para uma sessão da aplicação. Isso inclui o tipo da DWT, pesos opcionais definidos pelo usuário para escalonar as subbandas, informações sobre a imagem de entrada, e o comprimento da palavra codificada.

NOTA – Tabela 4-7 resume o conteúdo da Parte 4 do Cabeçalho do Segmento. Figura 4-7 ilustra a estrutura da Parte 4 do Cabeçalho do Segmento.

Tabela 4-7: Conteúdo da Parte 4 do Cabeçalho do Segmento

Parâmetro	Largura (byte)	Descrição	Detalhes
DWTtype	1	Especifica tipo da DWT	0: DWT de ponto flutuante 1: DWT de Inteiros
Reservado	2	Reservado para uso futuro	00
SignedPixels	1	Especifica se os valores do pixel de entrada são quantidades sinalizadas ou não sinalizadas	0: não sinalizado 1: não sinalizado
PixelBitDepth	4	Especifica a profundidade de bit do pixel entrada	Especifica valores de 1 a 16 por mod(16) de 1, 2, 3, ..., 16
ImageWidth	20	largura da imagem em píxeis	Valor codificado, mod 2^{20} , como um binário inteiro não sinalizado
TransposeImg	1	Indica se imagem inteira deveria ser transposta depois de reconstrução	0: não transpor a imagem 1: transpor a imagem
CodeWordLength	2	Indica o comprimento da palavra codificada	00: palavra de 8-bits 01: palavra de 16-bits 10: palavra de 24-bits 11: palavra de 32-bits
Reservado	1	Reservado para uso futuro	0
CustomWtFlag	1	Indica se são usados os pesos em 3.9 ou os definidos pelo usuário	0: usados os pesos de 3.9 1: pesos de definidos pelo usuário
CustomWtHH ₁	2	Peso da subbanda HH ₁	00: peso = 2^0 01: peso = 2^1 10: peso = 2^2 11: peso = 2^3 (estes campos são fixados em 00 quando CustomWtFlag é 0)
CustomWtHL ₁	2	Peso da subbanda HL ₁	
CustomWtLH ₁	2	Peso da subbanda LH ₁	
CustomWtHH ₂	2	Peso da subbanda HH ₂	
CustomWtHL ₂	2	Peso da subbanda HL ₂	
CustomWtLH ₂	2	Peso da subbanda LH ₂	
CustomWtHH ₃	2	Peso da subbanda HH ₃	
CustomWtHL ₃	2	Peso da subbanda HL ₃	
CustomWtLH ₃	2	Peso da subbanda LH ₃	
CustomWtLL ₃	2	Peso da subbanda LL ₃	
Reservado	11	Reservado para uso futuro	00000000000

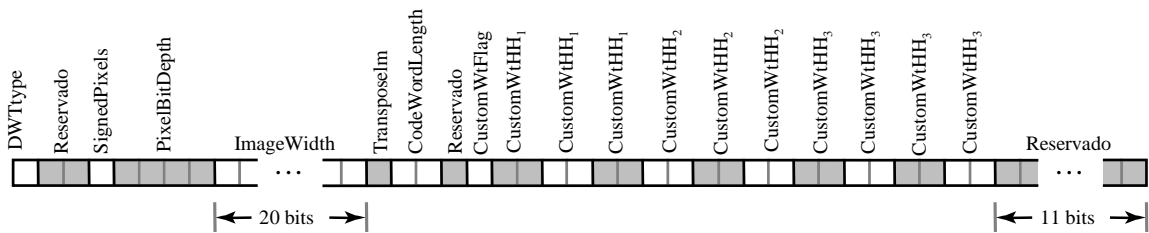


Figura 4-7: Parte 4 do Cabeçalho do Segmento

4.2.5.2 Conteúdo da Parte 4 do Cabeçalho do Segmento

4.2.5.2.1 Bit 0 da Parte 4 do Cabeçalho do Segmento deve conter o campo ‘Tipo DWT’ (DWType) e deve indicar o tipo de DWT como segue:

- ‘0’ indicará DWT de ponto flutuante;
- ‘1’ indicará DWT de inteiros.

4.2.5.2.2 Bits 1 e 2 da Parte 4 do Cabeçalho do Segmento são reservados para uso futuro pelo CCSDS e devem ser colocados em ‘tudo zero’ (ou, *all zero*).

4.2.5.2.3 Bit 3 da Parte 4 do Cabeçalho do Segmento deve conter o campo Pixel Sinalizado (SignedPixels) e deve indicar se os valores do pixel de entrada são quantidades sinalizadas (ou, *signed*) ou não sinalizadas (ou, *unsigned*):

- ‘0’ indicará não sinalizada (ou, *unsigned*);
- ‘1’ indicará sinalizada (ou, *signed*).

4.2.5.2.4 Bits de 4 a 7 da Parte 4 do Cabeçalho do Segmento devem conter o campo Pixel Bit Depth (PixelBitDepth). O campo de quatro-bits Pixel Bit Depth deve ser codificado como um binário inteiro não sinalizado e deve ser igual a profundidade de bit por pixel mod 16.

4.2.5.2.5 Bits de 8 a 27 da Parte 4 do Cabeçalho do Segmento devem conter o campo Largura da Imagem (ImageWidth). O campo de 20-bits Largura da Imagem deve ser codificado como um binário inteiro não sinalizado e será igual à largura da imagem em píxeis mod 2^{20} .

4.2.5.2.6 Bit 28 da Parte 4 do Cabeçalho do Segmento deve conter o campo Transpor Imagem (TransposeIm) e deve indicar se toda imagem deveria ser transposta depois da reconstrução:

- ‘0’ indicará que a imagem não deveria ser transposta;
- ‘1’ indicará que a imagem deveria ser transposta.

4.2.5.2.7 Bits de 29 a 30 da Parte 4 do Cabeçalho do Segmento devem conter o campo Comprimento da Palavra Código (ou, *Code Word Length*) (*CodeWordLength*) e deve indicar o comprimento da palavra código (ou, *code word*) como segue:

- ‘00’: palavra de 8-bits;
- ‘01’: palavra de 16-bits;
- ‘10’: palavra de 24-bits;
- ‘11’: palavra de 32-bits.

4.2.5.2.8 Bit 31 da Parte 4 do Cabeçalho do Segmento é reservado para uso futuro pelo CCSDS e será colocado em ‘0’.

4.2.5.2.9 Bit 32 da Parte 4 do Cabeçalho do Segmento deve conter a sinalização (ou, *flag*) Pesos Customizados (ou, *Custom Weights*) (*CustomWtFlag*) e deve indicar se são usados os pesos padrão (3.9) ou os definidos pelo usuário:

- ‘0’ deve indicar que os pesos da subbanda definidos em 3.9 são usados;
- ‘1’ deve indicar que os pesos da subbanda definidos pelo usuário são usados;

4.2.5.2.10 Bits de 33 a 52 devem fornecer 10 campos de 2-bits contendo o peso adaptado da subbanda, para cada uma das subbandas HH_1 – LL_3 :

a) Campos contendo o peso customizado da subbanda devem ser ordenados como segue:

- bits 33-34: HH_1 ,
- bits 35-36: HL_1 ,
- bits 37-38: LH_1 ,
- bits 39-40: HH_2 ,
- bits 41-42: HL_2 ,
- bits 43-44: LH_2 ,
- bits 45-46: HH_3 ,
- bits 47-48: HL_3 ,
- bits 49-50: LH_3 ,
- bits 51-52: LL_3 ;

b) se *CustomWtFlag* está colocado em ‘1’, os valores dos campos contendo o peso customizado da subbanda devem ser definidos como segue:

- ‘00’: peso = 2^0 ,
- ‘01’: peso = 2^1 ,
- ‘10’: peso = 2^2 ,
- ‘11’: peso = 2^3 ;

c) se *CustomWtFlag* é colocado em ‘0’, os valores dos bits de 33 a 52 devem ser colocados em ‘tudo zero’ (ou, *all zero*).

4.2.5.2.11 Bits de 53 a 63 da Parte 2 do Cabeçalho do Segmento são reservados para uso futuro pelo CCSDS e serão colocados em ‘tudo zero’ (ou, *all zero*).

4.3 CODIFICAÇÃO INICIAL DOS COEFICIENTES DC

4.3.1 GERAL

4.3.1.1 A codificação inicial dos coeficientes DC em um segmento é realizada em dois passos. O primeiro passo deve ser codificar uma representação quantizada dos coeficientes DC usando o esquema preditivo descrito nesta seção (ver 4.3.2). O segundo passo deve ser codificar q -BitDepthAC planos de bit adicionais dos coeficientes DC (ver 4.3.3), quando $q > \text{BitDepthAC}$ (q será descrito abaixo). Os bits restantes BitDepthAC do coeficiente DC no segmento devem ser codificados durante o estágio 0 do processo de codificação do plano de bits de 4.5, como necessário, quando o índice b do plano de bits satisfaz $b < q$.

4.3.1.2 O resultado da quantização dos coeficientes DC realizada nesse passo da codificação será determinado pelo range dinâmico (ou, *dynamic range*) dos coeficientes AC e DC em um segmento, através do parâmetro inteiro q' , o qual é definido de acordo com a tabela 4-8.

Tabela 4-8: Quantização dos Coeficientes DC

Range Dinâmico DC e AC	Valor q'	Observação
$\text{BitDepthDC} \leq 3$	$q' = 0$	Range dinâmico DC é muito pequeno: nenhuma quantização é realizada – toda informação do coeficiente DC é codificada neste passo.
$\text{BitDepthDC} - (1 + \lfloor \text{BitDepthAC}/2 \rfloor) \leq 1$ e $\text{BitDepthDC} > 3$	$q' = \text{BitDepthDC} - 3$	Range dinâmico DC é aproximadamente a metade do range dinâmico AC: os três bits mais significativos dos coeficientes DC são codificados de forma diferente neste passo.
$\text{BitDepthDC} - (1 + \lfloor \text{BitDepthAC}/2 \rfloor) > 10$ e $\text{BitDepthDC} > 3$	$q' = \text{BitDepthDC} - 10$	Range dinâmico DC é muito mais alto do que a metade do range dinâmico AC: os 10 bits mais significativos dos coeficientes DC são codificados de forma diferente neste passo.
de outra maneira	$q' = 1 + \lfloor \text{BitDepthAC}/2 \rfloor$	Range dinâmico DC é moderadamente mais alto do que a metade do range dinâmico AC: os bits do coeficiente DC que excedem a metade do range dinâmico AC são codificados de forma diferente neste passo.

4.3.1.3 O fator de quantização DC q , na equação 14, é definido como:

$$q = \max(q', \text{BitShift}(\text{LL}_3)) \quad (14)$$

para levar em conta os planos de bit DC que são necessariamente todos zero como resultado da operação de escalonamento da subbanda.

NOTA – O valor de q indica o número de bits menos significativos em cada coeficiente DC que são *não* codificados nos valores quantizados do coeficiente DC. O valor de q tende a ser pequeno o bastante, que uma quantidade significativa de correlação nos coeficientes DC pode ser explorada na compressão desses coeficientes quantizados. Ao mesmo tempo, o valor de q tende a ser grande o bastante para assegurar que o BPE não gaste um número grande de bits codificando os coeficientes DC em uma resolução muito alta, antes de codificar qualquer informação dos coeficientes AC.

4.3.1.4 Para quantização na fase inicial da codificação DC, o fator da quantização DC deve ser comparado ao range dinâmico BitDepthAC dos coeficientes AC do segmento, seguindo as regras definidas na tabela 4-8.

NOTA – Os bits DC de refinamento necessários para codificação dos coeficientes DC com maior precisão são codificados a posteriori, no estágio 0 da Codificação do Plano de Bits, como descrito em 4.5.

4.3.1.5 A seguir, dada uma seqüência de coeficientes DC $\{c_m: m=0, \dots, S-1\}$ em um segmento, o BPE deve calcular os coeficientes quantizados

$$c'_m = \lfloor c_m / 2^q \rfloor \quad (15)$$

4.3.1.6 Os coeficientes DC quantizados devem ser codificados usando o procedimento descrito em 4.3.2, o qual eficazmente codifica os $\text{BitDepthDC} - q$ bits mais significativos de cada coeficiente DC.

4.3.1.7 Quando $q > \text{BitDepthAC}$, os próximos bits mais significativos $q - \text{BitDepthAC}$ de cada coeficiente DC aparecem na seqüência de bits (ou, *bitstream*) codificado, como descrito em 4.3.3.

4.3.1.8 Bits restantes dos coeficientes DC aparecem na seqüência de bits codificada como parte do procedimento de codificação do plano de bits descrito em 4.5, exceto para os bits menos significativos $\text{BitShift}(\text{LL}_3)$ de cada coeficiente DC, qual precisa ser o zero devido ao escalonamento do coeficiente, e ser assim não codificado.

4.3.2 CODIFICAÇÃO DOS COEFICIENTES DC QUANTIZADOS

4.3.2.1 O número de bits necessários para representar cada coeficiente DC quantizado c'_m é dado na equação 16 como:

$$N = \max\{\text{BitDepthDC} - q, 1\} \quad (16)$$

NOTA – As regras de quantização do coeficiente DC, na tabela 4-8, limitam o valor de N a 10, e assim, as opções de codificação são definidas somente até $N=10$.

4.3.2.2 Quando N é 1, cada coeficiente DC quantizado, c'_m , consiste de um único bit. Nesse caso, os coeficientes DC quantizados para um segmento consistem desses bits concatenados. Por outro lado, $N>1$ e os coeficientes DC quantizados em um segmento, c'_m , devem ser codificados usando o procedimento descrito a seguir.

4.3.2.3 O primeiro coeficiente DC quantizado para toda seqüência de S coeficientes consecutivos, chamado de *amostra de referência* (ou, *reference sample*), deve ser escrito diretamente na seqüência de bits (ou, *bitstream*) codificada (i.e., sem qualquer processamento adicional ou codificação).

4.3.2.4 Para os $S-1$ coeficientes DC restantes, a diferença entre valores sucessivos de coeficientes quantizados (tomados na ordem de varredura (ou, *scan*)) devem ser codificados. Cada valor de diferença, δ'_m , na equação 17,

$$\delta'_m = c'_m - c'_{m-1} \quad (17)$$

deve ser mapeado para um inteiro não-negativo, δ_m , segundo as regras da equação 18:

$$\begin{aligned} \delta_m &= 2(\delta'_m), & \text{se } 0 \leq \delta'_m \leq \theta_m \\ \delta_m &= 2|\delta'_m|-1, & \text{se } -\theta_m \leq \delta'_m < 0 \\ \delta_m &= \theta_m + |\delta'_m|, & \text{de outra maneira} \end{aligned} \quad (18)$$

onde $\theta_m = \min(c'_{m-1} - x_{\min}, x_{\max} - c'_{m-1})$, e $x_{\min} = -2^{N-1}$, $x_{\max} = 2^{N-1} - 1$.

4.3.2.5 Cada conjunto (ou, *gaggle*) contém até 16 coeficientes quantizados e mapeados δ_m :

- a) o primeiro conjunto contém 15 valores de δ_m (o primeiro coeficiente DC quantizado é codificado diretamente como uma amostra de referência (ou, *reference sample*));
- b) cada conjunto (ou, *gaggle*) restante contém 16 valores de δ_m , com a exceção possível do último conjunto (ou, *gaggle*), o qual pode ter menos;
- c) se S não é um múltiplo de 16, então o último conjunto (ou, *gaggle*) contém J valores de δ_m , onde J iguala-se a $S \bmod 16$.

4.3.2.6 Os valores de δ_m , em cada conjunto, devem ser codificados usando uma das várias opções de código variando de não codificado (ou, *uncoded*) (isto é, cada δ_m é codificado usando a representação convencional de inteiros binária sem sinal de N -bits) para codificado através de um dos vários códigos de comprimento variável parametrizados por um inteiro não negativo k .

NOTA – Esse procedimento de predição, mapeamento para valores não negativos, e codificação segue o padrão CCSDS de Compressão Sem Perdas (referência [B2]) com um codificador por entropia (ou, *entropy coder*) ligeiramente simplificado e seqüência diferentes para identificar a opção de código selecionada para cada conjunto (ou, *gaggle*).

4.3.2.7 A opção de código selecionada para o conjunto deve ser indicada no início dos coeficientes DC quantizados codificados no conjunto usando o apropriado *identificador de opção de código* (ou, *code option identifier*) selecionado de tabela 4-9.

Tabela 4-9: Identificadores de Opção de Código

Parâmetro k do Código	$N=2$	$2 < N \leq 4$	$4 < N \leq 8$	$8 < N \leq 10$
$k = 0$	0	00	000	0000
$k = 1$	–	01	001	0001
$k = 2$	–	10	010	0010
$k = 3$	–	–	011	0011
$k = 4$	–	–	100	0100
$k = 5$	–	–	101	0101
$k = 6$	–	–	110	0110
$k = 7$	–	–	–	0111
$k = 8$	–	–	–	1000
não codificado	1	11	111	1111

NOTA: ‘–’ indica valor não aplicável.

4.3.2.8 Quando a opção ‘não codificado’ (ou, *uncoded*) é selecionada, o conjunto (ou, *gaggle*) codificado é fixado em comprimento, consistindo na opção ID (identificação), de forma opcional, seguido por uma amostra de referência de N -bits, e J valores de δ_m como mostrado na figura 4-8.

4.3.2.9 Por outro lado, dado o parâmetro de código k , a palavra código (ou, *codeword*) de comprimento-variável para δ_m tem duas partes. A primeira parte deve consistir de z zeros seguidos por um 1, onde $z = \lfloor \delta_m / 2^k \rfloor$. A segunda parte deve consistir de k bits menos-significativo da representação binária de δ_m .

NOTA – z pode ser calculado realizando k deslocamentos de bit a direita da representação binária de δ_m .

4.3.2.10 Os valores codificados de δ_m , dentro de um conjunto (ou, *gaggle*), devem consistir do identificador de opção de código, então, uma amostra de referência (para o primeiro conjunto (ou, *gaggle*) em um segmento), depois a seqüência correspondente de palavras código (ou, *codewords*) da primeira parte para as amostras em um conjunto (ou, *gaggle*), seguido pelas palavras código (ou, *codewords*) da segunda parte, como mostrado na figura 4-9.

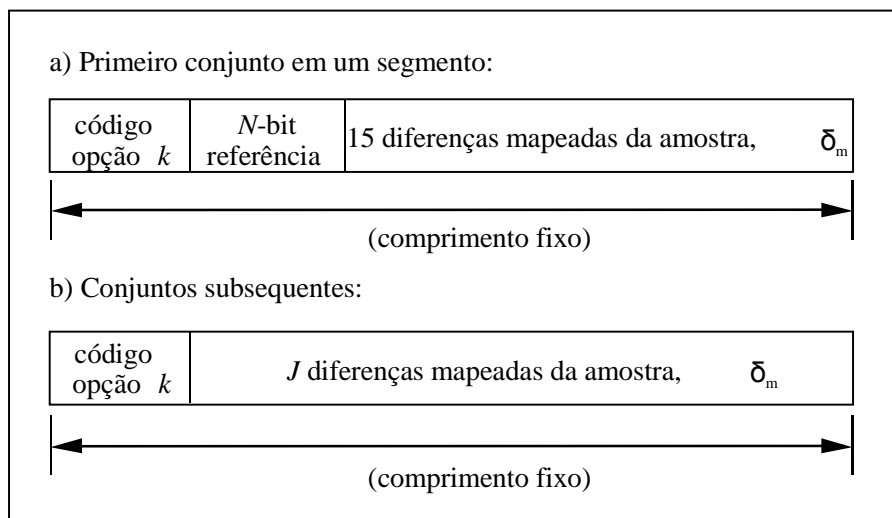


Figura 4-8: Formato dos Dados Codificados para um Conjunto (ou, Gaggle) Quando a Opção Não Codificada (ou, Uncoded) é Selecionada

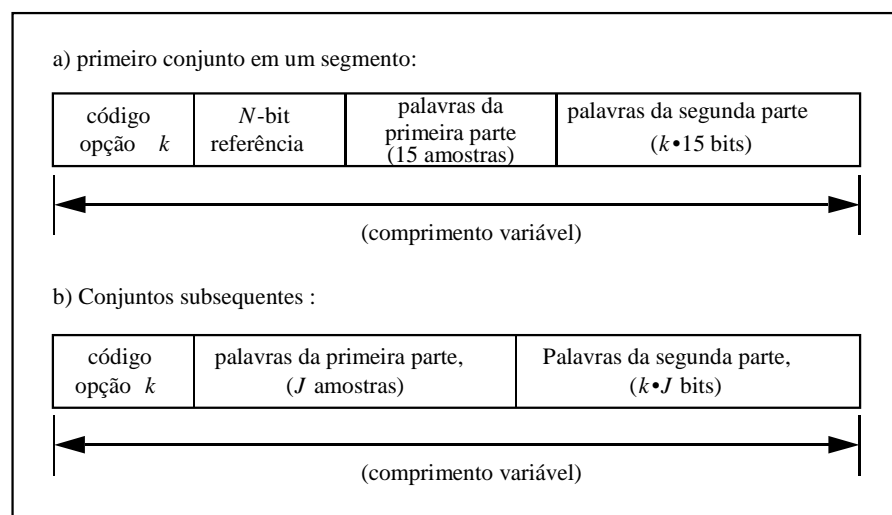


Figura 4-9: Formato dos Dados Codificados para um Conjunto (ou, Gaggle) Quando uma Opção Codificada (ou, Coded) é Selecionada

4.3.2.11 Para uma dada aplicação, um usuário deve usar um dos seguintes métodos para selecionar a opção de código:

- a) selecionar o valor ótimo de k para cada conjunto (ou, gaggle) (i.e., selecionar o valor de k que minimiza o número de bits codificados);

NOTA – Essa otimização poderia ser realizada calculando explicitamente o número de bits codificados sob cada valor permitido de k .

- b) selecionar o valor de k , usando o seguinte procedimento heurístico simplificado (mas em geral sub-ótimo):

Seja Δ a soma dos valores de δ_m em um conjunto (ou, *gaggle*):

$$\Delta = \sum_m \delta_m \quad (19)$$

A opção de código é selecionada dependendo do valor da soma Δ , de acordo com as regras da tabela 4-10.

Tabela 4-10: Regras de Seleção da Opção de Código

Condição	Opção de Código Selecionada
$64 \cdot \Delta \geq 23 \cdot J \cdot 2^N$	não codificado
$207 \cdot J > 128 \cdot \Delta$	$k=0$
$J \cdot 2^{N+5} \leq 128 \cdot \Delta + 49 \cdot J$	$k=N-2$
de outra forma	k é o maior inteiro não negativo menor que ou igual a $N-2$, tal que $J \cdot 2^{k+7} \leq 128 \cdot \Delta + 49 \cdot J$

NOTA – Outros métodos de seleção do parâmetro k não são considerados compatíveis com esta Recomendação para Normalização.

4.3.2.12 O método usado para selecionar o parâmetro k será indicado na Parte 3 do Cabeçalho do Segmento, quando a Parte 3 do Cabeçalho do Segmento é incluída.

NOTA – Desde que o valor do parâmetro k é explicitamente codificado na seqüência de dados comprimidos, não é necessário o decodificador saber se o método de seleção ótimo ou heurístico foi usado.

4.3.2.13 Quando o método ótimo de seleção da opção de código é usado, as seguintes regras são aplicadas quando a opção de código ótima não é única:

- A opção ‘não codificado’ será selecionada se ela minimizar o número de bits codificados, mesmo se outra opção der o mesmo número de bits.
- Quando dois ou mais parâmetros do código minimizarem o número de bits codificados, a menor opção de parâmetro do código será selecionada.

4.3.3 PLANOS DE BITS ADICIONAIS DE COEFICIENTES DC

NOTA – A codificação dos coeficientes DC quantizados, descrita em 4.3.2, codifica de forma eficaz os primeiros $\text{BitDepthDC}-q$ bits de cada coeficiente DC.

4.3.3.1 Quando $q > \text{BitDepthAC}$, os próximos $q-\text{BitDepthAC}$ bits de cada coeficiente DC deverão aparecer na seqüência de bits (ou, *bitstream*) codificada imediatamente seguindo os coeficientes DC quantizados codificados do segmento (descrito em 4.3.2).

4.3.3.2 Os bits apropriados de cada coeficiente DC em um segmento devem ser concatenados com cada plano de bits; isto é, o $(q-1)$ -ésimo bit mais significativo de cada coeficiente DC deve ser seguido pelo $(q-2)$ -ésimo bit mais significativo de cada coeficiente DC, todos não codificados, e assim por diante, até o BitDepthAC-ésimo bit de cada coeficiente DC.

4.4 ESPECIFICANDO A PROFUNDIDADE DE BITS AC EM CADA BLOCO

O primeiro passo na codificação das magnitudes do coeficiente AC em um segmento é especificar a seqüência de valores BitDepthAC_Block_m para o segmento. Esses valores serão codificados em uma das três formas, dependendo do valor de BitDepthAC que é codificado na Parte 1A do cabeçalho do segmento:

- a) Se BitDepthAC é 'zero', então a seqüência de valores BitDepthAC_Block_m é zero e, assim, não será codificada. Além disso, nenhuma codificação do plano de bits dos coeficientes AC deve ser realizada uma vez que todos os coeficientes AC no segmento precisam ser zero.
- b) Se BitDepthAC é 'um', então cada valor BitDepthAC_Block_m é zero ou um, e assim, cada valor BitDepthAC_Block_m deve ser codificado usando um único bit para indicar seu valor; i.e., os valores BitDepthAC_Block_m de bit-único devem ser concatenados.
- c) Por outro lado, a seqüência de valores BitDepthAC_Block_m para o segmento será codificada usando o mesmo procedimento de diferenciação (ou, *differencing*) e codificação de comprimento variável, especificado para codificação dos valores DC quantizados, descrito em 4.3.2. No entanto, alguns parâmetros diferem no caso de codificação dos valores BitDepthAC_Block_m:
 - 1) N deve ser igual ao número de bits necessários para representar a magnitude de BitDepthAC para o segmento, como na equação 20:

$$N = \lceil \log_2(1 + \text{BitDepthAC}) \rceil \quad (20)$$

Para identificação da opção de código, a tabela 4-9 será usada com esse valor de N .

NOTA – Para uma imagem fonte de 16-bits, não mais do que cinco bits são necessários para especificar o valor de BitDepthAC, i.e., $N \leq 5$.

- 2) Desde que os valores BitDepthAC_Block_m são necessariamente não negativos, x_{\min} e x_{\max} , devem ser redefinidos como mostrado na equação 21 e usados para mapear diferenças sucessivas para inteiros não negativos:

$$x_{\min} = 0, \quad x_{\max} = 2^N - 1 \quad (21)$$

NOTA – A seqüência (ou, *string*) de bits codificados para os valores BitDepthAC_Block_m segue o mesmo formato de dados ilustrado nas figuras 4-8 e 4-9.

4.5 CODIFICAÇÃO DE PLANO DE BITS

4.5.1 VISÃO GERAL

A codificação de um plano de bits é realizada em *estágios* numerados de 0 a 4. Os bits codificados, produzidos nos estágios para cada bloco, são intercalados, como ilustrado na figura 4-2(b) e na figura 4-10. Assim, um plano de bits codificado primeiro consiste em todos os bits do estágio 0 no segmento, então todos os bits codificados do estágio 1 no segmento, e assim por diante, terminando com todos os bits codificados do estágio 4 no segmento. Isso produz uma cadeia integrada de bits (ou, *embedded bit string*) com informação do mais alto plano de bits, de todos os S blocos, na primeira parte da cadeia (ou, *string*) de bits de saída, seguido pela informação do mais baixo plano de bits, e permite decodificação progressiva da cadeia (ou, *string*) codificada. Isso melhora a qualidade da reconstrução da imagem quando a seqüência de bits codificados é truncada.

Bits do estágio 0 provenientes de cada bloco no segmento
Estágio 1 codificado a partir do bloco 0, 1, ..., $S-1$
Estágio 2 codificado a partir do bloco, 1, ..., $S-1$
Estágio 3 codificado a partir do bloco, 1, ..., $S-1$
Estágio 4 codificado a partir do bloco, 1, ..., $S-1$

Figura 4-10: Estrutura do Plano de Bits Codificada para um Segmento

Observe que quando o índice b do plano de bits que está sendo codificado é maior ou igual profundidade (ou, *depth*) do bit AC do bloco, então, não há nada a codificar para o bloco. Os estágios de codificação em camadas (ou, *layered coding stages*) na figura 4-10 inerentemente permitem um controle da quantidade de informação do coeficiente da DWT a ser codificada dentro de um plano de bits, permitindo um controle da qualidade da imagem em níveis de subplano de bits. A relação (ou, *tradeoff*) entre a qualidade da imagem reconstruída e o volume de dados comprimidos para cada segmento é controlado especificando-se os primeiros quatro parâmetros da Parte 2 do Cabeçalho (tabela 4-5). O nível de qualidade e volume de dados são ambos especificados para cada segmento, e a saída comprimida, para um segmento, é produzida até que o limite da qualidade ou o limite do volume seja alcançado. Os usuários podem preferir alcançar uma taxa de saída fixa usando bits de enchimento (ou, *fill bits*). Estes atributos são descritos em 4.2.3.

Anexo 0 contém uma lista de símbolos usados nos vários estágios de codificação especificados em 4.5.2, 4.5.3, e 4.5.4.

4.5.2 VISÃO GERAL DOS ESTÁGIOS DE CODIFICAÇÃO

Os estágios de codificação para um bloco no plano de bits b são descritos nos parágrafos seguintes.

Estágio 0 para um bloco consiste no máximo de um único bit, o qual é, simplesmente, o bit mais significativo b -ésimo da representação em complemento de 2 do coeficiente DC. Note que se o índice do plano de bits, b , satisfaz $b \geq q$, esse valor do bit já é conhecido a partir da

informação do coeficiente DC já codificado, e nesse caso, o estágio 0 é vazio, isto é, nenhum bit é codificado no estágio 0. O estágio 0 é também vazio quando o escalonamento do coeficiente DC assegura que o bit deve ser o zero, isto é, quando $b < \text{BitShift}(\text{LL}_3)$.

Os estágios restantes (1-4) codificam os bits do coeficiente AC. O estágio, no qual os bits dos coeficientes AC em um plano de bits são codificados, depende do *tipo* (ou, *type*) do coeficiente AC no plano de bits, o qual será definido agora. Em um plano de bits b , o tipo de um coeficiente AC x , denominado $t_b(x)$, tem um dos seguintes valores:

- $t_b(x) = 0$ se $|x| < 2^b$, (x não é adequado para seleção nesse plano de bits);
- $t_b(x) = 1$ se $2^b \leq |x| < 2^{b+1}$, (x é adequado para seleção nesse plano de bits);
- $t_b(x) = 2$ se $2^{b+1} \leq |x|$, (x já foi selecionado em um plano de bits anterior);
- $t_b(x) = -1$ se $b < \text{BitShift}(\Gamma)$, (x precisa ser zero nesse plano de bits devido ao escalonamento da subbanda).

Aqui, Γ denota a subbanda contendo x . Assim, durante a codificação do plano de bits, cada coeficiente AC tipicamente prossegue do tipo 0 para 1, para 2, para -1. Para um conjunto de coeficientes Ψ , define-se $t_{\max}(\Psi)$ como o máximo dos tipos de coeficiente em Ψ .

Um coeficiente AC x é dito estar *selecionado* no plano de bits b se $t_b(x) = 1$. Ou seja, a ‘seleção’ de um coeficiente marca o primeiro plano de bits onde um bit de magnitude não zero é codificado para o coeficiente. Note que $t_b(x) = 1$ se o b -ésimo bit de magnitude mais significativa de x é igual a ‘1’, e todos os bits de x de magnitude significativa são iguais a ‘0’.

O tipo de um coeficiente determina o estágio no qual a codificação de um bit coeficiente ocorre. Quando um coeficiente AC ‘ x ’ é do tipo 0 ou 1 (implicando em $t_{b+1}(x)=0$), o b -ésimo bit de x de magnitude mais significativo é codificado nos estágios de 1 a 3. Por outro lado, o bit é incluído, e descomprimido (ou, *uncompressed*), no estágio 4 se x é do tipo 2, ou não é codificado de nenhuma forma quando x é do tipo -1.

Nos **estágios de 1-3** de codificação BPE do plano de bits b , o b -ésimo bit de magnitude de cada coeficiente AC x , tal que $t_{b+1}(x)=0$, é codificado. O b -ésimo bits de magnitude dos coeficientes mãe são codificados no estágio 1, os filhos no estágio 2, e os netos no estágio 3. Cada um desses estágios também inclui bits codificados que indicam o sinal de cada coeficiente x para o qual $t_b(x)=1$. A codificação nos estágios 1-3 faz uso da estrutura de família para agrupar os coeficientes AC para codificação por entropia.

A codificação realizada nos estágios 1-3 para um bloco consiste em duas partes. Primeiro, uma seqüência de palavras binárias de comprimento variável é definida, tal que, descrevem completamente os bits a serem codificados nesses estágios, como discutido em 4.5.3.1. A seguir, um subconjunto dessas palavras é também codificado por entropia, como descrito em 4.5.3.3.

O estágio 4 de codificação consiste no b -ésimo bit de magnitude de cada coeficiente AC x com $t_b(x)=2$. Esses bits são incluídos na seqüência de dados codificados descomprimidos (ou, *uncompressed*), na ordem especificada em 4.5.4.

4.5.3 ESTÁGIOS DE CODIFICAÇÃO DE 1 A 3

4.5.3.1 Palavra do Coeficiente AC

NOTA – Os bits codificados nos estágios de 1-3 para um bloco podem ser determinados por uma seqüência de palavras, como descrito abaixo. Algumas dessas palavras são também codificadas por entropia, como descrito em 4.5.3.3.

4.5.3.1.1 Além dos conjuntos C_i , G_i , H_{ij} , definidos em 4.1, P é definido como a lista de ‘mães’ (ou, *parents*) no bloco:

$$P = \{p_0, p_1, p_2\}.$$

4.5.3.1.2 A lista de descendentes na família i , denominada D_i , é definida como

$$D_i = \{C_i, G_i\}.$$

4.5.3.1.3 A lista de descendentes em um bloco, denominada B , é definida como

$$B = \{D_0, D_1, D_2\}.$$

NOTA – $\{A, B\}$ denota a concatenação das listas A e B .

4.5.3.1.4 Uma notação abreviada para certas palavras binárias que descrevem informações acerca do plano de bits b , para uma lista de coeficientes Ψ , é definida como segue:

- seja $types_b[\Psi]$ a palavra binária que consiste do b -ésimo bit de magnitude de cada coeficiente x em Ψ , tal que, $t_b(x)$ é igual 0 ou 1;
- seja $signs_b(\Psi)$ a palavra binária que consiste do bit de sinal de cada coeficiente x em Ψ , tal que, $t_b(x) = 1$, com um bit de sinal igual a ‘1’ para coeficientes negativos e ‘0’ para coeficientes não negativos;
- dada uma lista de tipos (ou, *type values*) $\Lambda = \{\lambda_0, \lambda_1, \lambda_2, \dots, \lambda_l\}$, seja $tword[\Lambda]$ a palavra binária que consiste da seqüência de valores do tipo λ_i em Λ que são iguais a 0 ou a 1.

NOTA – Qualquer dessas palavras pode ser nula (isto é, ter comprimento zero).

4.5.3.1.5 A lista P deve ser ordenada $P = \{p_0, p_1, p_2\}$, enquanto que a ordem nas listas C_i e H_{ij} será determinada pela ordem, na qual as coordenadas dos seus coeficientes membros são listados na tabela 4-1.

4.5.3.1.6 Os b -ésimo bits de magnitude para todos os coeficientes AC que são do tipo 0 no plano de bits $b+1$ (isto é, não selecionado antes do plano do bit atual) devem ser enviados ao decodificador agrupados para formar palavras binárias associadas a cada tipo (ou, *type*) de dado (mãe, filho, neto):

- $types_b[P]$;

- $types_b[C_i]$ para $i = 0, 1, 2$; e
- $types_b[H_{ij}]$ para $i = 0, 1, 2, j = 0, 1, 2, 3$.

4.5.3.1.7 Nos planos de bit iniciais, muitos conjuntos de coeficientes em um bloco tendem a ser todos do tipo 0 (ou, *type 0*), e assim, muitas destas palavras são, inicialmente, zero. Nesse caso, para uma codificação eficaz, o BPE deve fazer uso das seguintes palavras de *transição* (ou, *transition words*) para indicar quando grupos de coeficientes em uma profundidade mais baixa são todos do Tipo 0 (ou, *Type 0*):

- $tran_B = \begin{cases} \text{nulo,} & \text{se } tran_B = 1 \text{ em qualquer plano de bits mais} \\ & \text{significativo,} \\ \text{tword}[\{t_{\max}(B)\}], & \text{de outra maneira;} \end{cases}$
- $tran_D = \text{tword}[\{t_{\max}(D_i) : i=0,1,2, \text{ tal que, } t_{\max}(D_i) \neq 1 \text{ em todos os planos de bit mais significativos}\}];$
- $tran_G = \text{tword}[\{t_{\max}(G_i) : eu=0, 1, 2, \text{ tal que, } t_{\max}(D_i) > 0 \text{ no atual ou em qualquer plano de bits mais significativo}\}];$
- $tran_{Hi} = \text{tword}[\{t_{\max}(H_{i0}), t_{\max}(H_{i1}), t_{\max}(H_{i2}), t_{\max}(H_{i3})\}]$ para $i = 0, 1, 2$.

4.5.3.1.8 No plano de bits b , o BPE deve usar a seguinte seqüência de palavras, geradas nos três estágios, para atualizar todos os coeficientes AC no bloco que eram do Tipo 0 no plano de bits anterior:

a) **Estágio 1 (mãe):**

$types_b[P], signs_b[P]$.

b) **Estágio 2 (filhos):**

1) $tran_B$;

2) $tran_D$, se $tran_B \neq 0$ e $t_{\max}(B) \neq -1$;

3) $types_b[C_i]$ e $signs_b(C_i)$ para cada i , tal que, $t_{\max}(D_i) \neq 0, -1$.

c) **Estágio 3 (netos):**

Se $tran_B = 0$ ou $t_{\max}(B) = -1$, então, o estágio 3 é desnecessário e será omitido. Por outro lado, o estágio 3 consiste em:

1) $tran_G$;

2) $tran_{Hi}$, para cada i , tal que, $t_{\max}(G_i) \neq 0, -1$;

3) $types_b[H_{ij}]$ e $signs_b[H_{ij}]$ para cada i , tal que, $t_{\max}(G_i) \neq 0, -1$ e cada um j , tal que, $t_{\max}(H_{ij}) \neq 0, -1$.

NOTA – Todas as palavras geradas nos estágios a cima são de comprimento variável (incluindo a palavra nulo (ou, *null word*)).

4.5.3.1.9 As palavras $types_b[P]$, $types_b[C_i]$, $types_b[H_{ij}]$, $tran_D$, $tran_G$, $tran_{Hi}$ serão codificadas por entropia, isto é, cada uma será substituída por uma palavra código (ou, *codeword*) de comprimento variável correspondente, sempre que uma palavra tiver comprimento de pelo menos 2 bits.

NOTA – As palavras de bit de sinalização (ou, *sign bit*) não são codificadas, além disso, porque os coeficientes AC são geralmente positivos e negativos com probabilidade aproximadamente igual. A palavra $tran_B$ é sempre, no máximo, um bit em comprimento e nunca é codificada por entropia.

4.5.3.2 Mapeamento de Palavras para Símbolos

4.5.3.2.1 O procedimento de codificação por entropia usado para codificar as palavras $types_b[P]$, $types_b[C_i]$, $types_b[H_{ij}]$, $tran_D$, $tran_G$, $tran_{Hi}$ deve ser realizado usando códigos de comprimento variável, dados em 4.5.3.3. Palavras com um comprimento de um bit, e palavras de bit de sinal, devem ser incluídas na seqüência de dados comprimidos sem codificação adicional. Palavras de comprimento maior do que um bit devem ser codificadas na seqüência na qual elas ocorrem dentro de cada estágio com o método de codificação por entropia, descrito em 4.5.3.3.

NOTA – Certas seqüências de bit não podem aparecer como valores para algumas palavras e esse fato é levado em conta no processo de codificação por entropia. Por exemplo, $tran_D$ não pode nunca ser igual a 000, pois essa condição seria definida pelo fato que $tran_B=0$. A Tabela 4-11 resume os comprimentos máximos da palavra e os valores impossíveis para cada palavra que é codificada por entropia.

Tabela 4-11: Sumário de Comprimentos Máximos da Palavra e Valores Impossíveis da Palavra

Palavra	Comprimento Máximo (bits)	Valor Impossível
$types_b[P]$	3	–
$types_b[C_i]$	4	–
$types_b[H_{ij}]$	4	0000
$tran_D$	3	000
$tran_G$	3	–
$tran_{Hi}$	4	0000

4.5.3.2.2 O processo de codificação de comprimento variável dessas palavras deve seguir um processo de dois passos:

- primeiro, os valores da palavra devem ser mapeados para valores inteiros conhecidos como *símbolos*; então
- cada inteiro deve ser codificado usando uma palavra código (ou, *codeword*) binária de comprimento variável, como descrito em 4.5.3.3.

4.5.3.2.3 Durante o mapeamento, palavras de dois-bits, três-bits, quatro-bits devem ser mapeadas para símbolos usando as tabelas 4-12, 4-13 ou 4-14, respectivamente.

NOTA – O processo de mapeamento leva em conta o fato que certas palavras nunca podem ser atribuídas a algumas seqüências de bit, como tabelado em 4-11; isso repercute nas tabelas 4-13 e 4-14.

Tabela 4-12: Mapeamento de Inteiros para Palavras de Dois-Bits

Palavra	Símbolo
00	0
01	2
10	1
11	3

Tabela 4-13: Mapeamento de Inteiros para Palavras de Três-Bits

Palavra	Símbolo ($types_b[P]$, $types_b[C]$, $types_b[H_{ij}]$, $tran_G$, $tran_{Hi}$)	Símbolo ($tran_D$)
000	1	-
001	4	3
010	0	0
011	5	4
100	2	1
101	6	5
110	3	2
111	7	6

Tabela 4-14: Mapeamento de Inteiros para Palavras de Quatro-Bits

Palavra	Símbolo ($types_b[C_i]$)	Símbolo ($types_b[H_{ij}]$, $tran_{Hi}$)
0000	10	-
0001	1	1
0010	3	3
0011	6	6
0100	2	2
0101	5	5
0110	9	9
0111	12	11
1000	0	0

1001	8	8
1010	7	7
1011	13	12
1100	4	4
1101	14	13
1110	11	10
1111	15	14

NOTA – Os mapeamentos destinam-se a produzir valores de símbolo em ordem de frequência decrescente. (Ou seja, a palavra que ocorre mais freqüentemente é mapeada para valor de símbolo 0, a próxima mais freqüente para 1, etc.). Isso torna a codificação eficaz possível, usando o procedimento de codificação descrito em 4.5.3.3, pois as palavras código (ou, *codewords*) são arranjadas em ordem crescente de comprimento.

4.5.3.3 Codificação por Entropia dos Símbolos

4.5.3.3.1 Os símbolos de 4.5.3.2 devem ser codificados usando códigos binários de comprimento variável, como mostrado nas tabelas 4-15, 4-16, e 4-17.

4.5.3.3.2 Dentro de um conjunto (ou, *gaggle*),

- todas as palavras de dois-bits devem ser codificadas usando uma das duas opções de código de comprimento variável mostradas na tabela 4-15;
- todas as palavras de três-bits devem ser codificadas usando uma das três opções de código de comprimento variável mostradas na tabela 4-16;
- todas as palavras de quatro-bits devem ser codificadas usando uma das quatro opções de código de comprimento variável mostradas na tabela 4-17.

4.5.3.3.3 Os códigos de comprimento variável usados permitem alteração em cada conjunto (ou, *gaggle*).

Tabela 4-15: Opções de Código de Comprimento Variável para Palavras de Dois-Bits

Símbolo de Entrada	Opção de Código 0	Opção de Código 1 (não codificado)
0	1	00
1	01	01
2	001	10
3	000	11

Tabela 4-16: Opções de Código de Comprimento Variável para Palavras de Três-Bits

Símbolo de Entrada	Opção de Código 0	Opção de Código 1	Opção de Código 3 (não codificado)
0	1	10	000
1	01	11	001
2	001	010	010
3	0000	011	011
4	00001	0010	100
5	00010	0011	101
6	000110	0000	110
7	000111	0001	111

Tabela 4-17: Opções de Código de Comprimento Variável para Palavras de Quatro-Bits

Símbolo da Entrada	Opção de Código 0	Opção de Código 1	Opção de Código 2	Opção de Código 3 (não codificado)
0	1	10	100	0000
1	01	11	101	0001
2	001	010	110	0010
3	0001	011	111	0011
4	0000000	0010	0100	0100
5	0000001	0011	0101	0101
6	0000010	000000	0110	0110
7	0000011	000001	0111	0111
8	00001000	000010	00100	1000
9	00001001	000011	00101	1001
10	00001010	000100	00110	1010
11	00001011	000101	00111	1011
12	00001100	0001100	00000	1100
13	00001101	0001101	00001	1101
14	00001110	0001110	00010	1110
15	00001111	0001111	00011	1111

4.5.3.3.4 Para cada tamanho da palavra dentro de um conjunto (ou, *gaggle*) (i.e., palavras de dois-bits, três-bits, e de quatro-bits), a opção de código selecionada deve ser aquela que minimizará o comprimento codificado do conjunto (ou, *gaggle*) quando todas as palavras são codificadas. As otimizações de parâmetro do código são realizadas sobre todas as palavras no conjunto (ou, *gaggle*) mesmo quando a qualidade do modo de saída especificado

assegurar que algumas das palavras não estão incluídas na seqüência de bits (ou, *bit stream*) comprimido. A opção do código selecionada deve ser indicada usando os bits de ID (identificação) que empregam os valores da tabela 4-18.

NOTA – No máximo, cinco bits por conjunto (ou, *gaggle*) por plano de bits são necessários para especificar as opções do código.

4.5.3.3.5 A opção não codificado (ou, *uncoded*) é selecionada sempre que minimizar o número de bits codificados, mesmo que outra opção dê o mesmo número de bits. Quando dois ou mais parâmetros de código minimizarem o número de bits codificados, a opção de menor parâmetro de código (ou, *code parameter*) será selecionada.

Tabela 4-18: Identificação das Opções de Código de Comprimento Variável

No. de Bits para Padrões Mapeados	No. de Bits de ID	ID e Opção de Código Associada
2	1	0: opção de código 0 1: não codificado (ou, <i>uncoded</i>)
3	2	00: opção de código 0 01: opção de código 1 11: não codificado (ou, <i>uncoded</i>)
4	2	00: opção de código 0 01: opção de código 1 10: opção de código 2 11: não codificado (ou, <i>uncoded</i>)

4.5.3.3.6 Os bits de identificação ID para especificar opções de codificação para palavras de um dado comprimento devem ser inseridos imediatamente antes do primeiro aparecimento de uma palavra código (ou, *codeword*) para um dado comprimento dentro de um conjunto (ou, *gaggle*). Quando nenhuma palavra de um dado comprimento ocorre em um conjunto (ou, *gaggle*), nenhum bit de identificação ID está presente.

4.5.3.3.7 A palavra codificada deve substituir a palavra não codificada apropriada, identificada em 4.5.3.1.

4.5.4 ESTÁGIO 4 DE CODIFICAÇÃO

4.5.4.1 No estágio 4 de codificação, o *b*-ésimo bit de magnitude de cada coeficiente AC *x* com tipo $t_b(x)=2$ deve ser incluído na saída da seqüência de bits (ou, *bit stream*).

4.5.4.2 Para cada bloco, a seqüência de bits de saída (ou, *output bit string*) consiste no *b*-ésimo bit de magnitude do tipo 2 coeficientes, na seguinte ordem:

- P_i , para cada $i = 0,1,2$;

- membros de C_i , para cada $i = 0,1,2$;
- membros de H_{ij} , para cada $i = 0,1,2$, e cada $j = 0,1,2,3$.

4.5.4.3 Membros dos conjuntos C_i e H_{ij} devem ser processados na ordem listada na tabela 4-1. Nenhum bit deve ser codificado no estágio 4, para coeficientes AC x que não sejam do tipo 2 ($t_b(x) \neq 2$).

4.5.4.4 As seqüências (ou, *strings*) resultantes para todos os blocos no segmento devem ser concatenadas para produzir uma seqüência de saída do estágio 4 inteiro para o segmento.

5 SEGURANÇA

5.1 INTRODUÇÃO

Essa seção apresenta os resultados de uma análise de considerações sobre segurança aplicadas às tecnologias especificadas nesta Recomendação para Normalização.

5.2 CONSIDERAÇÕES DE SEGURANÇA COM RELAÇÃO A ESTA RECOMENDAÇÃO DE NORMA

5.2.1 PRIVACIDADE DOS DADOS

A privacidade dos dados comprimidos, de acordo com as especificações desta Recomendação de Norma, deveria ser assegurada pelos sistemas e redes nos quais esta Recomendação de Norma está implementada.

5.2.2 INTEGRIDADE DOS DADOS

A integridade de dados comprimidos, de acordo com as especificações desta Recomendação de Norma, deveria ser assegurada pelos sistemas e redes nos quais esta Recomendação de Norma está implementada.

5.2.3 AUTENTICAÇÃO DAS ENTIDADES DE COMUNICAÇÃO

A autenticação de entidades de comunicação envolvidas no transporte de dados comprimidos, de acordo com as especificações desta Recomendação de Norma, deveria ser fornecida pelos sistemas e redes nos quais esta Recomendação de Norma está implementada.

5.2.4 CONTROLE DE ACESSO AOS RECURSOS

Esta Recomendação de Norma supõe que o controle de acesso aos recursos será gerenciado pelos sistemas nos quais as entidades compressora e descompressora residem.

5.2.5 DISPONIBILIDADE DE RECURSOS

Esta Recomendação de Norma supõem uma disponibilidade suficiente de recursos nos sistemas nos quais as entidades compressora e descompressora residem.

5.2.6 AUDITORIA DO USO DE RECURSOS

Esta Recomendação de Norma supõe que a revisão do uso de recursos será tratada de pela administração dos sistemas e redes nos quais esta Recomendação de Norma está implementada.

5.3 AMEAÇAS POTENCIAIS E CENÁRIOS DE ATAQUE

Não há conhecimento de ameaças em potencial ou cenários de ataque que apliquem, especificamente, as tecnologias especificadas nesta Recomendação de Norma. Ameaças em potencial ou cenários de ataque aplicáveis aos sistemas e redes nos quais esta Recomendação de Norma é implementada deveriam ser tratados pela administração daqueles sistemas e redes.

5.4 CONSEQÜÊNCIAS DO NÃO EMPREGO DE SEGURANÇA PARA A TECNOLOGIA

Há não conseqüências conhecidas de não aplicar segurança às tecnologias especificadas nesta Recomendação de Norma. As conseqüências de não se aplicar segurança aos sistemas e redes, nos quais esta Recomendação de Norma é implementada, inclui a perda potencial, a corrupção e o roubo de dados.

ANEXO A

GLOSSÁRIO DE ACRÔNIMOS E TERMOS

(Informativo)

BPE	codificador de plano de bits: algoritmo de processamento recomendado usado para comprimir dados de coeficientes wavelet.
DWT	Transformada Wavelet Discreta (ou, <i>Discrete Wavelet Transform</i>): algoritmo de processamento recomendado para transformar os dados de imagem em coeficientes wavelet.
CCD	Dispositivo Acoplado de Carga (ou, <i>Charge Coupled Device</i>), um sensor de imagem.
DWT 9/7	Uma DWT usando um filtro de 9 coeficientes (ou, <i>tap</i>) para obter os coeficientes wavelet passa-baixa e um filtro de 7 coeficientes (ou, <i>tap</i>) para obter os coeficientes wavelet passa-alta. Duas Transformadas Wavelet Discretas 9/7 específicas e diferentes são recomendadas: DWT 9/7 de ponto flutuante para compressão com perdas (ou, <i>lossy</i>) e DWT 9/7 de Inteiros para compressão sem perdas (ou, <i>lossless</i>).
DWT 9/7 de Ponto Flutuante	DWT 9/7 de três níveis cujos coeficientes do filtro são especificados como números reais. Implementada usando aritmética de ponto flutuante. Cf. equação (3) e tabela 3-1.
DWT 9/7 de Inteiros	DWT 9/7 de três níveis cujos coeficientes do filtro são especificados como números racionais. Implementada usando aritmética de inteiros. Cf. equações (5) e (6).
MSB	Bit Mais Significativo (de uma palavra): bit mais à esquerda nas figuras: mais alta potência de dois na representação binária; primeiro bit a ser transmitido em uma saída serial.
LL_n	Subbanda 2-D no n -ésimo nível da DWT; para $n=1$ obtido depois de aplicar seqüencialmente o filtro DWT 1-d passa-baixa às linhas horizontais e verticais da imagem; para $n>1$ obtido depois de aplicar seqüencialmente o filtro DWT 1-d passa-baixa às linhas horizontais e verticais da imagem linhas da subbanda LL_{n-1} .
LH_n	Subbanda 2-d no n -ésimo nível da DWT; para $n=1$ obtido depois de aplicar seqüencialmente o filtro DWT 1-d passa-baixa às linhas horizontais e o filtro DWT 1-d passa-alta às linhas verticais da imagem; para $n>1$ obtido depois de aplicar seqüencialmente o filtro

	DWT 1-d passa-baixa às linhas horizontais, e filtro DWT 1-d passa-alta às linhas verticais da subbanda LL_{n-1} .
HL_n	Subbanda 2-d no n -ésimo nível da DWT; para $n=1$ obtido depois de aplicar seqüencialmente filtro passa-alta 1-d às linhas horizontal e vertical da imagem; para $n>1$ obtido após seqüencialmente aplicar filtro DWT 1-d passa-alta às linhas horizontal e vertical da subbanda LL_{n-1} .
HH_n	Subbanda 2-d no n -ésimo nível da DWT; para $n=1$ obtido depois de aplicar seqüencialmente filtro DWT 1-d passa-alta às linhas horizontais, filtro DWT 1-d passa-baixa às linhas verticais da imagem; para $n>1$ obtido depois de aplicar seqüencialmente filtro DWT 1-d passa-alta às linhas horizontais, e filtro DWT 1-d passa-baixa às linhas verticais da subbanda LL_{n-1} .
Fatores de Peso	depois de transformar os dados de imagem por meio da DWT 9/7 de inteiros, os coeficientes da wavelet obtidos precisam ser multiplicados por esses números antes da codificação com o BPE. Um fator de peso é definido para subbanda cada.
DC coeficiente	qualquer coeficiente wavelet proveniente da subbanda de mais baixa freqüência LL_3 .
AC coeficiente	qualquer coeficiente wavelet proveniente de qualquer subbanda exceto a LL_3 .
bloco	Coleção de 64 coeficientes wavelet, sendo um coeficiente DC proveniente da subbanda LL_3 e 63 coeficientes AC exclusivamente associados provenientes das nove subbandas restantes (ver figura 4-1 e tabela 4-1). Para codificação, o domínio wavelet completo é dividido em blocos que são mutuamente separados (ou, <i>pairwise disjoint</i>). Há um bloco para cada coeficiente DC.
segmento	Seqüência de bits (ou, <i>Bitstream</i>) do código comprimido consistindo de um campo de dados guiado por um cabeçalho. O cabeçalho do segmento é definido em 4.2. O campo de dados contém os bits codificados a partir de S blocos consecutivos. S é um parâmetro selecionado pelo usuário, tal que, $16 \leq S \leq 2^{20}$.
conjunto	Um conjunto (ou, <i>gaggle</i>) consiste de uma série de blocos consecutivos dentro de um segmento. Especificamente, um segmento é dividido em conjuntos (ou, <i>gaggles</i>), com cada conjunto (ou, <i>gaggle</i>) consistindo de 16 blocos, exceto, possivelmente, o último conjunto (ou, <i>gaggle</i>), o qual contém $(S \text{ mod } 16)$ blocos quando S não é um múltiplo de 16 (ver 4.3.2.5). Certos parâmetros de código são selecionados independentemente para cada conjunto (ou, <i>gaggle</i>). Especificamente, todos os valores do coeficiente DC quantizado em

NORMA RECOMENDADA CCSDS PARA COMPRESSÃO DE DADOS DE IMAGEM

um conjunto (ou, *gaggle*) são codificados usando a mesma opção de código (ver 4.3.2.6). Da mesma forma, em cada plano de bits, o mesmo grupo de opções de código de comprimento variável é usado para todas as palavras de comprimento variável codificadas em um conjunto (ou, *gaggle*) (ver 4.5.3.3.3).

ANEXO B

REFERÊNCIAS INFORMATIVOS

(Informativo)

- [B1] *Image Data Compression*, Report concerning Space Data Systems Standards, CCSDS 123.0-G-1, Green Book. [No momento da publicação, esse documento estava em desenvolvimento.]
- [B2] *Lossless Data Compression*. Recommendation for Space Data Systems Standards. CCSDS 121.0-B-1. Blue Book. Issue 1. Washington D.C.: CCSDS, May 1997.
- [B3] *Space Packet Protocol*. Recommendation for Space Data Systems Standards. CCSDS 133.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, September 2003.
- [B4] *CCSDS File Delivery Protocol (CFDP)*. Recommendation for Space Data System Standards, CCSDS 727.0-B-3. Blue Book. Issue 3. Washington, D.C.: CCSDS, June 2005.
- [B5] *AOS Space Data Link Protocol*. Recommendation for Space Data System Standards, CCSDS 732.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, September 2003.
- [B6] *Information Technology—JPEG 2000 Image Codificação System: Core Codificação System*. International Standard, ISO/IEC 15444-1:2004. 2nd ed. Geneva: ISO, 2004.
- [B7] S.G. Mallat, “A Theory for Multiresolution Signal Decomposition: The Wavelet Representation,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, July 1989.

ANEXO C

SÍMBOLOS USADOS NOS ESTÁGIOS DE CODIFICAÇÃO

(Informativo)

Tabela C-1: Símbolos usados nos estágios de codificação na Seção 4

Categoria	Símbolos	Sentido
coeficientes wavelet	DC, c c'	coeficiente DC coeficiente DC quantizado
	AC, x ρ_i	coeficiente AC coeficiente mãe ($i=0,1,2$)
	Γ	subbanda DWT
grupos ordenados de coeficientes wavelet atribuídos a um bloco	Ψ B P D_i C_i G_i H_{ij}	variável estabelecida conjunto de todos os coeficientes descendentes conjunto de coeficientes mãe conjunto de coeficientes descendente ($i=0,1,2$) conjunto de coeficientes filhos ($i=0,1,2$) conjunto de coeficientes netos ($i=0,1,2$) subconjunto do conjunto netos G_i ($i=0,1,2; j=0,1,2,3$)
funções de valor inteiro	$t_b(x)$	valor indicando o tipo de coeficiente x , com relação ao plano de bits atual com índice b
	$t_{\max}(\Psi)$	$\max\{t_b(x):x \in \Psi\}$
	$types_b[\Psi]$	palavra binária cujos bits são tipo $t_b(x)$ do coeficientes $x \in \Psi$, ignorando os coeficientes com outros tipos $t_b(x)=0$ ou $t_b(x)=1$
	$signs_b[\Psi]$	palavra binária cujos bits são os sinais dos coeficientes $x \in \Psi$, ignorando os coeficientes cujos tipos não são $t_b(x)=1$. Sinais positivos dão 0, sinais negativos dão 1.
	$tword[\{t_0, t_1, \dots, t_i\}]$	palavras binárias cujos bits são valores do tipo $\{t_0, t_1, \dots, t_i\}$, ignorando tipos cujos valores não são 0 ou 1
palavras binárias	$tran_B$	palavra de transição para o conjunto B
	$tran_D$	palavra de transição para coleção de conjuntos D_0, D_1, D_2
	$tran_G$	palavra de transição para coleção de conjuntos G_0, G_1, G_2
	$tran_{H_i}$	palavra de transição para coleção de conjuntos H_0, H_1, H_2, H_3 ($i=0,1,2$)