



República Federativa do Brasil
Ministério do Desenvolvimento, Indústria
e do Comércio Exterior
Instituto Nacional da Propriedade Industrial

(21) **PI0614759-3 A2**

(22) Data de Depósito: 03/08/2006
(43) Data da Publicação: 12/04/2011
(RPI 2101)



(51) *Int.Cl.:*
H03M 7/46
H03M 7/40
H04N 7/12

(54) Título: **CODIFICAÇÃO E DECODIFICAÇÃO EFICIENTE DE BLOCOS DE TRANSFORMAÇÃO**

(30) Prioridade Unionista: 12/08/2005 US 11/203.008

(73) Titular(es): MICROSOFT CORPORATION

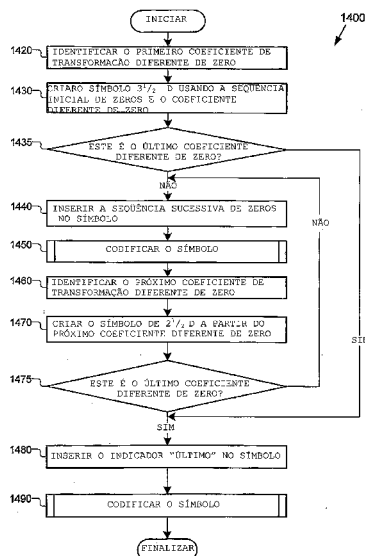
(72) Inventor(es): Sridhar Srinivasan

(74) Procurador(es): NELLIE ANNE DANIEL SHORES

(86) Pedido Internacional: PCT US2006030308 de 03/08/2006

(87) Publicação Internacional: WO 2007/021568 de 22/02/2007

(57) **Resumo:** CODIFICAÇÃO E DECODIFICAÇÃO EFICIENTE DE BLOCOS DE TRANSFORMAÇÃO. Trata-se de um codec de mídia digital baseado em blocos de transformação que codifica de modo mais eficiente coeficientes de transformação através da codificação conjunta de coeficientes diferentes de zero junto com sucessões subsequentes de coeficientes de valor igual a zero. Quando o coeficiente diferente de zero é o último de seu bloco, um último indicador é substituído pelo valor da sucessão no símbolo para aquele coeficiente. Coeficientes iniciais diferentes de zero são indicados em um símbolo especial que codifica em conjunto o coeficiente diferente de zero junto com as sucessões iniciais e subsequentes de valores iguais a zero. O codec ensina a codificação múltipla de contextos reconhecendo interrupções nas sucessões de coeficientes diferentes de zero em cada um dos lados da dita interrupção separadamente. O codec também reduz o tamanho da tabela de código indicando em cada símbolo se um coeficiente diferente de zero apresenta valor superior a 1 e se sucessões de zero apresentam valor positivo, codificando separadamente o nível dos coeficientes e o comprimento das sucessões fora dos símbolos.





"CODIFICAÇÃO E DECODIFICAÇÃO EFICIENTE DE BLOCOS
DE TRANSFORMAÇÃO"

ANTECEDENTES DA INVENÇÃO

CODIFICAÇÃO BASEADA EM BLOCOS DE TRANSFORMAÇÃO

5 A codificação por transformação é uma técnica de
compressão empregada em diversos sistemas de compressão de
áudio, vídeo e imagem. Imagens e vídeos digitais não
comprimidos são caracteristicamente representados ou
capturados como amostras de cores ou elementos de fotos em
10 locais dentro de um quadro de imagem ou de vídeo em rede
bidimensional (2D). A isso nos referimos como uma
representação de domínio espacial da imagem ou vídeo. Por
exemplo, um formato característico para imagens consiste de
um fluxo de amostras de elementos de 24 bits de imagem
15 colorida dispostas em uma grade. Cada amostra é um número
que representa os componentes coloridos em uma localização
de pixel na grade dentro de um padrão de cor, como o RGB ou
YIQ, dentre outros. Diversos sistemas de imagem e vídeo
podem utilizar várias e diferentes resoluções de tempo, cor
20 e espaço de amostragem. Da mesma forma, o áudio digital é
tipicamente representado como um fluxo do sinal de áudio
amostrado em intervalos de tempo. Por exemplo, um formato
de áudio típico consiste de um fluxo de amostras com
amplitude de 16 bits de um sinal de áudio capturado em
25 intervalos de tempo regulares.

Sinais não comprimidos de vídeo, imagem e áudio
digitais podem consumir considerável capacidade de
armazenamento e de transmissão. A codificação por
transformação reduz o tamanho do vídeo, das imagens e do

áudio digitais, ao transformar a representação de domínio espacial do sinal em representação de domínio de frequência (ou outro domínio de transformação similar), e em seguida reduzindo a resolução de determinados componentes da frequência geralmente menos perceptíveis da representação do domínio de transformação. Essa situação produz, geralmente, uma degradação menos perceptível do sinal digital comparado à redução da cor ou da resolução espacial de imagens ou vídeos no domínio espacial ou de áudio no domínio de tempo.

Mais especificamente, um codec típico baseado na transformação de bloco 100 apresentado na Figura 1 divide os pixels da imagem digital não comprimida em blocos bidimensionais de tamanho fixo $(X_1, \dots, X_n)$, sendo que cada bloco possivelmente se sobrepõe a outros blocos. Uma transformação linear 120-121 que executa a análise de frequência espacial é aplicada a cada bloco, o qual converte as amostras espaçadas dentro do bloco em um conjunto de coeficientes frequência (ou de transformação) que geralmente representam a força do sinal digital nas bandas de frequência correspondentes sobre o intervalo do bloco. Para a compressão, os coeficientes de transformação podem ser seletivamente quantificados 130 (i.e., reduzidos na resolução, eliminando bits menos significativos dos valores de coeficiente ou mapeando valores de outra forma em um número de resolução mais elevado ajustado para uma resolução mais baixa), e também a entropia ou um codificado de comprimento variável 130 em um fluxo de dados comprimidos. Na decodificação, os coeficientes de

transformação irão transformar inversamente 170-171 para reconstruir aproximadamente o sinal original de imagem/vídeo com amostragem de cor/espacial (blocos reconstruídos $X_1, \dots, X_n$).

5 O bloco de transformação 120-121 pode ser definido como uma operação matemática em um vetor x de tamanho N . Mais comumente, a operação é uma multiplicação linear, produzindo a saída de domínio de transformação $y = Mx$, sendo M a matriz de transformação. Quando os dados de
10 entrada são arbitrariamente longos, são segmentados em vetores de tamanho N , aplicando-se uma transformação por bloco a cada segmento. Para fins de compressão de dados, são escolhidos transformações de bloco reversíveis. Em outras palavras, a matriz M é invertível. Nas dimensões
15 múltiplas (por exemplo, para imagem e vídeo), as transformações por bloco são tipicamente implantadas como operações isoladas. Aplica-se a multiplicação da matriz separadamente ao longo de cada dimensão de dados (i.e., ambas as fileiras e colunas).

20 Para compressão, os coeficientes de transformação (componentes do vetor y) podem ser seletivamente quantificados (i.e., reduzidos na resolução, eliminando bits menos significativos dos valores de coeficiente ou mapeando valores de outra forma em um número de resolução
25 mais elevado ajustado para uma resolução mais baixa), e também a entropia ou comprimentos variáveis codificados em um fluxo de dados comprimidos.

Decodificando no decodificador 150, o inverso dessas operações (dequantização/decodificando a entropia

160 e transformação por bloco inversa 170-171) é aplicado no decodificador do lado 150, conforme demonstrado na Fig. 1. Enquanto reconstruindo os dados, a matriz inversa M^{-1} (transformação inversa 170-171) é aplicada como um

5 multiplicador aos dados do domínio de transformação. Quando aplicado aos dados do domínio de transformação, a transformação inversa reconstrói aproximadamente a mídia digital original do domínio do tempo ou do domínio espacial.

10 Em diversas aplicações de codificação baseada na transformação por bloco, a transformação é reversível e desejável, a fim de suportar a compressão com perda de dados e a compressão sem perda de dados, dependendo do fator de quantização. Sem quantização (geralmente

15 representada como um fator de quantização de 1), por exemplo, um codec utilizando uma transformação reversível pode reproduzir exatamente os dados de entrada na decodificação. Entretanto, a reversibilidade requerida nessas aplicações restringe a escolha de transformações

20 sobre as quais o codec pode ser projetado.

Numerosos sistemas de compressão de imagem e de vídeo, como o MPEG e o Windows Media, entre outros, empregam transformações baseadas na Transformação Discreta do Cosseno (DCT). A DCT é conhecida por suas propriedades

25 favoráveis de compactação de energia que proporcionam uma compressão de dados próxima ao ideal. Nesses sistemas de compressão, a DCT inversa (IDCT) é empregada nas malhas de reconstrução no codificador e no decodificador do sistema de compressão para reconstruir blocos de imagem individual.

Codificação de Entropia de Coeficientes de Transformação com Ampla Variação

Dados de entrada com ampla variação dinâmica conduzem a coeficientes de transformação com variação
5 dinâmica ainda maior, gerados durante o processo de codificação de uma imagem. Por exemplo, os coeficientes de transformação gerados por um N através de uma operação N DCT apresenta uma variação N vezes maior do que a variação
10 dinâmica dos dados originais. Com fatores de quantização pequenos ou unitários (empregados para efetuar compressão com pequena perda ou sem perda), a variação dos coeficientes de transformação quantificados também é grande. Estatisticamente, esses coeficientes apresentam uma
15 distribuição laplaciana conforme observado nas Figuras 2 e 3. A Figura 2 apresenta uma distribuição Laplaciana para coeficientes de ampla variação dinâmica. A Figura 3 apresenta uma distribuição laplaciana para coeficientes característicos de menor variação dinâmica.

A codificação por transformação convencional é
20 ajustada para uma pequena variação dinâmica de dados de entrada (tipicamente 8 bits), e quantificadores relativamente grandes (como valores numéricos de 4 e acima). A Figura 3, portanto, é representativa da distribuição dos coeficientes de transformação na dita
25 codificação de transformação convencional. Ainda mais, a codificação de entropia empregada com a dita codificação de transformação convencional pode ser uma variante da codificação do nível de execução, onde uma sucessão de zeros é codificada junto com um símbolo diferente de zero.

Esse pode ser um meio efetivo de representar execuções de zeros (que ocorre com alta probabilidade), assim como de capturar correlações inter-símbolos.

Por outro lado, a codificação por transformação convencional é menos adequada para se comprimir distribuições de ampla variação dinâmica tais como as apresentadas na Figura 2. Embora os símbolos sejam zero, com maior probabilidade do que qualquer outro valor (i.e., os picos de distribuição em zero), a probabilidade de um coeficiente ser exatamente zero é diminuta para a distribuição de ampla variação dinâmica. Como consequência, os zeros não ocorrem com muita regularidade, e as técnicas de codificação de entropia de comprimento de execução baseadas no número de zeros entre coeficientes sucessivos diferentes de zero são bastante ineficientes para dados de entrada de ampla variação dinâmica.

A distribuição de ampla variação dinâmica também apresenta um alfabeto crescente de símbolos, quando comparada com a distribuição de estreita variação. Em função desse crescente alfabeto de símbolos, a tabela de entropia utilizada para codificar os símbolos precisará ser maior. Caso contrário, muitos símbolos não serão codificados, o que é ineficiente. As tabelas maiores requerem mais memória e também podem resultar em maior complexidade.

Portanto, a codificação por transformação convencional carece de versatilidade - trabalhando satisfatoriamente para os dados de entrada com distribuição de variação dinâmica estreita, mas não para a distribuição

com ampla variação dinâmica.

No entanto, nos dados de variação estreita, encontrar codificação de entropia eficiente de coeficientes de transformação quantificados é um processo crítico.

5 Qualquer ganho de desempenho que possa ser adquirido nessa etapa (ganhos em termos de eficiência de compressão e de velocidade de codificação/decodificação) se traduz por ganhos de qualidade geral.

Modelos diferentes de codificação de entropia são
10 marcados por sua habilidade de se beneficiar positivamente dos ditos critérios de eficiência díspares como: o uso de informação contextual, compressão maior (como a codificação aritmética), requerimentos computacionais inferiores (como encontrado nas técnicas de codificação de Huffman), e
15 empregando um conjunto conciso de tabelas de código com o objetivo de minimizar a sobrecarga (*overhead*) da memória do codificador/decodificador. Métodos de codificação de entropia convencionais, que não se ajustam a todas essas características, não demonstram eficiência completa de
20 codificação de coeficientes de transformação.

SUMÁRIO

Uma técnica para decodificação e codificação de uma mídia digital e a realização da técnica em um codec de mídia digital descrito no presente atinge maior eficiência
25 de compressão dos coeficientes de transformação. Por exemplo, um codec exemplificativo de mídia digital baseado na transformação de bloco ilustrado no presente codifica com maior eficiência coeficientes de transformação codificando em conjunto coeficientes diferentes de zero

junto das execuções seguintes de coeficientes de valor igual a zero. Quando um coeficiente diferente de zero ocupa a última posição no bloco, um último indicador é substituído pelo valor de execução no símbolo relativo àquele coeficiente. Coeficientes iniciais diferentes de zero são indicados por um símbolo especial que codifica em conjunto o coeficiente diferente de zero junto com as execuções iniciais e subseqüentes dos zeros.

O codec exemplificativo permite contextos de codificação múltipla reconhecendo interrupções nas execuções dos coeficientes diferentes de zero e codificando coeficientes diferentes de zero em cada lado da dita interrupção separadamente. Contextos adicionais são fornecidos através da comutação de contexto baseada nas transformações internas, intermediárias e externas assim como nas comutações de contexto baseadas em que transformações correspondem a canais de luminância ou de crominância. Esse fato permite que as tabelas de código apresentem menor entropia, evitando criar tantos contextos, de forma a diluir seu proveito.

O codec exemplificativo também reduz a tabela de código indicando em cada símbolo se um coeficiente diferente de zero apresenta um valor absoluto superior a 1 e se as execuções de zero apresentam um valor positivo, e codifica separadamente o nível de coeficientes e o comprimento das execuções fora dos símbolos. O codec pode se beneficiar da comutação do contexto no que diz respeito a esses níveis e execuções com codificados separadamente.

Os vários sistemas e técnicas podem ser

utilizados em associação ou de forma independente.

Esse Sumário é fornecido a fim de introduzir uma escolha de conceitos de forma simplificada que serão ainda descritos abaixo na Descrição Detalhada. Esse Sumário não
5 pretende identificar elementos essenciais ou elementos chave da matéria objeto reivindicada, nem pretende ser utilizada para auxiliar na determinação do escopo da matéria objeto reivindicada.

Características e vantagens adicionais tornar-se-
10 ão aparentes a partir da seguinte descrição detalhada das modalidades que procedem com referência aos desenhos anexos.

BREVE DESCRIÇÃO DOS DESENHOS

A Figura 1 é um diagrama em bloco de um codec
15 convencional baseado na transformação de bloco na técnica anterior.

A Figura 2 é um histograma mostrando uma distribuição de coeficientes de transformação dotados de ampla variação dinâmica.

20 A Figura 3 é um histograma apresentando uma distribuição de coeficientes com estreita variação.

A Figura 4 é um diagrama de fluxo de um codificador representativo que incorpora uma codificação adaptativa de coeficientes de ampla variação.

25 A Figura 5 é um diagrama de fluxo de um decodificador representativo que incorpora a decodificação de coeficientes com ampla variação codificados adaptativamente.

A Figura 6 é um diagrama de fluxo ilustrando o

agrupamento e a disposição em camadas de coeficiente de transformação na codificação adaptativa de coeficientes de ampla variação, assim como no codificador da Figura 4.

5 A Figura 7 é um fluxograma mostrando um processo realizado pelo codificador da Figura 4 com o intuito de codificar um coeficiente de transformação para um dado agrupamento de coeficientes de transformação em bins.

10 A Figura 8 é um fluxograma apresentando um processo realizado pelo codificador da Figura 5 com o intuito de reconstruir o coeficiente de transformação codificado através do processo da Figura 7.

15 A Figura 9 é um fluxograma apresentando um processo de adaptação com o intuito de variar o agrupamento da Figura 6 visando produzir uma distribuição melhor para a codificação de entropia dos coeficientes.

As Figuras 10 e 11 são uma lista de pseudocódigos do processo de adaptação da Figura 9.

A Figura 12 ilustra exemplos de coeficientes de transformação codificados na técnica anterior.

20 A Figura 13 ilustra um exemplo de coeficientes de transformação codificados de acordo com as técnicas de codificação descritas no presente.

25 A Figura 14 é um fluxograma apresentando um processo realizado pelo decodificador da Figura 4 para codificar coeficientes de transformação.

A Figura 15 ilustra exemplos de diferentes contextos de tabelas de código empregados para codificar coeficientes de transformação de acordo com as técnicas descritas no presente.

A Figura 16 é um fluxograma apresentando um processo realizado pelo codificador da Figura 4 a fim de determinar contextos de codificação a serem empregados ao codificar coeficientes de transformação.

5 A Figura 17 ilustra um exemplo de coeficientes de transformação reduzidos que foram codificados de acordo com as técnicas descritas no presente.

A Figura 18 é um fluxograma apresentando um processo realizado pelo codificador da Figura 4 a fim de
10 codificar e enviar coeficientes iniciais de transformação na forma reduzida.

A Figura 19 é um fluxograma apresentando um processo realizado pelo codificador da Figura 4 a fim de codificar e enviar coeficientes subseqüentes na forma
15 reduzida.

A Figura 20 é um fluxograma apresentando um processo realizado pelo codificador da Figura 5 a fim de codificar coeficientes de transformação codificados.

A Figura 21 é um fluxograma apresentando um
20 processo realizado pelo codificador da Figura 5 a fim de ocupar os coeficientes de transformação a partir dos símbolos decodificados.

A Figura 22 é um diagrama de bloco de um ambiente de computação adequado para a implantação da codificação adaptativa dos coeficientes de ampla variação da Figura 6.
25

DESCRIÇÃO DETALHADA

A seguinte descrição refere-se às técnicas de codificação e de decodificação que se ajustam de modo adaptativo à codificação de entropia mais eficiente dos

coeficientes de transformação de ampla variação, assim como à codificação de entropia mais eficiente dos coeficientes de transformação em geral. A descrição a seguir descreve uma implantação exemplificativa no contexto de um sistema de compressão de mídia digital ou codec. O sistema de mídia digital codifica dados de mídia digital na forma comprimida para transmissão ou armazenamento, e decodifica os dados para reprodução ou outra forma de processamento. Para fins ilustrativos, esse sistema exemplificativo de compressão que incorpora essa codificação adaptativa dos coeficientes de ampla variação é um sistema de compressão de imagem ou de vídeo. Como alternativa, a técnica também pode ser incorporada a sistemas de compressão ou codecs para outros dados de 2D. A técnica de codificação adaptativa de coeficientes de ampla variação não requer que o sistema de compressão de mídia digital codifique os dados de mídia digital comprimidos em um formato particular de codificação.

1. CODIFICADOR/DECODIFICADOR

As Figuras 4 e 5 são um diagrama generalizado dos processos empregados em um decodificador 500 e um codificador 400 representativos de dados bidimensionais (2D). Os diagramas apresentam uma ilustração generalizada ou simplificada de um sistema de compressão que incorpora um decodificador e um codificador de dados 2D que implantam a codificação adaptativa de coeficientes de ampla variação. Em sistemas de compressão alternativos que utilizam a codificação adaptativa de coeficientes de ampla variação, processos adicionais ou apenas alguns daqueles ilustrados

nesse codificador e decodificador representativos podem ser empregados para a compressão de dados 2D. Por exemplo, alguns codificadores/decodificadores também incluem a conversão de cor, formatos de cor, codificação graduável, codificação sem perda de dados, modos de macroblocos, etc. O sistema de compressão (codificador e decodificador) pode proporcionar compressão com perda de dados 2D e/ou sem perda de dados 2D, dependendo da quantização, podendo essa ser baseada em um parâmetro de quantização que varia de dados perdidos a não perdidos.

O codificador de dados 2D 400 produz um fluxo de bits comprimido 420 que é uma representação mais compacta (para a entrada típica) de dados 2D 410 apresentados como a entrada para o codificador. Por exemplo, a entrada de dados 2D pode ser uma imagem, um quadro de uma sequência de vídeo ou outros dados de duas dimensões. O codificador de dados 2D 430 recorta os dados de entrada em macroblocos de 16 x 16 pixels de tamanho nesse codificador representativo. O codificador de dados 2D ainda recorta cada macrobloco em blocos de 4 x 4. Uma operadora de "sobreposição antecipada" 440 é aplicada a cada bordo entre blocos, e em seguida cada bloco 4 x 4 é transformado utilizando uma transformação de bloco 450. Essa transformação de bloco 450 pode ser a transformação 2D reversível e livre de gradação descrita por Srinivasan, Pedido de Patente N° U.S. 11/015.707 intitulada "Reversible Transform for Lossy and Lossless 2-D Data Compression", depositado em 17 de dezembro de 2004. A operadora de sobreposição 440 pode ser a operadora de sobreposição reversível descrita por Tu et al, Pedido de

Patente N° U.S. 11/015.148 intitulada "Reversible Overlap Operator for Efficient Lossless Data Compression" depositado em 17 de dezembro de 2004; e por Tu et al, Pedido de Patente N° U.S. 11/035.991 intitulado "Reversible 2-Dimensional Pré-/Post-Filtering for Lapped Biorthogonal Transform" depositado em 4 de janeiro de 2005. Como alternativa, pode-se utilizar a transformação discreta do cosseno. Em seguida à transformação, o coeficiente DC 460 de cada bloco de transformação 4x4 é submetido à cadeia de processamento similar (recorte, sobreposição antecipada, seguida da transformação do bloco 4x4). Os coeficientes de transformação DC resultantes e os coeficientes de transformação AC são quantificados 470, codificados por entropia 480 e empacotados 490.

O decodificador executa o processo reverso. No lado do decodificador, os bits do coeficiente de transformação são extraídos 510 de seus respectivos pacotes, a partir dos quais os coeficientes em si são decodificados 520 e dequantificados 530. Os coeficientes DC 540 são regenerados aplicando-se a transformação inversa, e o plano dos coeficientes DC é "sobreposto de forma inversa" aplicando uma operadora de uniformização apropriada perpendicularmente às bordas dos blocos DC. Em seguida, todos os dados são regenerados aplicando-se a transformação inversa 4x4 550 aos coeficientes DC e os coeficientes AC 542 são decodificados a partir do fluxo de bits. Finalmente, as bordas dos blocos nos planos das imagens resultantes são filtradas com sobreposição inversa 560, o que produz uma saída de dados de 2D reconstruídos.

Em uma implantação exemplificativa, o codificador 400 (Figura 4) comprime uma imagem de entrada em um fluxo de bits comprimidos 420 (por exemplo, um arquivo) e o decodificador 500 (Figura 5) reconstrói a entrada original ou uma aproximação da mesma, utilizando como base se a codificação empregada apresenta ou não perda de dados. O processo de codificação envolve a aplicação de uma transformação sobreposta anterior (LT) discutida abaixo, que é implantada junto com uma pré-/pós-filtragem reversível bidimensional, também descrita abaixo em sua totalidade. O processo de decodificação envolve a aplicação da transformação sobreposta inversa (ILT) que utiliza a pré-/pós-filtragem reversível bidimensional.

A LT e a ILT ilustradas são a inversão uma da outra, no sentido exato e, portanto, podem ser coletivamente denominadas como uma transformação sobreposta reversível. Como uma transformação reversível, o par LT/ILT pode ser utilizado para uma compressão de imagem sem perda de dados.

Os dados de entrada 410 comprimidos pelo codificador 400/decodificador 500 ilustrados podem ser imagens com formatos de cores variadas (por exemplo, formatos de imagens coloridas RGB/YUV4: 4:4, YUV4: 2:2 ou YUV4: 2:0). Tipicamente, a imagem de entrada sempre apresenta um componente de luminância (Y). Caso seja uma imagem RGB/YUV4: 4:4, YUV4: 2:2 ou YUV4: 2:0, a imagem também apresenta componentes de crominância, como um componente U e um componente V. Os planos coloridos separados da imagem podem possuir diferentes resoluções

espaciais. No caso de uma imagem de entrada no formato colorido YUV 4:2: 0, por exemplo, os componentes U e V apresentam metade da largura e da altura do componente Y.

Conforme discutido acima,, o codificador 400
 5 agrupa a imagem de entrada ou a figura em macroblocos. Em uma implantação exemplificativa, o codificador 400 agrupa a imagem de entrada em macroblocos de 16 x 16 no canal Y (que podem apresentar áreas de 16 x 16, 16 x 8 ou 8 x 8 nos canais U e V, dependendo do formato colorido). Cada plano
 10 colorido do macrobloco é agrupado em blocos ou regiões 4 x 4. Portanto, um macrobloco é composto por diversos formatos coloridos da seguinte maneira para a presente implantação de codificador exemplificativo:

1. Para uma imagem em tons de cinza, cada
 15 macrobloco contém 16 blocos 4 x 4 de luminância (Y).

2. Para uma imagem colorida com formato YUV4: 2:0, cada macrobloco contém 16 blocos 4 x 4 Y e 4 blocos 4 x 4 com crominância (U e V) cada.

3. Para uma imagem colorida com formato YUV4: 2:2, cada macrobloco contém 16 blocos 4 x 4 Y e 4 blocos 4 x 4 com crominância (U e V) cada.

4. Para uma imagem colorida RGB ou YUV4: 4:4, cada macrobloco contém 16 blocos cada um com canais Y, U e V.

25 2. CODIFICAÇÃO ADAPTATIVA DE COEFICIENTES DE AMPLA VARIAÇÃO

No caso de dados de ampla variação dinâmica, em especial dados de transformação não correlacionados (tais como os coeficientes 460, 462 no codificador da Figura 4),

um número significativo de bits de razão menor são imprevisíveis e "ruidosos". Em outras palavras, não há correlação estreita dos bits de razão menor que podem ser utilizados para a codificação de entropia eficiente. Os
 5 bits apresentam entropia elevada, aproximando-se de 1 bit para cada bit codificado.

2.1 AGRUPAMENTO

Mais além, a função de distribuição de probabilidade Laplaciana dos coeficientes de transformação
 10 de ampla variação demonstrados na Figura 3 é dada por:

$$\frac{\lambda}{2} e^{-\lambda|x|}$$

(por conveniência, a variável aleatória correspondente aos coeficientes de transformação é tratada como um valor contínuo). Para dados de ampla variação
 15 dinâmica, γ é pequeno, e a média absoluta $1/\gamma$ é grande. O gradiente dessa distribuição está limitado a $\pm \frac{1}{2} (\gamma^2)$, o que é bem pequeno. Significa que a probabilidade de um coeficiente de transformação ser igual a x está muito próxima à probabilidade de $x + \xi$ para um ξ de pequeno. No
 20 domínio discreto, isso traduz a reivindicação, "a probabilidade de um coeficiente de transformação adotar valores adjacentes j e $(j+1)$ é quase idêntica".

Fazendo referência à Figura 6, a codificação adaptativa de coeficientes de ampla variação desempenha um
 25 agrupamento 610 de símbolos sucessivos do alfabeto em "bins" de símbolos N . O número de símbolos por bin pode ser qualquer número N . Para comodidade, no entanto, o número N é, de preferência, uma potência de 2 (i.e., $N = 2^k$), de

forma que o indicador ou endereço de um coeficiente dentro de um bin pode ser codificado com eficiência como um código de comprimento fixo. Por exemplo, os símbolos podem ser agrupados em pares, de modo que um símbolo possa ser
5 identificado como o indicador do par, junto com o indicador do símbolo dentro do par.

Esse agrupamento se beneficia do fato de que, com uma escolha apropriada de N , a distribuição de probabilidade do indicador do bin para coeficientes de
10 ampla variação se parece mais com uma distribuição de probabilidade de dados de estreita variação, por exemplo, aqueles demonstrados na Figura 3. O agrupamento é matematicamente semelhante à operação de quantização, o que significa que o indicador de bin pode ser codificado com
15 eficiência utilizando técnicas de codificação de entropia com comprimento variável que atuam melhor para dados com distribuição de probabilidade de variação estreita.

Baseando-se no agrupamento de coeficientes em bins, o codificador pode então codificar um coeficiente de
20 transformação 615 empregando um indicador de seu bin (também citado no presente como o coeficiente normalizado 620) e seu endereço dentro do bin (citado no presente como o endereço de bin 625). O coeficiente normalizado é codificado utilizando uma codificação de entropia com
25 comprimento variável, enquanto o endereço do bin é codificado por meio de um código de comprimento fixo.

A escolha de N (ou de forma equivalente, o número de bits k para a codificação de comprimento fixo do endereço bin) determina a granularidade do agrupamento. Em

geral, quanto maior a variação dos coeficientes de transformação, maiores os valores k que devem ser escolhidos. Quando k é escolhido com critério, o coeficiente normalizado Y é zero, com elevada probabilidade de se ajustar ao modelo de codificação de entropia para Y .

Conforme descrito acima, pode-se atribuir adaptativamente diferentes valores a k (de forma adaptativa com a anterior) em um codificador e decodificador. Mais especificamente, o valor de K tanto no codificador quanto no decodificador varia com base apenas nos dados previamente codificados/decodificados.

Em um exemplo particular desta codificação apresentada na Figura 7, o codificador codifica um coeficiente de transformação X da forma adiante. Para uma ação inicial 710, o codificador calcula um coeficiente normalizado Y para o coeficiente de transformação. Nessa implantação exemplificativa, o coeficiente normalizado Y é definido como

$Y = \text{sinal}(X) * \text{piso}(\text{abs}(X)/N)$, para uma determinada escolha do tamanho do bin $N = 2^k$. O codificador codifica o símbolo Y usando um código de entropia (ação 720), seja isoladamente ou em conjunto com outros símbolos. A seguir, na ação 730, o codificador determina um endereço bin (Z) do coeficiente de transformação X . Nessa implantação exemplificativa, o endereço bin é o resíduo da divisão inteira de $\text{abs}(X)$ pelo tamanho de bin N ou $Z = \text{abs}(X) \% N$. O codificador codifica esse valor como um código de comprimento fixo de k bits na ação 740. Mais além, no caso de um coeficiente de transformação diferente de zero, o

codificador também codifica o sinal. Mais especificamente, conforme indicado nas ações 750-760, o codificador codifica o sinal do coeficiente normalizado (Y) quando o coeficiente normalizado é diferente de zero. Ainda mais, no caso do coeficiente normalizado ser zero e o coeficiente de transformação ser diferente de zero, o codificador codifica o sinal do coeficiente de transformação (X). Uma vez que o coeficiente seja codificado através de um código de entropia com comprimento variável, também será citado no presente como a parte de comprimento variável, e o endereço bin (Z) também será citado no presente como a parte de comprimento fixo. Em outras implantações alternativas, as definições matemáticas do coeficiente normalizado, do endereço bin e do sinal para um coeficiente de transformação podem variar.

Continuando com esse exemplo, a Figura 8 apresenta um processo exemplificativo 800 realizado pelo codificador 500 (Figura 5) a fim de reconstruir o coeficiente de transformação que foi codificado pelo processo 700 (Figura 7). Na ação 810, o codificador codifica o coeficiente normalizado (Y) a partir do fluxo de bits comprimido 420 (Figura 5), tanto isoladamente quanto em conjunto com outros símbolos segundo definido no processo de codificação do bloco. O decodificador ainda lê a palavra código com k bits para o endereço bin e o sinal (quando codificado) a partir do fluxo de bits comprimido na ação 820. Nas ações 830 a 872, o codificador então reconstrói o coeficiente de transformação, da seguinte maneira:

1. Quando $Y > 0$ (ação 830), então o coeficiente de transformação é reconstruído como $X = Y * N + Z$ (ação 831)).

2. Quando $Y < 0$ (ação 840), então o coeficiente de transformação é reconstruído como $X = Y * N - Z$ (ação 841).

5 3. Quando $Y = 0$ e $Z = 0$ (ação 850), então o coeficiente de transformação é reconstruído como $X = 0$ (ação 851).

4. Quando $Y = 0$ e $Z \neq 0$, o codificador ainda lê o sinal codificado (S) a partir do fluxo de bits comprimido (ação 860). Se o sinal for positivo ($S = 0$) (ação 870), então
10 o coeficiente de transformação é reconstruído como $X = Z$ (ação 871). Ainda, se o sinal for negativo ($S = 1$), o coeficiente de transformação é reconstruído como $X = Z$ (ação 872).

15 2.2 DISPOSIÇÃO EM CAMADAS

Fazendo referência novamente à Figura 6, o codificador e o decodificador, de preferência, extraem os endereços bin de comprimento fixo codificados 625 e o sinal para uma camada codificada separada (no presente denominada
20 camada "Flexbits" 645) no fluxo de bits comprimido 420 (Figura 4). Os coeficientes normalizados 620 são codificados em uma camada do fluxo de bits do núcleo 640, o que permite ao codificador e/ou decodificador optar por
25 minimizar ou abandonar inteiramente essa porção Flexbits da codificação, conforme desejado, para se ajustar à taxa de bit ou a outras restrições. Mesmo que o codificador abandone inteiramente a camada Flexbit, o fluxo de bits comprimido ainda poderia efetuar a decodificação, embora com degradação da qualidade. O decodificador poderia ainda

reconstruir o sinal a partir apenas da porção dos coeficientes normalizados. Essa situação é efetivamente semelhante à aplicação de um grau mais elevado de quantização 470 (Figura 4) no decodificador. A

5 decodificação do sinal e dos endereços bin como uma camada de flexbits separada também apresenta o benefício potencial que em algumas implantações de codificador/decodificador, uma outra codificação de entropia com comprimento variável (por exemplo, codificação aritmética, Lempel-Ziv, Burrows-

10 Wheeler, etc.) poderia ser aplicada aos dados nessa camada para futuro aperfeiçoamento de compressão.

Para a disposição de camadas, as seções do fluxo de bits contendo as porções flexbits são sinalizadas através de um cabeçote de camada separada ou outra

15 indicação no fluxo de bits, de modo que o decodificador possa identificar e separar (i.e., analisar) a camada de Flexbits 645 (quando não omitida) a partir do fluxo de bits principal 640.

A disposição de camadas apresenta um desafio

20 adicional no projeto do agrupamento adaptativo anterior (descrito na próxima seção). Uma vez que a camada Flexbits pode estar presente ou ausente em um dado fluxo de bits, o modelo de agrupamento adaptativo anterior não pode, com confiança, se referir a nenhuma informação na camada

25 Flexbits. Toda a informação necessária para se determinar o número de bits de código de comprimento fixo k (correspondente ao tamanho bin $N=2^k$) deveria residir no fluxo de bits causal, principal.

2.3 ADAPTAÇÃO

O codificador e decodificador proporcionam ainda um processo adaptativo anterior para ajustar adaptativamente a escolha do número k de bits de código de comprimento fixo e, de modo correspondente, o tamanho de bin N do agrupamento descrito acima, durante a codificação e a decodificação. Em uma implantação, o processo de adaptação pode se basear na modelagem dos coeficientes de transformação como uma distribuição Laplaciana, de forma que o valor de k deriva do parâmetro Laplaciano λ . No entanto, o dito modelo sofisticado poderia requerer que o decodificador executasse o inverso do agrupamento (reconstruir os coeficientes de transformação a partir dos coeficientes normalizados no fluxo de bits principal e o endereço/sinal bin na camada Flexbits) na Figura 6 antes de modelar a distribuição para blocos futuros. Esse requerimento poderia violar a restrição da disposição de camadas que o decodificar deveria permitir abandonando a camada Flexbits a partir do fluxo de bits comprimido.

Na implantação exemplificativa apresentada na Figura 9, o processo de adaptação se baseia, ao contrário, na observação de que uma melhor codificação de comprimento de execução dos coeficientes de transformação é atingida quando em torno de um quarto dos coeficientes são diferentes de zero. Assim, um parâmetro de adaptação que pode ser empregado para ajustar o agrupamento em direção a uma situação "ideal" irá promover um bom desempenho da codificação de entropia. Da mesma forma, o número de coeficientes normalizados diferentes de zero em um bloco é utilizado como um parâmetro de adaptação na implantação

exemplificativa. Esse parâmetro de adaptação incorpora a vantagem de depender apenas na informação contida no fluxo de bits principal, que se ajusta à restrição da disposição de camada que os coeficientes de transformação podem ainda ser decodificados com a omissão da camada Flexbits. O processo é uma adaptação anterior no sentido de que o modelo de adaptação aplicado quando codificando/decodificando o atual bloco se baseia na informação do(s) bloco(s) anterior(es).

No seu processo de adaptação, o codificador e o decodificador exemplificativos realizam a adaptação com base na adaptação anterior. Quer dizer, uma iteração atual da adaptação se baseia na informação previamente observada no processo de codificação ou de decodificação, tal como no bloco ou macrobloco anterior. No decodificador e no codificador exemplificativo, a atualização da adaptação ocorre uma vez por bloco para uma dada banda de transformação, a qual deve atuar reduzindo a latência e a dependência cruzada. Implantações alternativas de codec podem realizar a adaptação em intervalos diferentes, como após cada bloco de transformação.

No decodificador e no codificador exemplificativos, o processo de adaptação atualiza o valor k . Se o número do coeficiente normalizado diferente de zero for muito alto, então k assumirá valores maiores, de forma que esse número tenderá a ser abandonado nos blocos futuros. Se o número de coeficientes normalizados diferentes de zero for muito pequeno, então o valor k é reduzido com a expectativa de que blocos futuros, então,

irão produzir mais coeficientes normalizados diferentes de zero, porque o tamanho de bin N é menor. O processo de adaptação exemplificativo restringe o valor k ao conjunto de números $\{0, 1, \dots, 16\}$, mas implantações alternativas 5 poderiam utilizar outras variações de valores para k . A cada atualização de adaptação, o codificador e o decodificador elevam, reduzem ou mantêm o valor k inalterado. O codificador e o decodificador elevam ou reduzem k em um, mas implantações alternativas poderiam 10 empregar outros tamanhos de etapas.

O processo de adaptação 900 no codificador e decodificador exemplificativo utiliza ainda um parâmetro de modelo interno ou uma variável de estado (M) para controlar a atualização do parâmetro de agrupamento k com efeito de 15 histerese. Esse parâmetro de modelo produz um intervalo antes de atualizar o parâmetro de agrupamento k , evitando causar flutuação rápida no parâmetro de agrupamento. O parâmetro do modelo no processo de adaptação exemplificativo é composto por 17 etapas inteiras, de -8 a 20 8.

Fazendo referência agora à Figura 9, o processo de adaptação exemplificativa 900 ocorre da seguinte forma. Esse processo de adaptação exemplificativa é mais detalhado na relação de pseudocódigos das Figuras 10 e 11. Nas 25 indicações das ações 910, 990, o processo de adaptação no codificador e decodificador exemplificativo é realizado separadamente em cada banda de transformação que está representada no fluxo de bits comprimido, inclusive a banda de luminância e as bandas de crominância, os coeficientes

AC e DC, etc. Os codecs alternativos podem apresentar variação no número de bandas de transformação, e depois podem aplicar a adaptação às bandas de transformação isoladamente ou em conjunto.

5 Na ação 920, o processo de adaptação, então, contabiliza o número de coeficientes normalizados diferentes de zero da banda de transformação dentro do próximo macrobloco previamente codificado/decodificado. Na
10 ação 930, essa contabilização bruta é normalizada com o objetivo de refletir o número inteiro de coeficientes diferentes de zero em uma área de tamanho regular. O processo de adaptação, então, calcula (ação 940) o desvio da contagem a partir do modelo desejado (i.e., o "ponto ideal" de um quarto dos coeficientes que é diferente de
15 zero). Por exemplo, um macrobloco de coeficientes AC no codificador exemplificativo apresentado na Figura 4 apresenta 240 coeficientes. Então, o modelo desejado deve apresentar 70 coeficientes diferentes de zero dos 240 coeficientes. Em seguida, o desvio é dimensionado, nivelado
20 e empregado para atualizar o parâmetro do modelo interno.

 Nas ações seguintes 960, 965, 970, 975, então, o processo de adaptação adapta o valor k de acordo com qualquer alteração no parâmetro do modelo interno. Se o parâmetro do modelo interno é inferior a um patamar
25 negativo, o valor k é reduzido (dentro de seus limites admissíveis). Essa adaptação deveria produzir mais coeficientes diferentes de zero. Por outro lado, se o parâmetro do modelo exceder um patamar positivo, o valor k é elevado (dentro dos limites admissíveis). A dita

adaptação deve produzir poucos coeficientes diferentes de zero. Caso contrário, o valor k permanece inalterado.

Mais uma vez, conforme indicado nas ações 910, 980, o processo de adaptação é repetido separadamente para cada canal e sub-banda dos dados, tais como separadamente para os canais de luminância e crominância.

O processo de adaptação exemplificativo 900 é observado com mais detalhes na relação de pseudocódigo 1000 apresentada nas Figuras 10 e 11.

10 3. CODIFICAÇÃO DE ENTROPIA EFICIENTE

3.1 MÉTODOS DA TÉCNICA ANTERIOR

Em diversos padrões de codificação, o processo de codificação dos blocos de transformação se reduz à codificação de uma sequência de coeficientes. Um exemplo da dita sequência é dada na Figura 12 como exemplo de coeficientes de transformação 1200. No exemplo 1200, os coeficientes C_0 , C_1 , C_2 , C_3 e C_4 representam quatro valores de coeficientes diferentes de zero (tanto de sinal positivo quanto de sinal negativo) enquanto os outros coeficientes na série apresentam valor igual a zero.

Determinadas propriedades são tradicionalmente características da dita sequência de coeficientes de transformação:

- O número total de coeficientes é tipicamente determinista e é dado pelo tamanho da transformação.
- Segundo a probabilidade, um grande número de coeficientes é zero.
- Pelo menos um coeficiente é diferente de zero. No caso de todos os coeficientes serem iguais a zero,

o caso é tipicamente sinalizado através de padrão de bloco codificado, assim como descrito em Srinivasan, Pedido N° U.S. TDB, "Non-Zero Coefficient Block Pattern Coding", depositado em 12 de agosto de 2005.

5 • Segundo a probabilidade, coeficientes diferentes de zero e com valores maiores ocorrem no início da seqüência, e coeficientes de valor igual a zero e menores ocorrem no sentido do final da seqüência.

 • Coeficientes diferentes de zero assumem
10 valores inteiros mínimo/ máximo conhecidos.

Inúmeras técnicas de codificação se beneficiam do fato de que os coeficientes de valor igual a zero, que tipicamente ocorrem com menor frequência, podem ser codificados com códigos de comprimento de execução.

15 Entretanto, quando a imagem de entrada que está sendo codificada apresenta dados de elevada variação dinâmica (por exemplo, maior do que 8 bits) ou quando o parâmetro de quantização é igual a uma unidade ou é pequeno, poucos coeficientes de transformação são zero, conforme discutido

20 abaixo. Na dita situação, a codificação adaptativa e as técnicas de decodificação descritas acima podem ser utilizadas para condicionar os dados de modo que os dados condicionados apresentem essas características. Outras técnicas também podem produzir conjuntos de coeficientes de

25 transformação semelhantes àqueles da exemplificação de coeficientes de transformação 1200 através de outros meios, como, por exemplo, ao ajustar um nível elevado de quantização.

A Figura 12 também ilustra dois métodos de

codificar coeficientes de transformação como os da exemplificação dos coeficientes de transformação 1200. Esses métodos se beneficiam da codificação conjunta de uma seqüência de zeros junto com o coeficiente diferente de zero seguinte visando proporcionar um benefício à codificação. A exemplificação de codificação 2D 1220 demonstra uma técnica para o dito modelo de codificação de nível de seqüência. Conforme ilustrado pela exemplificação 1220, na codificação 2D uma seqüência de coeficientes diferentes de zero (sendo a seqüência de comprimento zero ou positivo) é codificada junto como um símbolo 1225 com o coeficiente diferente de zero seguinte na série de coeficientes de transformação; no caso ilustrado o símbolo $\langle 0, C0 \rangle$ indica que não há zeros antecedendo o coeficiente diferente de zero $C0$. Um símbolo especial 1235 denominada "final do bloco", ou BOB, é empregado para sinalizar a última série de zeros. Essa codificação é tipicamente denominada codificação 2D porque cada símbolo codifica em conjunto a seqüência (a seqüência de coeficientes com valor igual a zero) e o nível (o valor do coeficiente diferente de zero) e, portanto, apresenta dois valores, e pode ser interpretado como uma codificação de duas dimensões dos dados de coeficiente de transformação. Esses símbolos, então, podem ser codificados por entropia, utilizando os códigos Huffman ou a codificação aritmética e são enviados ao fluxo de bits comprimidos 420 da Figura 4.

Outro modelo de codificação alternativo é a codificação 3D, cuja exemplificação é ilustrada na exemplificação 1240. Na codificação 3D, a seqüência de

zeros é tipicamente codificada em conjunto com o coeficiente seguinte diferente de zero, da mesma forma que na codificação 2D. Ainda, um elemento de dados booleano, "último", que indica se esse coeficiente diferente de zero é o último coeficiente diferente de zero no bloco, é codificado. O símbolo 1245, portanto, codifica conjuntamente a sequência, o nível e o último; no caso ilustrado o símbolo <2, C1, não último> indica que dois zeros antecedem o coeficiente diferente de zero C1, e que não é o último coeficiente diferente de zero na série. Uma vez que todos esses elementos podem adotar livremente todos os valores, o símbolo codifica três dimensões independentes, originando o nome "codificação 3D".

Cada uma dessas técnicas apresenta vantagens isoladas. Cada símbolo na técnica de codificação 2D apresenta entropia menor do que o símbolo usado na codificação 3D, porque a primeira transmite menos informações do que a última. Assim, o número de possíveis símbolos em um dado modelo de codificação 3D será duas vezes maior do que para um modelo comparável de codificação 2D, o que eleva o tamanho da tabela de código, e pode desacelerar a codificação e a decodificação do modelo de codificação 3D. Entretanto, na codificação 2D um símbolo adicional é enviado para sinalizar o fim do bloco, e é dispendioso requerer o envio de um símbolo adicional inteiro, da perspectiva do tamanho do fluxo de bits. De fato, na prática, a codificação 3D é mais eficiente do que a codificação 2D, apesar dos tamanhos maiores das tabelas de código.

3.2 CODIFICAÇÃO 3½ D - 2½ D

Enquanto as técnicas anteriores ilustradas na Figura 12 utilizam a codificação conjunta de níveis de coeficiente diferentes de zero junto com as seqüências anteriores de zeros, pode ser demonstrado que a seqüência de zeros posteriores aos coeficientes diferentes de zero apresenta forte correlação com a magnitude do coeficiente diferente de zero. Essa propriedade sugere a utilidade do nível de codificação conjunta e da seqüência seguinte.

A Figura 13 demonstra uma dita técnica alternativa de codificação que aprimora as técnicas de 2D e de 3D esquematizadas na Figura 12. A Figura 13 ilustra uma exemplificação 1340 de um modelo de codificação que utiliza a idéia de codificar seqüências sucessivas de zero com o intuito de criar símbolos destinados a uma série exemplificativa de coeficientes de transformação 1300. A Figura 13 ilustra que os coeficientes são codificados em conjunto em símbolos 1355, que contém o valor de um coeficiente diferente de zero junto com o comprimento da seqüência de zeros que sucedem o coeficiente diferente de zero (se houver algum) como um par; <nível, seqüência>. Assim, o símbolo ilustrado 1355, <C1, 4> codifica em conjunto o coeficiente diferente de zero C1 e os quatro coeficientes de valor igual a zero que o sucedem.

Além de se beneficiar da forte correlação entre os coeficientes diferentes de zero e as seqüências de zeros sucessivos, esse método proporciona uma vantagem adicional quando um coeficiente diferente de zero é o último coeficiente diferente de zero no bloco, utilizando um valor

especial de seqüência para sinalizar que o coeficiente diferente de zero é o último na série. Assim, na codificação conjunta de um símbolo, a informação que é enviada é um valor de nível e um outro valor que indica tanto o comprimento de uma seqüência de zeros quanto o "último" valor. A Figura 13 ilustra essa situação através do símbolo 1365, <C4, último>, que compreende um valor de nível e um "último" valor mais do que o comprimento de uma seqüência. Pelo fato dessas diferentes situações serem codificadas no mesmo lugar de um símbolo, a seqüência e o "último" não são independentes; apenas um é enviado por símbolo. Assim, a dimensionalidade do símbolo não é 2 nem 3, mas algo entre esses dois valores. Essa codificação é denominada "codificação 2 ½".

Esse elemento da codificação 2 ½ D não é necessariamente requerido no modelo de codificação conjunta que associa níveis e seqüências sucessivas; em uma implantação alternativa, o símbolo final transmitido poderia simplesmente codificar o comprimento da seqüência final de zeros, embora isso pudesse ser indesejável porque poderia aumentar sobremaneira o tamanho do fluxo de bits codificado. Em outra alternativa, um símbolo EOB, semelhante ao utilizado na codificação 2D, poderia ser utilizado. No entanto, assim como na codificação 3D, a utilização de um "último" valor na codificação 2½ D admite uma vantagem sobre a codificação 2D, não há necessidade de se codificar um símbolo extra para denotar o fim do bloco. Adicionalmente, a codificação 2½ D, admite vantagens sobre a codificação 3D (1) a entropia de cada símbolo da primeira

é inferior à da última e (2) o modelo da tabela de código da primeira é mais simples do que a última. As duas vantagens decorrem do fato da codificação $2\frac{1}{2}$ D apresentar menores possibilidades do que a codificação 3D.

5 No entanto, apenas a codificação $2\frac{1}{2}$ D não pode descrever uma seqüência completa de coeficientes de transformação porque não fornece uma maneira de enviar um comprimento de seqüência antes do primeiro coeficiente diferente de zero. Conforma ilustra a Figura 13, para essa
10 finalidade, um símbolo especial 1375 é utilizado, que codifica adicionalmente o comprimento da primeira seqüência de zeros. Isso torna o primeiro símbolo uma codificação conjunta da `first_run`, nível e (seqüência OU último). Na Figura 13, o primeiro símbolo 1375, $\langle 0, C0, 2 \rangle$ envia a
15 primeira seqüência (que é zero), o nível do primeiro coeficiente diferente de zero e a segunda seqüência (que é 2, e o primeiro coeficiente diferente de zero não é o último coeficiente diferente de zero no bloco). Pelo fato desse símbolo compreender uma dimensão adicional, a
20 codificação é mencionada como "codificação $3\frac{1}{2}$ D".

Embora a informação extra na codificação $3\frac{1}{2}$ D possa parecer, ao primeiro olhar, negar algumas das vantagens da codificação $2\frac{1}{2}$ D, o manuseio distinto do primeiro símbolo é realmente vantajoso segundo a
25 perspectiva da eficiência da codificação. Um símbolo $3\frac{1}{2}$ D necessariamente apresenta um alfabeto diferente do outro, símbolos $2\frac{1}{2}$ D, o que significa que ele é codificado separadamente de outros símbolos e não eleva a entropia de $2\frac{1}{2}$ D.

A Figura 14 mostra um processo exemplificativo 1400 realizado pelo codificador 400 (Figura 4) para codificar coeficientes de transformação de acordo com a codificação $2\frac{1}{2}$ D - $3\frac{1}{2}$ D. Em um meio, o processo 1400 pode
5 ser incluído como parte do processo 720 da Figura 7 para codificar coeficientes normalizados. Em outro, o processo 1400 pode ser utilizado para codificar coeficientes de transformação que foram quantificados por meio de técnicas tradicionais. Em numerosas implantações do processo 1400,
10 as ações podem ser removidas, combinadas ou fragmentadas em sub-ações.

O processo principia pela ação 1420, onde o primeiro coeficiente de transformação diferente de zero é identificado. Em seguida, na ação 1430, um símbolo $3\frac{1}{2}$ D é
15 criado a partir do comprimento inicial da seqüência de zeros (que tanto poderia apresentar comprimento 0 quanto comprimento positivo) e o primeiro coeficiente diferente de zero. Nesse ponto, o símbolo $3\frac{1}{2}$ D não está completo. A seguir, o processo atinge a ação decisiva 1435, onde
20 determina se o coeficiente diferente de zero que é atualmente identificado é o último coeficiente diferente de zero na série de coeficientes de transformação. Se esse for o último coeficiente diferente de zero, o processo continua até a ação 1480, onde o "último" indicador é inserido no
25 símbolo, em vez de ser inserido na seqüência de zeros sucessivos. Esse processo, então, codifica o símbolo usando a codificação de entropia na ação 1490, e o processo é finalizado. Um exemplo do dito processo de codificação de símbolos é dado abaixo, com referência à Figura 16.

Se, entretanto, o processo determinar na ação decisiva 1435 que esse não é o último coeficiente diferente de zero, então, na ação 1440 o comprimento da sequência sucessiva de zeros (que pode ser 0 ou um número positivo) é inserido no símbolo e o símbolo é codificado na ação 1450. Um exemplo do dito processo de codificação de símbolos é dado abaixo com referência à Figura 16. Em seguida, o processo identifica o próximo coeficiente diferente de zero na ação 1460, cuja existência é conhecida porque o coeficiente diferente de zero anterior foi determinado como não sendo o último. Na ação 1470, um símbolo $2\frac{1}{2}$ D é então criado utilizando esse coeficiente diferente de zero. Nesse momento, como o símbolo $3\frac{1}{2}$ D acima, o símbolo ainda não está completo. Então, na ação decisiva 1435, o processo determina se o atual coeficiente diferente de zero é o último da série. Se assim for, o processo continua até a ação 1480, onde o "último" indicador é incluído e o símbolo é codificado. Caso contrário, o processo retorna à ação 1440, onde a próxima sequência de zeros é incluída, o símbolo é codificado e o processo continua com o próximo coeficiente diferente de zero.

3.3 INFORMAÇÕES DO CONTEXTO

Além de codificar símbolos de acordo com as codificações $2\frac{1}{2}$ D e $3\frac{1}{2}$ D, inúmeras partes da informação causal podem ser utilizadas para criar um contexto para o símbolo que é codificado. Esse contexto pode ser utilizado pelo codificador 400 (Figura 4) ou pelo decodificador 500 (Figura 5) para indexar em uma de uma coleção de tabelas de codificação de entropia para codificar e decodificar o

símbolo. Aumentar o número de contextos proporciona maior flexibilidade ao codec para adaptar, ou para usar tabelas talhadas para cada contexto específico. No entanto, as desvantagens de se definir um grande número de contextos são que (1) há diluição do contexto (em que cada contexto se aplica apenas a um pequeno número de símbolos, reduzindo assim a eficiência da adaptação) e (2) maior número de tabelas de código significa maior complexidade e requerimentos de memória.

Com esses pontos em mente, o modelo do contexto descrito no presente é escolhido para consultar três fatores com o intuito de determinar que contexto é escolhido para cada símbolo. Em uma implantação, esses três fatores são (1) o nível de transformação - se a transformação é interna, intermediária ou externa, (2) se os coeficientes são dos canais de luminância e crominância e (3) se houve alguma interrupção na sequência de zeros dentro da série de coeficientes. Nas implantações alternativas, um ou mais desses fatores podem ser usados para determinar o contexto da codificação, e/ou outros fatores podem ser considerados.

Assim, por meio de (1), uma transformação interna emprega um conjunto diferente de tabelas de código do que uma transformação intermediária, que emprega um conjunto diferente de tabelas de código do que uma transformação externa. Em outras implantações, os modelos do contexto podem apenas diferenciar entre dois níveis de transformação. De forma análoga, por meio de (2) os coeficientes de luminância empregam um conjunto diferente

de tabelas de código do que os coeficientes de crominância. Ambos os fatores desse contexto não se modificam dentro de um dado conjunto de coeficientes de transformação.

No entanto, o fator (3) varia dentro de um conjunto de coeficientes de transformação. A Figura 15 ilustra três séries exemplificativas de coeficientes de transformação que melhor ilustram essa mudança de contexto. Em todas as três séries 1500, 1520 e 1540, coeficientes diferentes de zero são representados por letras.

Como os três exemplos ilustram, o primeiro símbolo em um bloco, sendo um símbolo $3\frac{1}{2}$ D é necessariamente codificado com uma tabela diferente do que os outros símbolos, porque seu alfabeto é diferente dos outros. Esse forma um contexto "natural" para o primeiro símbolo. Assim, o coeficiente A, sendo o primeiro coeficiente diferente de zero de todos os três exemplos, é codificado com um código $3\frac{1}{2}$ D. Adicionalmente, pelo fato do símbolo $3\frac{1}{2}$ D codificar as seqüências de zero anteriores e seguintes em torno do primeiro coeficiente diferente de zero, os dois primeiros coeficientes do exemplo 1520 (A, 0) e os dois primeiros coeficientes do exemplo 1540 (0, A) são codificados em conjunto em um símbolo $3\frac{1}{2}$ D. Em função disso, em uma implantação, o fator (3) não pode ser aplicado na determinação do contexto dos símbolos $3\frac{1}{2}$ D.

Em contrapartida, os símbolos $2\frac{1}{2}$ D são codificados diferentemente dependendo do fator (3). Assim, no exemplo 1500, pode ser observado que pelo fato de não haver interrupção na seqüência de coeficientes diferentes de zero até depois do coeficiente D, os coeficientes B, C e

D (assim como os zeros seguintes ao D) são codificados com o primeiro modelo do contexto. No entanto, os zeros após o D constituem uma interrupção na sequência de coeficientes diferentes de zero. Portanto, os coeficientes restantes E, F, G, H (e todos os que se seguirem)... são codificados usando o segundo modelo do contexto, o que significa que, enquanto cada coeficiente diferente de zero e de A é codificado com um símbolo $2\frac{1}{2}$ D, diferentes tabelas de código serão usadas para os coeficientes B, C e D (e quaisquer sequências associadas com valor zero) do que serão usadas para os coeficientes E, F, G e H.

Por outro lado, no exemplo 1520, há uma interrupção entre A e B, o que constitui uma interrupção na sequência de coeficientes diferentes de zero e, portanto, o coeficiente B e todos os coeficientes subseqüentes são codificados com o segundo modelo do contexto. De forma análoga, no exemplo 1540, há uma interrupção antes de A. Assim, como no exemplo 1520, os coeficientes B, C, D... são codificados com o segundo modelo do contexto.

A Figura 16 mostra um processo exemplificativo 1600 realizado pelo codificador 400 (Figura 4) para codificar um símbolo. Em uma implantação, o processo 1600 realiza o processo das ações 1450 e 1490 do processo 1400 (Figura 14). Em diversas implantações do processo 1600, as ações podem ser removidas, combinadas ou fragmentadas em sub-ações. O processo se inicia na ação decisiva 1605, onde o codificador determina se o símbolo é um símbolo $3\frac{1}{2}$ D. Se assim for, o processo prossegue até a ação 1610, onde o símbolo é codificado utilizando a tabela $3\frac{1}{2}$ D e o processo

é finalizado. Em várias implantações o símbolo pode ser codificado através da codificação de entropia, tal como a codificação de Huffman ou a codificação aritmética. Como alternativa outros modelos de codificação podem ser utilizados.

Se o símbolo não for um símbolo de $3\frac{1}{2}$ D, o processo prossegue até a ação decisiva 1615, onde o codificador determina se pelo menos um zero antecedeu o coeficiente diferente de zero que é codificado conjuntamente no símbolo. Caso contrário, o processo persiste até a ação 1620, onde o símbolo é codificado por meio das tabelas de código $2\frac{1}{2}$ D a partir do primeiro modelo do contexto e o processo é finalizado. Se tiver havido uma interrupção, então na ação 1630 o símbolo é codificado usando as tabelas de código $2\frac{1}{2}$ D a partir do segundo modelo do contexto e o processo é finalizado.

3.4 REDUÇÃO DO TAMANHO DA TABELA DE CÓDIGO

Enquanto as técnicas descritas acima são mais eficientes em relação às técnicas tradicionais, elas ainda não são capazes, por si próprias, de reduzir significativamente o tamanho da tabela de código. As tabelas de código criadas para as técnicas deveriam ser capazes de transmitir todas as combinações ($\text{max_level} \times (\text{max_run} + 2)$) para os símbolos $2\frac{1}{2}$ D e ($\text{max_level} \times (\text{max_run} + 1) \times (\text{max_run} + 2)$) para os símbolos $3\frac{1}{2}$ D, onde o max_level é o valor máximo (absoluto) de um coeficiente diferente de zero e max_run é o comprimento máximo possível de uma sequência de zeros. O valor ($\text{max_run} + 1$) é derivado para a sequência inicial do símbolo de $3\frac{1}{2}$ D porque os

valores possíveis para uma seqüência de seqüência de zeros de 0 para série_máx, para um total de (max_run + 1). De modo idêntico, cada símbolo codifica uma seqüência sucessiva de zeros de comprimento entre 0 e max_run, assim como um último símbolo, para um total de valores (max_run + 2). Mesmo com codificação de escape (onde símbolos que ocorrem raramente são agrupados em um ou múltiplos meta-símbolos sinalizados através de códigos de escape), os tamanhos das mesas de código podem ser formidáveis.

10 A fim de reduzir ainda mais o tamanho das tabelas de código, as técnicas descritas acima podem ser ainda mais refinadas. Primeiro cada seqüência e cada nível é fragmentado em um par de símbolos:

15 seqüência = nonZero_run (+ série1)
 nível = nonOne_level (+ nível1)

Nesse par de símbolos, os símbolos nonZero_run e nonOne_level são booleanos, indicando respectivamente se a seqüência é maior que zero, e o valor absoluto é maior que 1. Os valores sequencial e nível1 são usados apenas quando os booleanos são verdadeiros, e indicam a série (entre 1 e max_run) e o nível (entre 2 e o max_level). No entanto, pelo fato do caso do "último" também precisar ser codificado, o valor (seqüência OU último) de qualquer seqüência sucessiva de zeros em um símbolo codificado em conjunto é enviado como um símbolo ternário nonZero_run_last, que assume o valor 0 quando a seqüência apresenta comprimento igual a zero, 1 quando a seqüência apresenta comprimento diferente de zero e 2 quando o coeficiente diferente de zero do símbolo for o último da

série.

Portanto, para utilizar essa codificação reduzida, o primeiro símbolo $3\frac{1}{2}$ D assume a forma `<nonZero_run, nonOne_level, nonZero_run_last>`. Isso cria um alfabeto de tamanho $2 \times 2 \times 3 = 12$. Os símbolos subseqüentes $2\frac{1}{2}$ D assumem as formas `<nonOne_level, nonZero_run_last>` criando um alfabeto de tamanho $2 \times 3 = 6$. Em uma implantação, a série1 também é denominada NonzeroRun e o nível 1 é denominado SignificantLevel.

10 Pelo fato do Índice conter apenas informações sobre se os níveis e seqüências são significantes, pode ser necessário enviar informações adicionais com os símbolos a fim de permitir que um decodificador recrie com precisão uma série de coeficientes de transformação. Assim, após
15 cada símbolo do índice, se o nível for um nível significante, o valor do nível é separadamente codificado e enviado após o símbolo. Da mesma forma, se um símbolo indicar que uma seqüência de zeros apresenta um comprimento diferente de zero (positivo), aquele comprimento é
20 codificado separadamente e enviado após o símbolo.

A Figura 17 ilustra um exemplo de uma codificação reduzida $3\frac{1}{2}$ D- $2\frac{1}{2}$ D 1740 que representa uma série exemplificativa 1700 de valores absolutos de coeficientes de transformação. Os sinais dos coeficientes de
25 transformação podem ser codificados em outro lugar. Conforme ilustra a figura 17, a série exemplificativa dos coeficientes 1700 começa com "5, 0, 0". Na $3\frac{1}{2}$ D- $2\frac{1}{2}$ D não reduzida, assim como naqueles ilustrados acima, o primeiro símbolo seria então `<0, 5, 2>`. No entanto, na codificação

reduzida, a Figura 17 ilustra um primeiro símbolo 1745 do Índice $\langle 0, 1, 1 \rangle$. Esse símbolo indica que não há zeros antes do primeiro coeficiente diferente de zero, que o primeiro coeficiente diferente de zero apresenta valor absoluto maior do que 1, e que há pelo menos um zero depois desse coeficiente diferente de zero. A esse símbolo segue-se o valor SignificantLevel "level_5" (1755), indicando que o valor absoluto do coeficiente diferente de zero é 5, e um valor NonzeroRun "run_2" (1765), que indicam a existência de dois zeros depois do coeficiente. Por outro lado, o símbolo 1775, $\langle 0, 0 \rangle$, que indica a existência de um coeficiente diferente de zero de valor absoluto 1 seguido por valores diferentes de zero, não requer que outros valores o sucedam para fornecer informações.

Em função de alguns símbolos requererem o envio de informações adicionais após os mesmos, os símbolos do Índice deveriam ser analisados para determinar se informações adicionais deveriam ser enviadas junto deles. A Figura 18 apresenta um processo exemplificativo 1800 realizado pelo codificador 400 (Figura 4) com o intuito de determinar que informação está contida em um símbolo do Índice $3\frac{1}{2}$ D e de enviar informações adicionais, caso apropriado. Em diversas implantações do processo 1800, ações podem ser removidas, combinadas ou fragmentadas em sub-ações. Na descrição dos símbolos para a Figura 18, o valor "x" é um marcador de posição, representando qualquer valor possível para aquela porção particular de um símbolo. O processo começa na ação 1810, onde o primeiro símbolo codificado é enviado. A seguir, na ação decisiva 1820, o

codificador determina se o símbolo é da forma $\langle x, 1, x \rangle$, o que significa perguntar se o coeficiente diferente de zero representado pelo símbolo adota um valor absoluto superior a 1. Se o codificador estipular que esse é o caso, o valor
 5 do coeficiente diferente de zero é codificado e enviado na ação 1830. É importante ressaltar que, enquanto a Figura 18 não discute explicitamente a codificação do sinal de um coeficiente diferente de zero, esse sinal poderia ser incluído em diversos pontos no processo 1800. Em numerosas
 10 implantações, isso poderia envolver enviar o sinal imediatamente a seguir do símbolo de codificação conjunta, dentro do símbolo de codificação conjunta, e/ou junto do valor absoluto do nível.

Independente da determinação na ação 1820, na
 15 ação decisiva 1840, o codificador determina se o símbolo adota a forma $\langle 1, x, x \rangle$. Essa determinação equivale a perguntar se o coeficiente diferente de zero representado pelo símbolo apresenta qualquer zero que o preceda. Se assim for, na ação 1850, o codificador codifica o
 20 comprimento da sequência de zeros que precede o coeficiente diferente de zero e envia esse valor.

A seguir, na ação decisiva 1860, o codificador considera que o valor de t onde o símbolo for $\langle x, x, t \rangle$. Essa determinação equivale a perguntar se o coeficiente
 25 diferente de zero representado pelo símbolo apresenta quaisquer zeros que o sucedam. Se $t=0$, então o codificador sabe que não há zeros que o sucedam, e continua a enviar mais símbolos na ação 1880 e o processo 1800 é finalizado. Em uma implantação, o processo 1900 da Figura 19 começa,

então, pelo próximo símbolo. Se $t=1$, o codificador então codifica e envia o comprimento da sequência de zeros que sucedem o coeficiente diferente de zero na ação 1870, e então continua a enviar símbolos na ação 1880 e o processo 1800 é finalizado. No entanto, se $t=2$, o codificador sabe que o coeficiente diferente de zero representado pelo símbolo é o último (e único) na série, e assim o bloco representado pelos coeficientes de transformação está completo. Assim, o processo 1800 é finalizado e o próximo bloco pode ser transformado e codificado, se aplicável.

A Figura 19 apresenta um processo exemplificativo 1900 realizado pelo codificador 400 (Figura 4) para determinar que informação está contida em um símbolo de Índice $2\frac{1}{2}$ D e para enviar informações adicionais, se apropriado. Em diversas implantações do processo 1900, ações podem ser removidas, combinadas ou fragmentadas em sub-ações. Assim como na Figura 18, na Figura 19, o valor "x" é um marcador, representando qualquer valor possível para aquela porção particular do símbolo. O processo inicia na ação 1910, onde o próximo símbolo codificado é enviado. A seguir, na ação decisiva 1920, o codificador determina se o símbolo é da forma $\langle 1, x \rangle$, o que equivale a perguntar se o coeficiente diferente de zero representado pelo símbolo adota um valor absoluto superior a 1. Se o codificador determinar que esse é o caso, o valor do coeficiente diferente de zero é codificado e enviado na ação 1930. Assim como no processo 1800, é importante ressaltar que enquanto a Figura 19 não discute explicitamente a codificação do sinal de um coeficiente diferente de zero,

esse sinal poderia ser incluído em diversos pontos do processo 1900.

A seguir, na ação decisiva 1940, o codificador considera o valor de t onde o símbolo for $\langle x, t \rangle$, o que equivale a perguntar se o coeficiente diferente de zero representado pelo símbolo apresenta zeros que o sucedam. Se $t=0$, então o codificador reconhece que não há zeros o sucedendo, e continua a enviar mais símbolos na ação 1960 e o processo 1900 é finalizado. Em uma implantação, o processo 1900 da Figura 19 então repete para o próximo símbolo. Se $t=1$, o codificador então codifica e envia o comprimento da sequência de zeros que sucedem o coeficiente diferente de zero na ação 1950, e então continua a enviar símbolos na ação 1960 e o processo 1900 é finalizado. No entanto, se $t=2$, o codificador reconhece que o coeficiente diferente de zero representado pelo símbolo é o último da série e, portanto, o bloco representado pelos coeficientes de transformação está completo. Assim, o processo 1900 é finalizado e o próximo bloco pode ser transformado e codificado, se aplicável.

3.5 EFICIÊNCIAS ADICIONAIS

Além do tamanho da redução da tabela de código discutida acima, um benefício proporcionado pela interrupção dos símbolos da sequência e do nível é que, em seguida à transmissão do símbolo conjunto $3\frac{1}{2}$ D, o decodificador pode determinar a existência ou não de zeros orientadores. Isso significa que a informação do contexto descrevendo se o primeiro ou o segundo modelo do contexto se mantém é reconhecida no lado do decodificador e

constitui um contexto válido para codificar o valor do nível do primeiro coeficiente diferente de zero. Isso quer dizer que os contextos que aplicam ao nível valores dos símbolos de $2\frac{1}{2} D$ podem aplicar igualmente ao nível valores de símbolos $3\frac{1}{2} D$, mesmo enquanto os símbolos do Índice codificados conjuntamente utilizam alfabetos diferentes.

Mais ainda, uma vez que o número total de coeficientes de transformação em um bloco é um número constante, cada sequência sucessiva é delimitada por uma sequência monotônica decrescente. Em uma implantação preferida, essa informação é explorada na codificação de valores da sequência. Por exemplo, uma tabela de código pode incluir um conjunto de códigos de valor da sequência para sequências que começam na primeira metade de um conjunto de coeficientes e um conjunto diferente para sequências que começam na segunda metade. Pelo fato do comprimento de qualquer sequência possível que se inicia na segunda metade ser necessariamente menor do que os comprimentos possíveis de sequências que iniciam na primeira metade, o segundo conjunto de códigos não devem ser grandes, reduzindo a entropia e aperfeiçoando o desempenho da codificação.

Outras informações podem ser coletadas por meio da observação cautelosa do posicionamento dos coeficientes. Por exemplo, se o coeficiente diferente de zero representado por um símbolo ocorrer na última posição na série de coeficientes, então "último" sempre é verdadeiro. De forma semelhante, se o coeficiente diferente de zero representado por um símbolo ocorrer na penúltima posição do

arranjo, então, tanto o "último" é verdadeiro quanto a sequência sucessiva é zero. Cada uma dessas observações permite a codificação através de tabelas mais curtas.

3.6 EXEMPLIFICAÇÃO DA IMPLANTAÇÃO DO ÍNDICE

5 O primeiro Índice possui um alfabeto de tamanho 12. Em uma implantação, cinco tabelas de Huffman estão disponíveis para esse símbolo, que é definido como $\text{FirstIndex} = a + 2b + 4c$, onde o símbolo é $\langle a, b, c \rangle$, sendo que a e b assumem valores 0 ou 1 e c assume valores 0, 1
10 ou 2. Uma implantação dos comprimentos da palavra código para os doze símbolos relativos a cada uma das tabelas é dado abaixo. Os procedimentos para a construção do código padrão de Huffman podem, em uma implantação, ser aplicados de forma a originar esses conjuntos de palavras código pré-
15 determinadas.

Tabela 1: 5,6,7,7,5,3,5,1,5,4,5,3

Tabela 2: 4,5,6,6,4,3,5,2,3,3,5,3

Tabela 3: 2,3,7,7,5,3,7,3,3,3,7,4

Tabela 4: 3,2,7,5,5,3,7,3,5,3,6,3

20 Tabela 5: 3,1,7,4,7,3,8,4,7,4,8,5

Os símbolos subsequentes de Índice possuem um alfabeto de tamanho 6. Em uma implantação, o Índice é definido como $\text{Índice} = a + 2b$, onde o símbolo é $\langle a, b \rangle$ e a é booleano, e b pode assumir valores de 0, 1 ou 2. Quatro
25 tabelas de Huffman são definidas para o Índice, conforme demonstrado abaixo:

Tabela 1: 1,5,3,5,2,4

Tabela 2: 2,4,2,4,2,3

Tabela 3: 4,4,2,2,2,3

Tabela 4: 5,5,2,1,4,3

Além disso, em uma implantação, com o objetivo de se beneficiar de algumas das informações descritas na Seção 3.5 acima, quando o coeficiente está posicionado na última disposição do arranjo, um código de um bit (definido por a) é usado (b é unicamente 2 nesse caso). Em uma implantação, quando o coeficiente ocupa a penúltima posição, um código de 2 bits é utilizado, visto ser conhecido que $b \neq 1$.

Uma implantação dos códigos SignificantLevel codifica o nível utilizando um procedimento denominado "binning", que fragmenta uma gradação de níveis em sete bins. Os níveis dentro de um bin são codificados a partir de códigos de comprimento fixo, e os bins em si são codificados utilizando os códigos de Huffman. Isso pode ser efetuado, em uma implantação, através das técnicas de agrupamento descritas acima. Da mesma forma, em uma implantação, NonzeroRun é codificado empregando um procedimento de "binning" que indexa em cinco bins com base na localização do símbolo atual.

3.7 SÍMBOLOS DE DECODIFICAÇÃO $3\frac{1}{2}$ D - $2\frac{1}{2}$ D OU DECODIFICANDO SÍMBOLOS $3\frac{1}{2}$ D - $2\frac{1}{2}$ D?

A Figura 20 apresenta um processo exemplificativo 2000 efetuado pelo decodificador 500 (Figura 5) visando decodificar uma série de símbolos em coeficientes de transformação. Em várias implantações do processo 2000, as ações podem ser removidas, combinadas ou fragmentadas em sub-ações. Mais ainda, as ações podem ser definidas com o objetivo de manipular as condições de erro semelhantes àquelas acionadas pelo fluxo de bits corrompido. O processo

se inicia na ação 2010, onde o decodificador recebe o primeiro símbolo codificado em conjunto e o decodifica utilizando a tabela de código $3\frac{1}{2}$ D. A seguir, na ação 2020, os coeficientes de transformação são ocupados com base no símbolo decodificado (inclusive qualquer informação de sequência ou nível também presente no fluxo de bits comprimido). Uma implantação dessa ação é descrita em maior detalhe abaixo em relação à Figura 21. O processo continua até a ação decisiva 2030, onde o decodificador determina se o símbolo indica se esse se refere ao último coeficiente diferente de zero. Se assim for, o processo continua até a ação 2090, onde quaisquer coeficientes residuais não ocupados são ocupados com valores iguais a zero e o processo 2000 é finalizado.

Se o símbolo não se refere ao último coeficiente diferente de zero, o processo continua até a ação decisiva 2040, onde o decodificador determina se algum coeficiente diferente de zero foi indicado por qualquer símbolo até o momento. Em caso negativo a ação prossegue até a ação 2050, onde o próximo símbolo é recebido e decodificado empregando as tabelas de código $2\frac{1}{2}$ D em seguida ao primeiro modelo do contexto. Se, em vez disso, os coeficientes zero tiverem sido indicados na ação decisiva 2040, no processo 2060, o decodificador recebe e decodifica o próximo símbolo empregando tabelas de código $2\frac{1}{2}$ D em seguida ao segundo modelo do contexto. Independente de qual modelo do contexto foi utilizado, o processo então continua até a ação 2070, onde os coeficientes de transformação são preenchidos com base no símbolo decodificado (inclusive qualquer informação

de seqüência ou nível também presente no fluxo de bits comprimido). De forma idêntica à ação 2020, uma implantação dessa ação é descrita em maior detalhe abaixo em relação à Figura 21. O processo continua até a ação decisiva 2080, onde o decodificador determina se o símbolo indica que esse se refere ao último coeficiente diferente de zero. Caso contrário, o processo retorna à ação decisiva 2040 e se repete. Se assim for, o processo continua até a ação 2090, onde quaisquer coeficientes não ocupados são ocupados com valores iguais a zero, e o processo 2000 é finalizado.

A Figura 21 apresenta um processo exemplificativo 2100 efetuado pelo decodificador 500 (Figura 5) para ocupar coeficientes de transformação. Em numerosas implantações do processo 2100, as ações podem ser removidas, combinadas ou fragmentadas em sub-ações. Na medida em que o processo 2100 é configurado para decodificar símbolos codificados de acordo com as técnicas da Seção 3.4 acima, nas implantações alternativas, os valores de nível e os comprimentos da seqüência podem ser incluídos em símbolos $2\frac{1}{2}$ D e $3\frac{1}{2}$ D, que poderiam permitir que o processo 2100 fosse simplificado. O processo se inicia na ação decisiva 2010, onde o decodificador determina se o símbolo é um símbolo $3\frac{1}{2}$ D. Em caso negativo, o processo salta para a ação decisiva 2140, descrita abaixo. No entanto, se o símbolo for um símbolo $3\frac{1}{2}$ D, o decodificador determina na ação decisiva 2120 se o símbolo indica uma seqüência inicial de comprimento positivo de coeficientes de valor igual a zero. Isso pode ser feito determinando se o valor de nonZero_run no símbolo $3\frac{1}{2}$ D é 1, indicando uma seqüência de comprimento positivo,

ou 0, indicando uma sequência de comprimento igual a zero. Se o símbolo indicar uma sequência de comprimento positivo de coeficientes de valor igual a zero, o processo continua até a ação 2130, onde o comprimento da sequência é
5 decodificado, com base no nível codificado em seguida ao símbolo $3\frac{1}{2}$ D, e os coeficientes iniciais de transformação são ocupados com zero de acordo com o comprimento da sequência.

A seguir, o processo continua até a ação decisiva
10 2140, onde o decodificador determina se o símbolo indica que seu coeficiente diferente de zero apresenta valores absolutos maiores do que 1. Isso pode ser feito determinando se o valor de nonOne_level no símbolo é igual a 1, indicando que o nível apresenta valor absoluto maior
15 do que 1, ou 0, indicando que o coeficiente diferente de zero é -1 ou 1. Se o símbolo não indicar um coeficiente de valor absoluto maior do que 1, o processo continua até a ação 2150, onde o próximo coeficiente é ocupado com -1 ou 1, dependendo do sinal do coeficiente diferente de zero. Se
20 o símbolo indicar um coeficiente com valor absoluto maior do que 1, o processo, ao contrário, continua até a ação 2160, onde o nível do coeficiente é decodificado e o coeficiente é ocupado com o valor do nível, junto de seu sinal. Conforme discutido acima, o sinal pode ser indicado
25 de maneiras variadas, de forma que a decodificação do sinal não é discutida explicitamente nas ações 2150 ou 2160.

Continuando, na ação decisiva 2170, o decodificador determina se o símbolo indica uma sequência subsequente de comprimento positivo de coeficientes iguais

a zero, o que pode ser feito determinando se o valor `nonZero_run_last` no símbolo é igual a 1, indicando uma sequência de comprimento positivo, ou 0, indicando uma sequência de comprimento zero. (O caso onde `nonZero_run_last` igual a 2 não é apresentado, já que esse caso é entendido no processo 2000). Se o símbolo indicar uma sequência de comprimento positivo de coeficientes iguais a zero, o processo continua até a ação 2180, onde o comprimento da sequência é decodificado, com base na sequencial codificada em seguida ao símbolo, e os coeficientes de transformação subsequentes são preenchidos com zeros de acordo com o comprimento da sequência, finalizando assim o processo 2100.

4. AMBIENTE COMPUTACIONAL

O decodificador 400 (Figura 4) descrito acima, o decodificador 500 (Figura 5) e as técnicas para codificar e decodificar com eficiência os coeficientes de transformação podem ser executados em uma gama de dispositivos em que o processamento do sinal da mídia digital é executado, incluindo, entre outros exemplos, computadores; gravação de vídeo e imagem, equipamentos de transmissão e de recepção; aparelhos de reprodução de vídeo portáteis; vídeo-conferências; e etc. As técnicas de codificação de mídia digital podem ser implantadas em um conjunto de circuitos de *hardware*, assim como em *software* de processamento de mídia digital, operando em computador ou em outro ambiente computacional semelhantes aos apresentados na Figura 22.

A Figura 22 ilustra um exemplo generalizado de um ambiente computacional adequado (2200) em que as

modalidades descritas podem ser implantadas (2200). O ambiente computacional (2200) não pretende sugerir limitações ao escopo do uso ou da funcionalidade da invenção, já que a presente invenção pode ser implantada em
5 diversos ambientes computacionais com finalidades gerais ou específicas.

Com referência à Figura 22, o ambiente computacional (2200) inclui pelo menos uma unidade de processamento (2210) e a memória (2220). Na Figura 22, a
10 configuração mais básica (2230) é incluída dentro de uma linha pontilhada. A unidade de processamento (2210) cumpre as instruções executáveis pelo computador, podendo ser um processador real ou virtual. Em um sistema de multiprocessamento, as unidades de multiprocessamento
15 cumprem as instruções executáveis pelo computador visando elevar a potência de processamento. A memória (2220) pode ser uma memória volátil (por exemplo, registros, cache, RAM), uma memória não-volátil (por exemplo, ROM, EEPROM, memória flash, etc.) ou uma combinação das duas. A memória
20 (2220) armazena o *software* (1280) que implanta o codificador/decodificador descrito, assim como as técnicas eficientes de codificação/decodificação do coeficiente de transformação.

Um ambiente computacional pode apresentar
25 elementos adicionais. Por exemplo, o ambiente computacional (2200) inclui o armazenamento (2240), um ou mais dispositivos de entrada (2250), um ou mais dispositivos de saída (2260) e uma ou mais conexões de comunicação (2270). Um mecanismo de interconexão (não apresentado), como um

barramento, uma controladora ou uma rede realiza a interconexão dos componentes do ambiente computacional (2200). Tipicamente, um *software* de sistema em operação (não apresentado) proporciona um ambiente de operação para
5 outros *softwares* que operam no ambiente computacional (2200), e coordena as atividades dos componentes do ambiente computacional (2200).

O armazenamento pode ser removível ou não removível e inclui discos magnéticos, cassetes ou fitas
10 magnéticas, CD-ROMS, CD-RWs, DVDs ou outras mídias que possam ser usadas para armazenar informações e que possam ser acessadas no ambiente computacional (2200). O armazenamento (2240) armazena instruções para o *software* que implementa o codificador/decodificador descrito e as
15 técnicas eficientes de codificação/decodificação do coeficiente de transformação.

O(s) dispositivo(s) de entrada (2250) podem ser um dispositivo de entrada mediante o toque, como um teclado, um *mouse*, uma caneta ou *trackball* (*mouse* de
20 esfera), um dispositivo de entrada de voz, um dispositivo de varredura ou algum outro dispositivo que forneça entradas ao ambiente computacional (2200). Para áudio, o dispositivo de entrada (2250) pode ser uma placa de som ou um dispositivo semelhante que aceite entrada de áudio em
25 formato analógico ou digital, ou uma leitora de CD-ROM que forneça amostras de áudio para o ambiente computacional. O(s) dispositivo(s) de saída (2260) pode ser um vídeo, uma impressora, um alto-falante, um gravador de CD ou outro dispositivo que permita a saída a partir do ambiente

computacional (2200).

A(s) conexões de comunicação (2270) permitem a comunicação sobre uma mídia de comunicação pra outra entidade computacional. A mídia de comunicação transfere
5 informações como as instruções executáveis pelo computador, informações de áudio ou vídeo comprimidas, ou outros dados em um sinal de dados modulados. Um sinal de dados modulado é um sinal que apresenta uma ou mais de suas características ajustadas ou modificadas de modo a
10 codificar informações no sinal. Para fins de exemplificação, e não visando restringir, mídias de comunicação incluem técnicas com fio ou sem fio implantadas com uma portadora elétrica, óptica, RF, infravermelha, acústica ou qualquer outra.

15 As técnicas de processamento de mídia digital do presente podem ser descritas no contexto geral das mídias legíveis por computador. A mídia legível por computador é qualquer mídia disponível que pode ser acessada dentro de um ambiente computacional. Para fins de exemplificação, e
20 não de limitação, com o ambiente computacional (2200), a mídia legível por computador inclui memória (2220), armazenamento (2240), mídia de comunicação e as combinações de qualquer uma das acima.

25 As técnicas de processamento de mídia digital no presente podem ser descritas no contexto geral das instruções executáveis por computador, semelhantes àquelas incluídas nos módulos do programa, que são executadas em um ambiente computacional no processador alvo real ou virtual. Geralmente, os módulos do programa incluem rotinas,

programas, bibliotecas, objetos, classes, componentes, estrutura dos dados, etc. que desempenhar tarefas específicas ou implantam tipos específicos de dados abstratos. A funcionalidade dos módulos do programa pode ser combinada ou desmembrada em módulos do programa conforme desejado em diversas modalidades. As instruções executáveis por computador para os módulos do programa podem ser executadas em um ambiente computacional local ou distribuído.

Para fins de apresentação, a descrição detalhe emprega termos como "determinar", "gerar", "ajustar" e "aplicar" para descrever operações de computador em um ambiente computacional. Esses termos são abstrações de nível elevado para as operações executadas por computador e não devem ser confundidas com atos desempenhados por um ser humano. As operações reais do computador correspondentes a esses termos variam dependendo da implantação.

Em vista das muitas variações possíveis em torno da matéria objeto descrita no presente, reivindicamos como sendo nossas as ditas modalidades que possam surgir dentro do escopo das seguintes reivindicações e equivalentes às mesmas.

REIVINDICAÇÕES

1. Método de codificar uma série de coeficientes de transformação (1300) representando dados de mídias digitais, **CARACTERIZADO** pelo fato de que compreende:

5 representar (1400) a série de coeficientes de transformação como uma série de símbolos (1340), em que cada símbolo (1355) compreende uma indicação de um coeficiente diferente de zero obtido a partir de uma série de coeficientes de transformação e uma indicação de um
10 comprimento de uma seqüência subsequente de coeficientes de valor igual a zero;

 para cada símbolo na série de símbolos, enviar (1600) o símbolo a ser codificado em um fluxo de bits comprimido (420); e

15 em que um último coeficiente diferente de zero em uma seqüência de coeficientes de transformação é representado por um símbolo (1365), onde a indicação de um comprimento de uma seqüência subsequente de coeficientes de
valor igual a zero compreende um indicador de que o
20 coeficiente é o último coeficiente diferente de zero.

2. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o primeiro símbolo na série de símbolos compreende adicionalmente uma indicação de um comprimento de uma seqüência de coeficientes de valor igual
25 a zero que precede um primeiro coeficiente diferente de zero na série de coeficientes de transformação.

3. Método, de acordo com a reivindicação 2, **CARACTERIZADO** pelo fato de que uma série de coeficientes de transformação resulta da determinação de uma porção

normalizada de uma série de coeficientes de transformação de ampla variação.

4. Método, de acordo com a reivindicação 2, **CARACTERIZADO** pelo fato de que um conjunto de tabelas de código é utilizada para codificar o primeiro símbolo na série de símbolos, e um conjunto diferente de tabelas de código é utilizada para codificar outros símbolos na série de símbolos.

5. Método, de acordo com a reivindicação 2, **CARACTERIZADO** pelo fato de que cada símbolo na série de símbolos possui um contexto relacionado.

6. Método, de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que o contexto para um símbolo se baseia, pelo menos em parte, em se um coeficiente diferente de zero representado pelo símbolo ocorre antes ou depois de um primeiro coeficiente de valor igual a zero na série de coeficientes.

7. Método, de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que o contexto para um símbolo se baseia, pelo menos em parte, em se a série de coeficientes de transformação se destinada aos canais de luminância ou de cromaticidade.

8. Método, de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que o contexto para um símbolo se baseia, pelo menos em parte, se a série de coeficientes de transformação se destina a uma transformação interna, intermediária ou externa.

9. Método, de acordo com a reivindicação 2, **CARACTERIZADO** pelo fato de que para cada símbolo, a

indicação naquele símbolo do comprimento da seqüência de coeficientes subseqüentes com valores iguais a zero compreende um símbolo ternário que indica uma das seguintes condições no símbolo:

5 a seqüência de coeficientes de valor igual a zero apresenta comprimento igual a zero;

 a seqüência de coeficientes de valor igual a zero apresenta um comprimento positivo;

 o coeficiente indicado pelo símbolo é o último
10 coeficiente na série de coeficientes de transformação.

 10. Método, de acordo com a reivindicação 9, **CARACTERIZADO** pelo fato de que a indicação do comprimento da seqüência de coeficientes de valor igual a zero que precede o primeiro coeficiente diferente de zero no
15 primeiro símbolo na série de símbolos compreende um booleano que indica se a seqüência de coeficientes de valor igual a zero que precede o primeiro coeficiente diferente de zero apresenta ou não comprimento igual a zero ou se apresenta comprimento positivo.

20 11. Método, de acordo com a reivindicação 9, **CARACTERIZADO** ainda pelo fato de que, quando um símbolo compreende uma indicação de que uma seqüência de coeficientes de valor igual a zero apresenta um comprimento positivo, o comprimento da seqüência de coeficientes de
25 valor igual a zero a ser codificada é enviado através de um fluxo de bits comprimido.

 12. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que, para cada símbolo, a indicação naquele símbolo do coeficiente diferente de zero

a partir da série de coeficientes de transformação compreende um booleano que indica se o valor absoluto do coeficiente diferente de zero é ou não maior do que um.

13. Método, de acordo com a reivindicação 12,
5 **CARACTERIZADO** ainda pelo fato de que, quando um símbolo compreende uma indicação de que um coeficiente diferente de zero apresenta um valor absoluto maior do que 1, o valor absoluto do coeficiente diferente de zero a ser codificado deve ser enviado através de um fluxo de bits comprimido.

10 14. Decodificador de mídias digitais,
CARACTERIZADO pelo fato de que compreende:

uma memória de armazenamento de dados (2280) para armazenar dados de mídias digitais codificados; e

um processador (2210) configurado para:

15 receber um conjunto de símbolos comprimidos (420) descrevendo uma série de coeficientes de transformação;

descomprimir os símbolos (510); e

reconstruir (520) as séries de coeficientes de transformação analisando o conjunto de símbolos
20 descomprimidos; e

onde:

o conjunto de símbolos comprimidos compreende símbolos codificados em conjunto (1340) codificados (1600) a partir de um conjunto de tabelas de código de acordo com
25 um modelo do contexto;

cada símbolo codificado em conjunto (1355) descreve um nível diferente de zero a partir da série de coeficientes de transformação e se há coeficientes de valor igual a zero depois do nível diferente de zero.

15. Decodificador de mídias digitais, de acordo com a reivindicação 14, **CARACTERIZADO** pelo fato de que:

5 cada símbolo codificado em conjunto descreve um nível diferente de zero indicando se um valor absoluto de um nível diferente de zero é maior do que 1:

o conjunto de símbolos comprimidos compreende adicionalmente símbolos de nível que descrevem o valor de cada nível diferente de zero e indicações dos sinais do símbolo; e

10 o processador é ainda configurado para, ao analisar um símbolo codificado em conjunto que indica um nível diferente de zero cujo valor absoluto é maior do que 1, reconstruir o nível diferente de zero em uso encontrando o símbolo do nível que descreve o valor do nível diferente
15 de zero.

16. Decodificador de mídias digitais, de acordo com a reivindicação 14, **CARACTERIZADO** pelo fato de que o primeiro símbolo no conjunto de símbolos comprimidos descreverem adicionalmente se há coeficientes de valor
20 igual a zero que precedem o nível diferente de zero na série de coeficientes de transformação.

17. Decodificador de mídias digitais, de acordo com a reivindicação 14, **CARACTERIZADO** pelo fato de que:

25 cada símbolo codificado em conjunto descreve uma sequência de coeficientes de valor igual a zero indicando se um comprimento da sequência de coeficientes de valor igual a zero é maior do que 0;

o conjunto de símbolos comprimidos compreende ainda símbolos de sequência que descrevem o comprimento de

cada seqüência de coeficientes de valor igual a zero cujo comprimento é maior do que 0.

18. Uma ou mais mídias legíveis por computador, portando instruções que descrevem uma mídia digital decodificando um programa para executar um método de decodificar dados de mídia digital comprimida, sendo que o método é **CARACTERIZADO** pelo fato de que compreende:

receber (2000) um fluxo de bits (420) que compreende símbolos de códigos conjuntos comprimidos (1340) codificados a partir de um conjunto de tabelas de código de acordo com um modelo do contexto;

decodificar (2000) os símbolos de códigos conjuntos para determinar, para cada símbolo, um nível de coeficiente de transformação e uma seqüência de coeficientes de transformação de valor igual a zero subsequentes; e

reconstruir (2100) um conjunto de coeficientes de transformação a partir de níveis determinados e de seqüências de coeficiente de transformação de valor igual a zero.

19. Mídia legível por computador, de acordo com a reivindicação 18, **CARACTERIZADA** pelo fato de que o fluxo de bits ainda compreende um símbolo inicial de códigos conjuntos e o método ainda compreender decodificar o símbolo inicial de códigos conjuntos para determinar uma seqüência de coeficientes de transformação igual a zero, um nível de coeficiente de transformação e uma seqüência de coeficientes de transformação subsequentes de valor igual a zero.

20. Mídia legível por computador, de acordo com a reivindicação 18, **CARACTERIZADA** pelo fato de que:

os símbolos de códigos conjuntos indicam, para cada nível de coeficiente de transformação, se aquele nível apresenta um valor absoluto superior a 1;

os símbolos de códigos conjuntos indicam, para cada seqüência de coeficientes de transformação de valor igual a zero, se aquela seqüência apresenta comprimento superior a 0;

o fluxo de bits compreende ainda símbolos comprimidos que descrevem o comprimento da seqüência de coeficientes de transformação de valor igual a zero, e o sinal e a magnitude dos níveis de coeficientes de transformação; e

decodificar os símbolos de códigos conjuntos compreende:

ao decodificar um símbolo de código conjunto indicando que um nível de coeficiente de transformação apresenta um valor absoluto superior a 1, determinar o nível do coeficiente de transformação através da decodificação de um ou mais símbolos no fluxo de bits que indicam o sinal e o valor do nível do coeficiente de transformação; e

ao decodificar um símbolo de código conjunto indicando que uma seqüência de coeficientes de transformação de valor igual a zero é maior que 0, determinar o comprimento da seqüência de coeficientes de transformação de valor igual a zero através da decodificação de um símbolo em um fluxo de bits que indica o comprimento da sucessão.

FIG. 1
TÉCNICA ANTERIOR

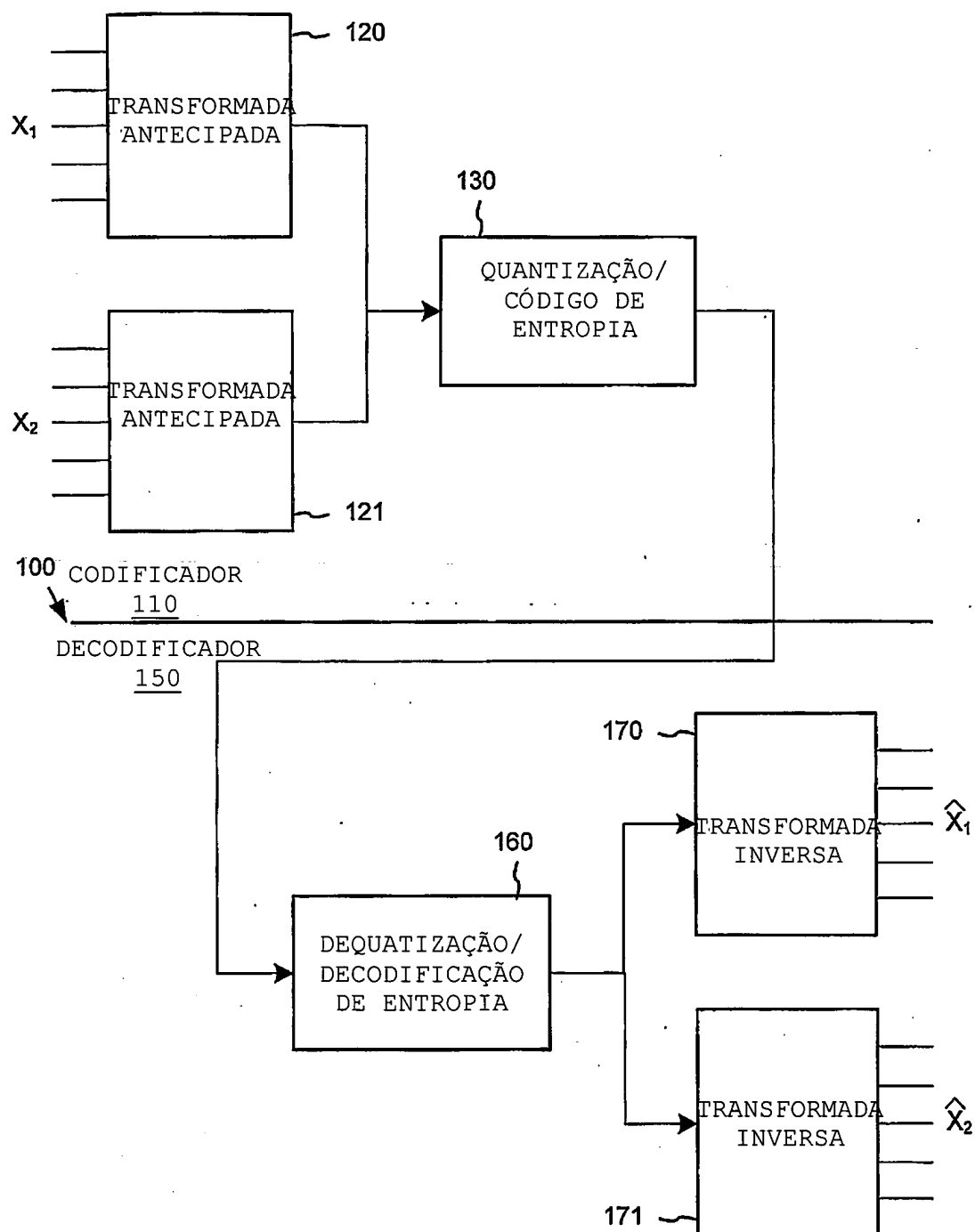


FIG. 2

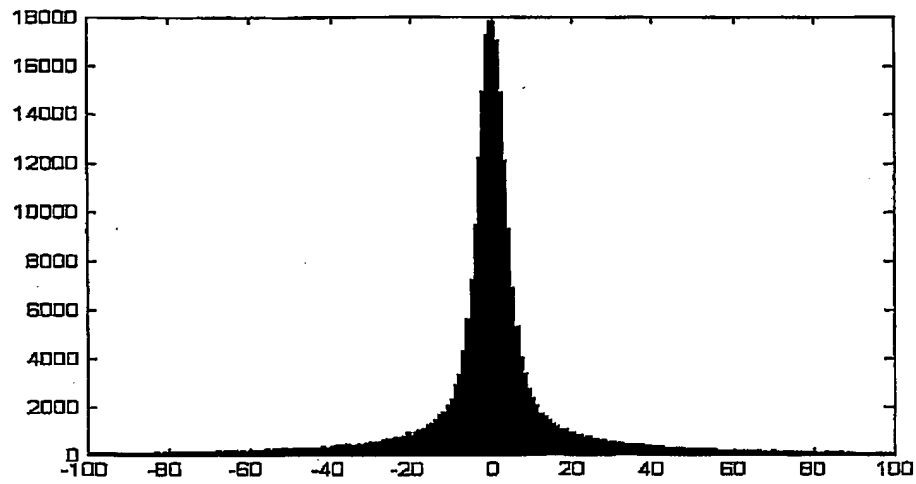


FIG. 3

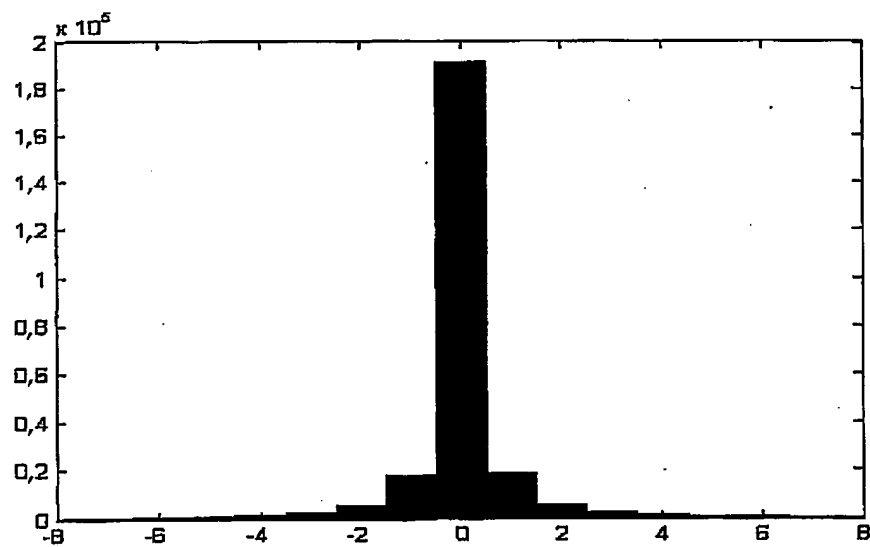


FIG. 4

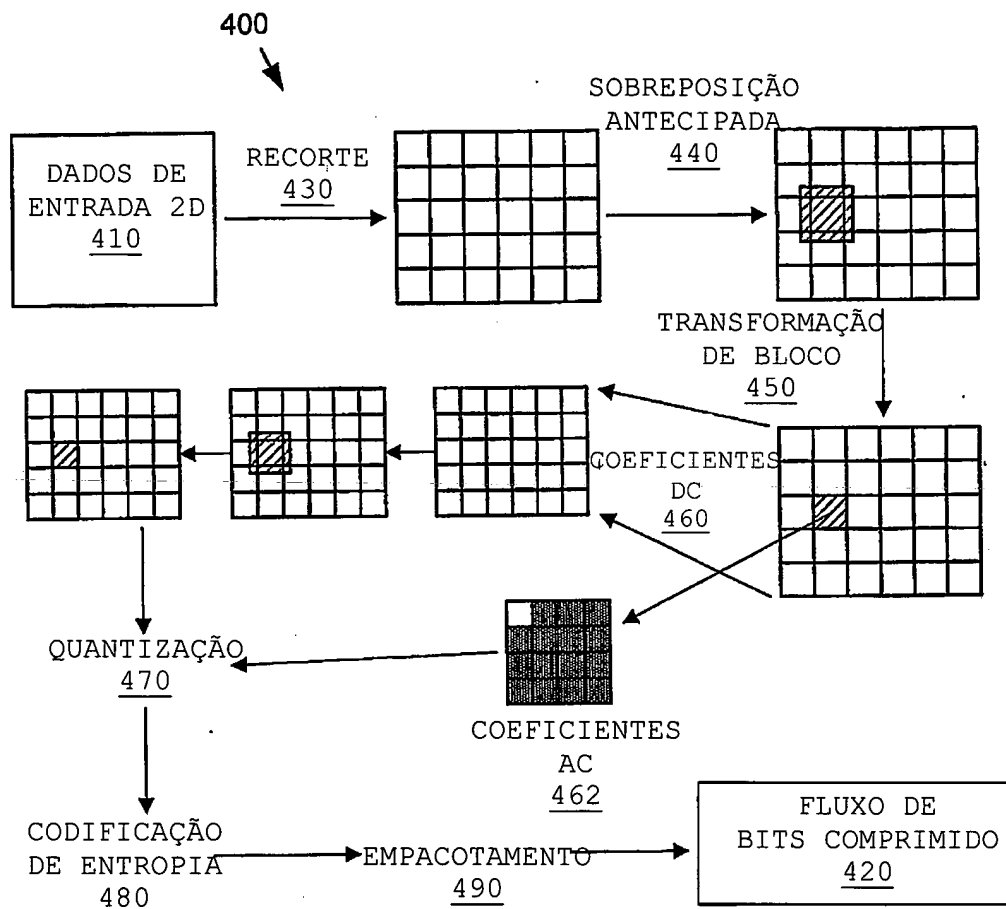


FIG. 5

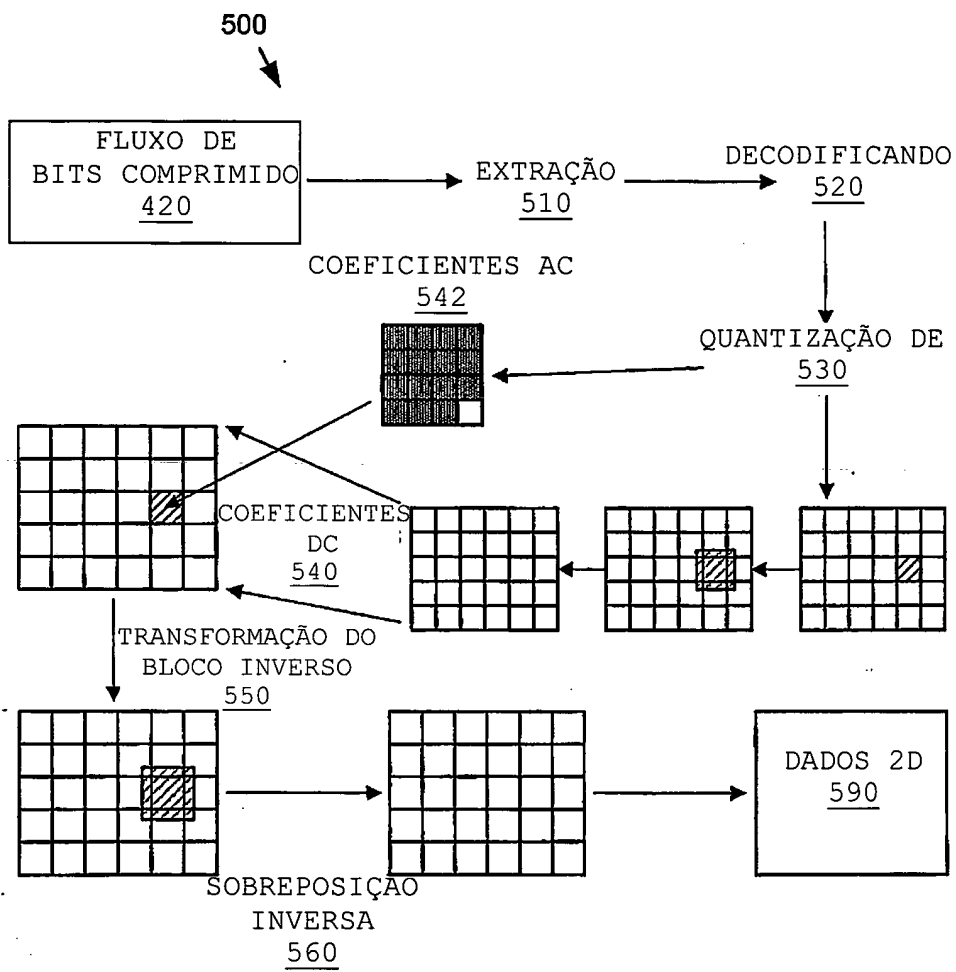


FIG. 6

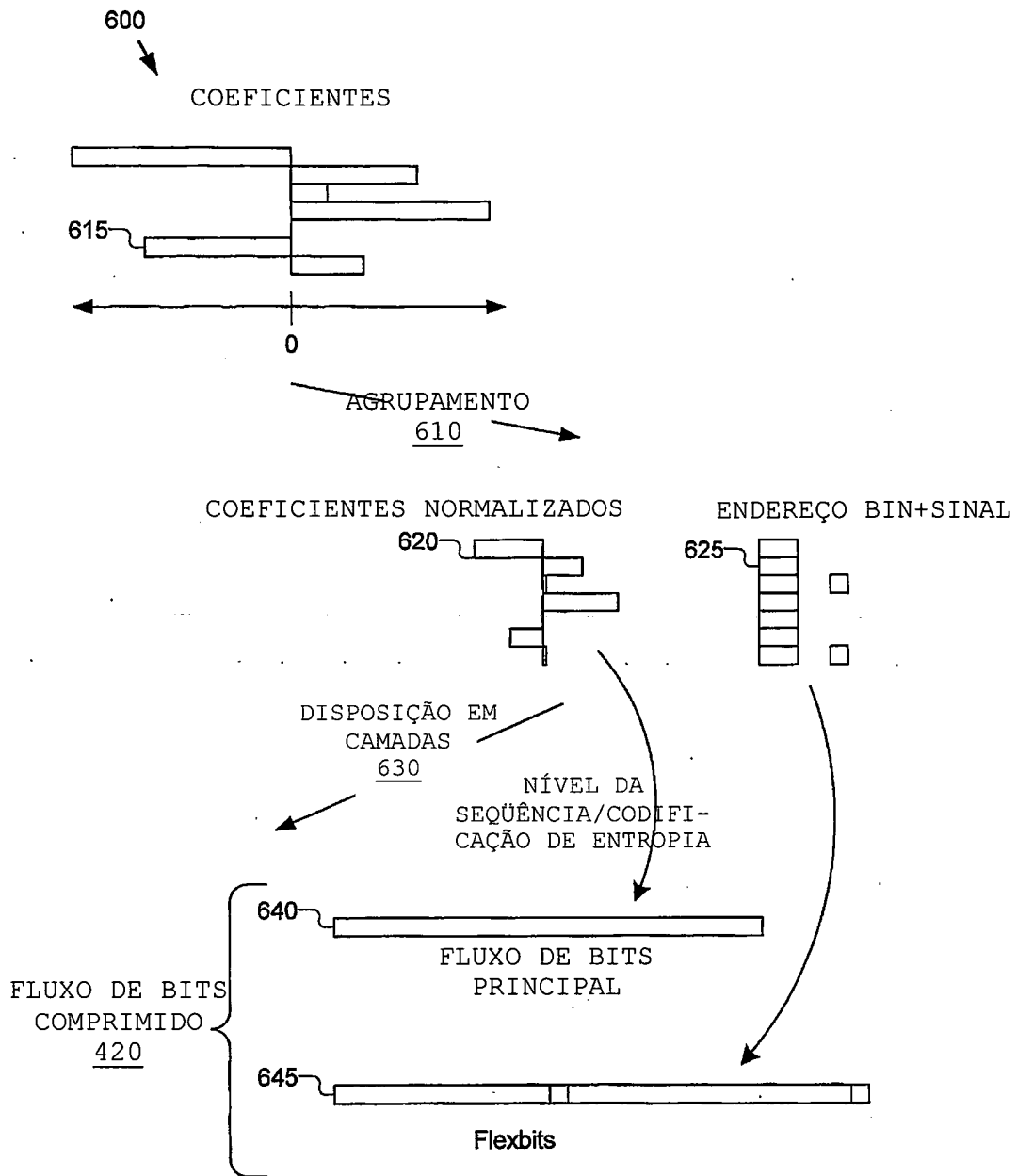


FIG. 7

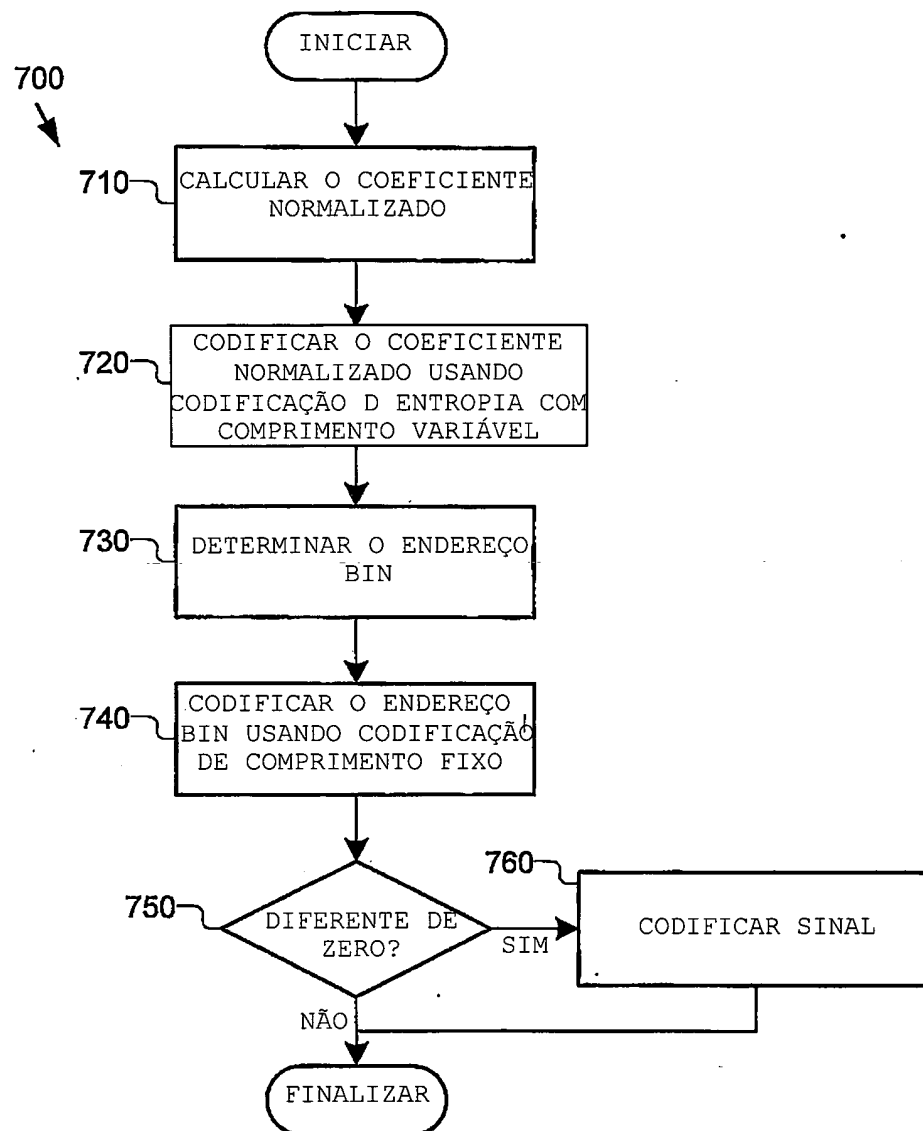


FIG. 8

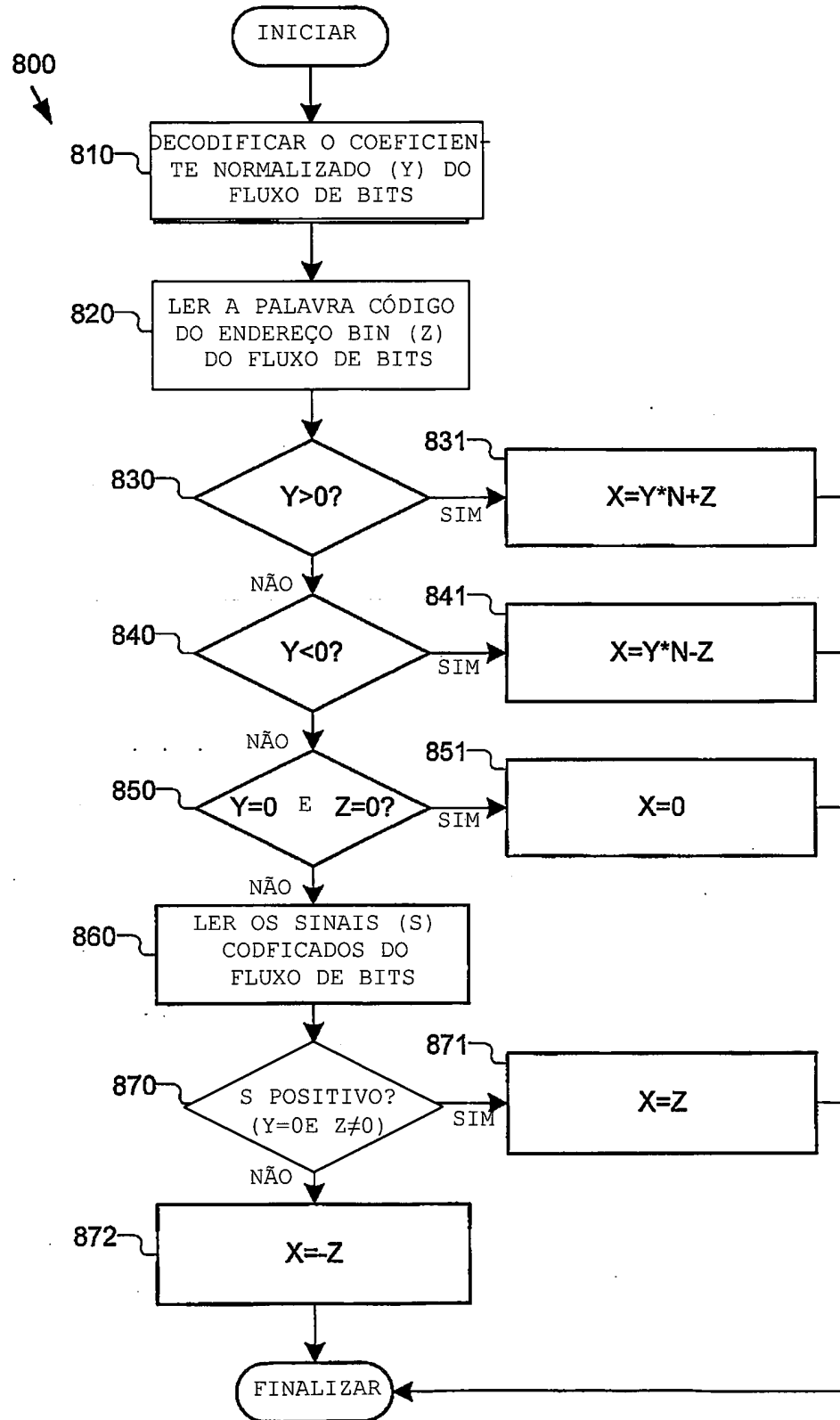


FIG. 9

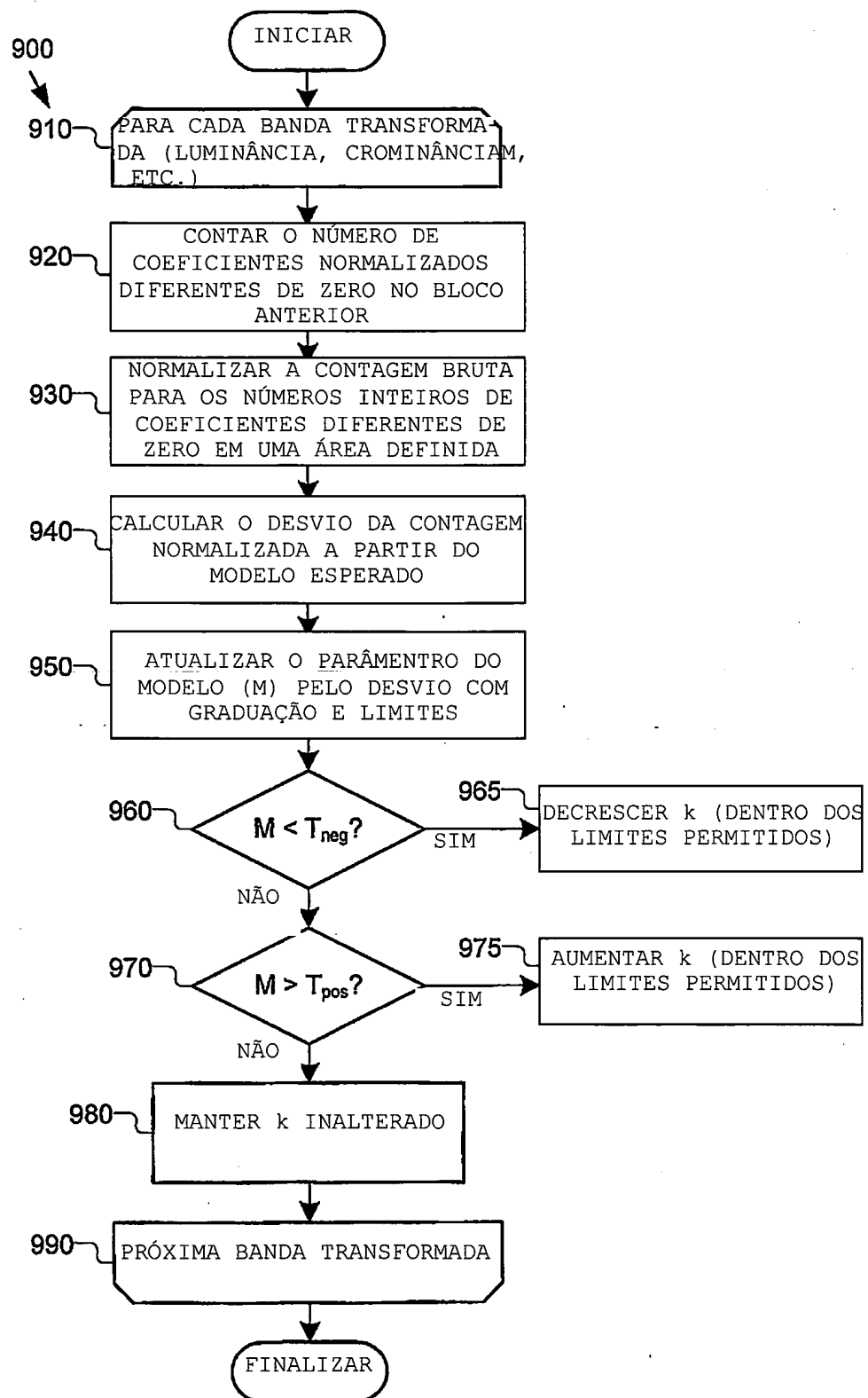


FIG. 10

1000
↓

```

CONSTANTES

MODELO PONDERAL= 70
PESO 0 [3]= { 240/*DC*/, 12/*LP*/, 1/*HP*/ }
PESO 1 [3][16] = { /* PERMITIDO ATÉ 16 CANAIS */
    { 0,240,120,80, 60,48,40,34, 30,27,24,22, 20,18,17,16 },
    { 0,12,6,4,    3,2,2,2,    2,1,1,1,    1,1,1,1 },
    { 0,16,8,5,    4,3,3,2,    2,2,2,1,    1,1,1,1 }
}
PESO 2[6] = { 120,37,2/*YUV 420*/,
              120,18,1/*YUV 422*/ }

ESTRUTURA DOS DADOS DO MODELO:
CAdaptiveModel {
    FLCState /* ESTADO INTERNO OPERANDO -8 a 7* /
              /* FLCState É INICIALIZADO A ZERO */
    FLCBits /* k */
              /* FLCState É INICIALIZADO O 0 PARA PASSE ALTO
*/
              /*4 PARA PASSE BAIXO E 8 PARA DC */
}

FUNÇÃO UpdateModel1MB(
    BANDA Int/*banda 0/1/2 => DC/LP/HP */
    FORMATO DA COR/*Formato da cor da imagem */
    CANAIS INT/*Número de canais croma <=16*/
    CONTAGEM INT[]/*Número de coeficientes em bandas de luminância e
crominância*/
    CAdaptiveModel Model)
{
    /* NORMALIZAÇÃO*/
    iLaplacianMean[0] *= PESO 0 [BANDA]
    se(cf == YUV_420) {
        iLaplacianMean[1] *= PESO 2 [BANDA]
    }
    Ou então(cf == YUV_422) {
        iLaplacianMean[1] *= aPESO 2 [3+BANDA]
    }
    ainda{
        iLaplacianMean[1] *= PESO 1 [BANDA] [iCANAIS - 1]
        Se (banda=HP)
            iLaplacianMean[1] >>= 4
    }
    ...
}

```

FIG. 11

1000
(cont.)
↓

```

...
/* Dar a volta em luma & croma */
para (j = 0; j < 2; j++) {
    iLM = Count[j]
    iMS = Model.FLCState[j]

/* Calcular o desvio a partir da linha de base */
    iDelta = (iLM - MODELWEIGHT) >> 2

/* Poucos coeficientes no MB anterior?
    se (iDelta < -8) {
        iDelta += 4 /* histerese */
        se (iDelta < -16)
            iDelta = -16
        iMS += iDelta
        se (iMS < -8) {
            if (Model.FLCBits [j] == 0)
                iMS = -8
            else {
                se iMS = 0
                Model.FLCBits[j]-- /* bump k down */
            }
        }
    }

/* Poucos coeficientes no MB anterior?
    Ou então (iDelta > 8) {
        iDelta -= 4 /* histerese */
        se (iDelta > 15)
            iDelta = 15
        iMS += iDelta
        se (iMS > 8) {
            iMS = 0
            Model.FLCBits [j]++ /* bump k up */
            (Model.FLCBits [j] > 16)
            Model.FLCBits [j] = 16
        }
    }

    Model.FLCState[j] = iMS
}
}

```

FIG. 12

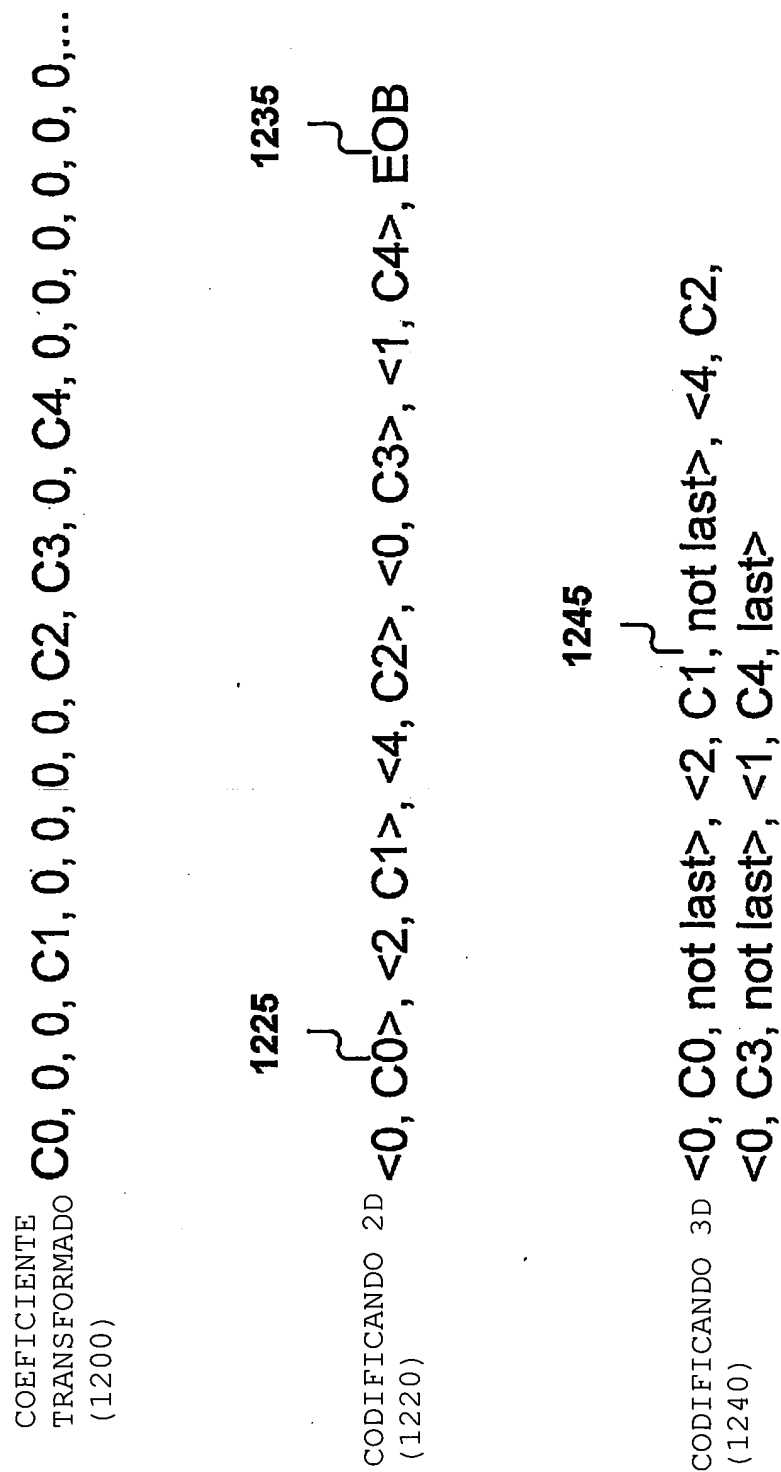


FIG. 13

COEFICIENTE
TRANSFORMADO
(1300) $C_0, 0, 0, C_1, 0, 0, 0, C_2, C_3, 0, C_4, 0, 0, 0, 0, \dots$

CODIFICANDO
 $3^{1/2} D - 2^{1/2} D$
(1340)

$\underbrace{\hspace{10em}}_{3^{1/2}D}$	$\underbrace{\hspace{10em}}_{2^{1/2}D}$
$\underbrace{\hspace{10em}}_{1375}$	$\underbrace{\hspace{10em}}_{1355}$
$\underbrace{\hspace{10em}}_{1365}$	$\underbrace{\hspace{10em}}_{1365}$
$<0, C_0, 2>, <C_1, 4>, <C_2, 0>, <C_3, 1>, <C_4, \text{ÚLTIMO}>$	

FIG. 14

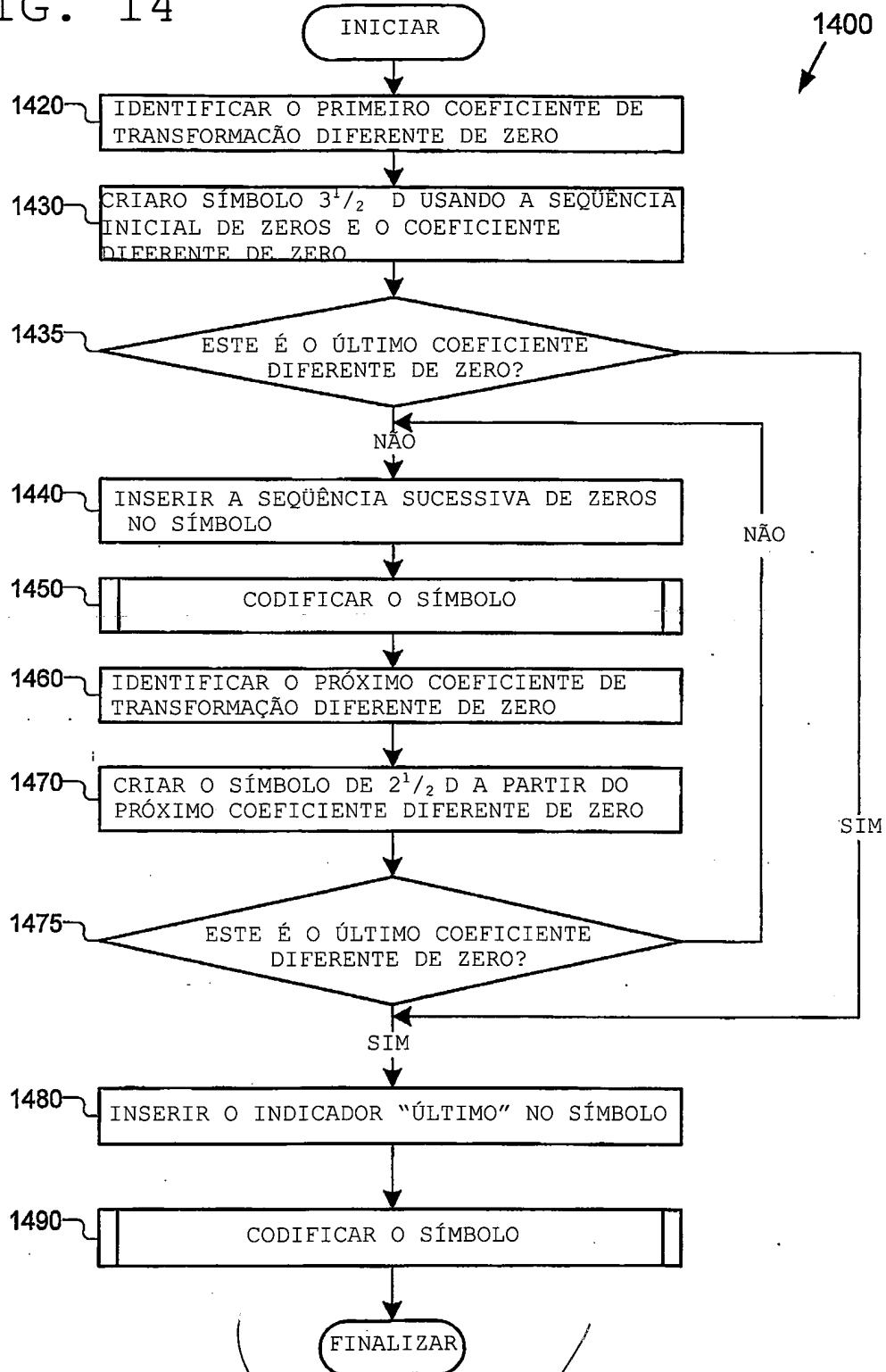


FIG. 15

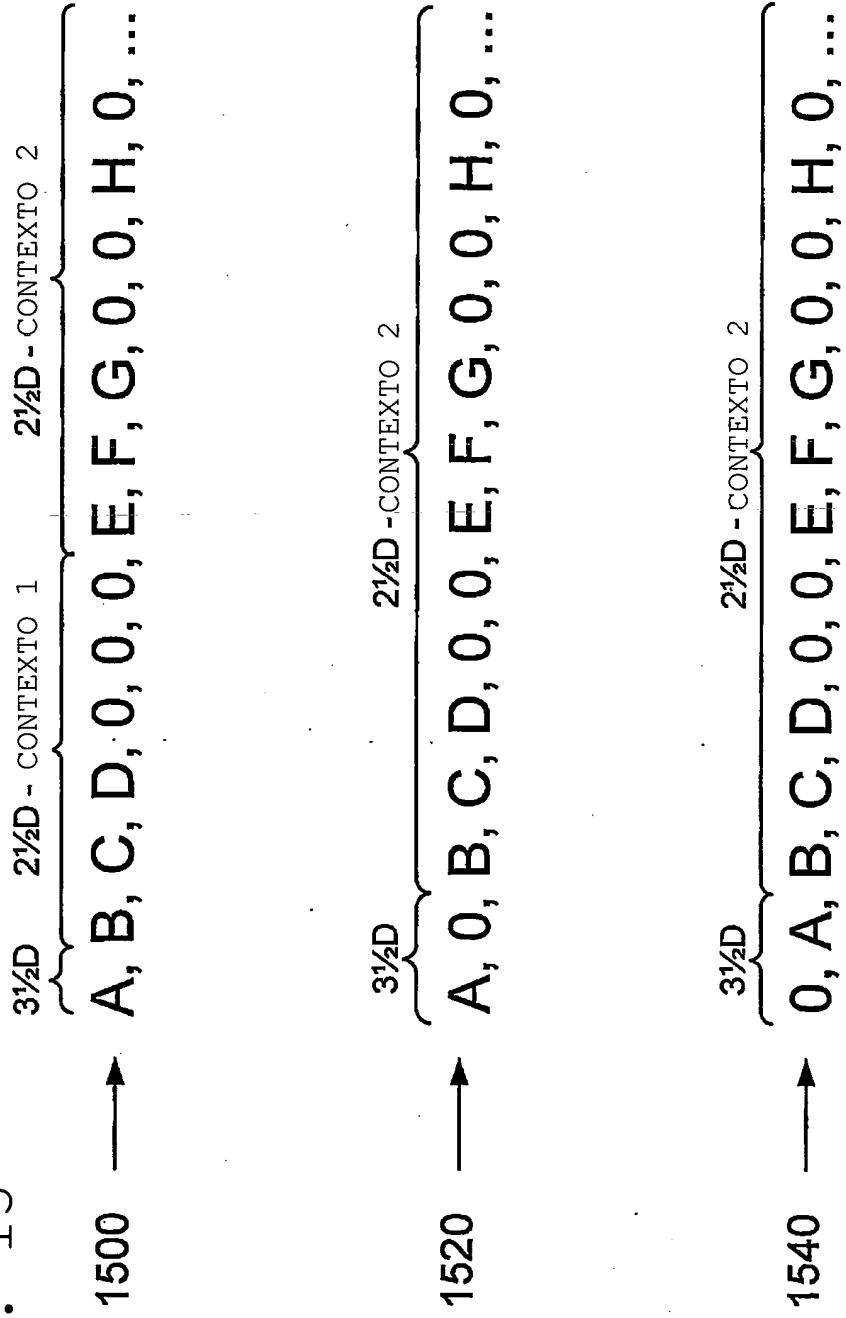


FIG. 16

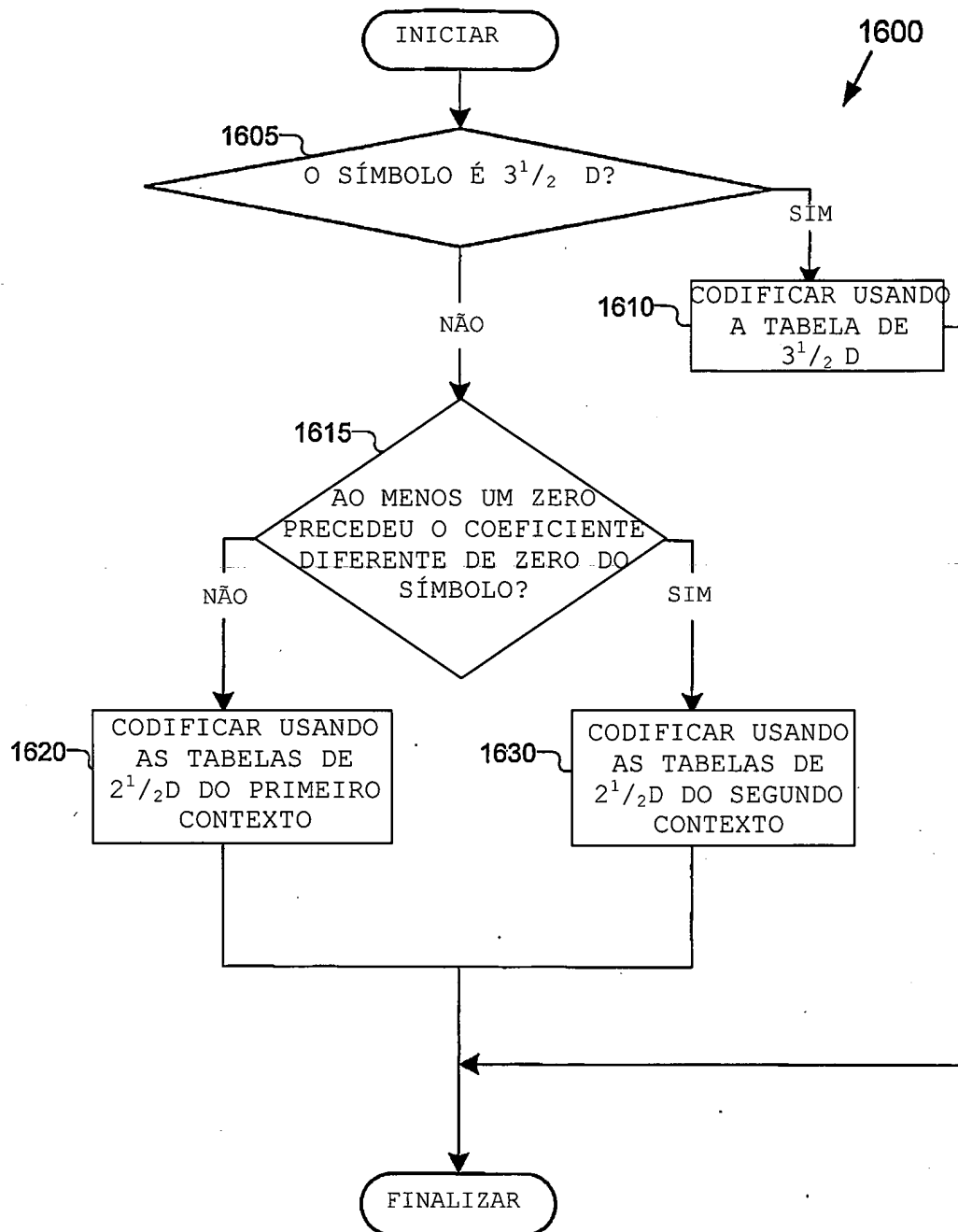


FIG. 17

COEFICIENTE
TRANSFORMADO
(1700)

5, 0, 0, 1, 0, 0, 0, 0, 1, 2, 0, 1, 0, 0, 0, 0, 0, ...

CODIFICAÇÃO REDUZIDO $3^{1/2}$ D- $2^{1/2}$ D (1740)	1745	1755	1765	1775
	{	{	{	{
	<0, 1, 1>	level_5,	run_2, <0, 1>	run_4, <0, 0>
				<1, 1>

FIG. 18

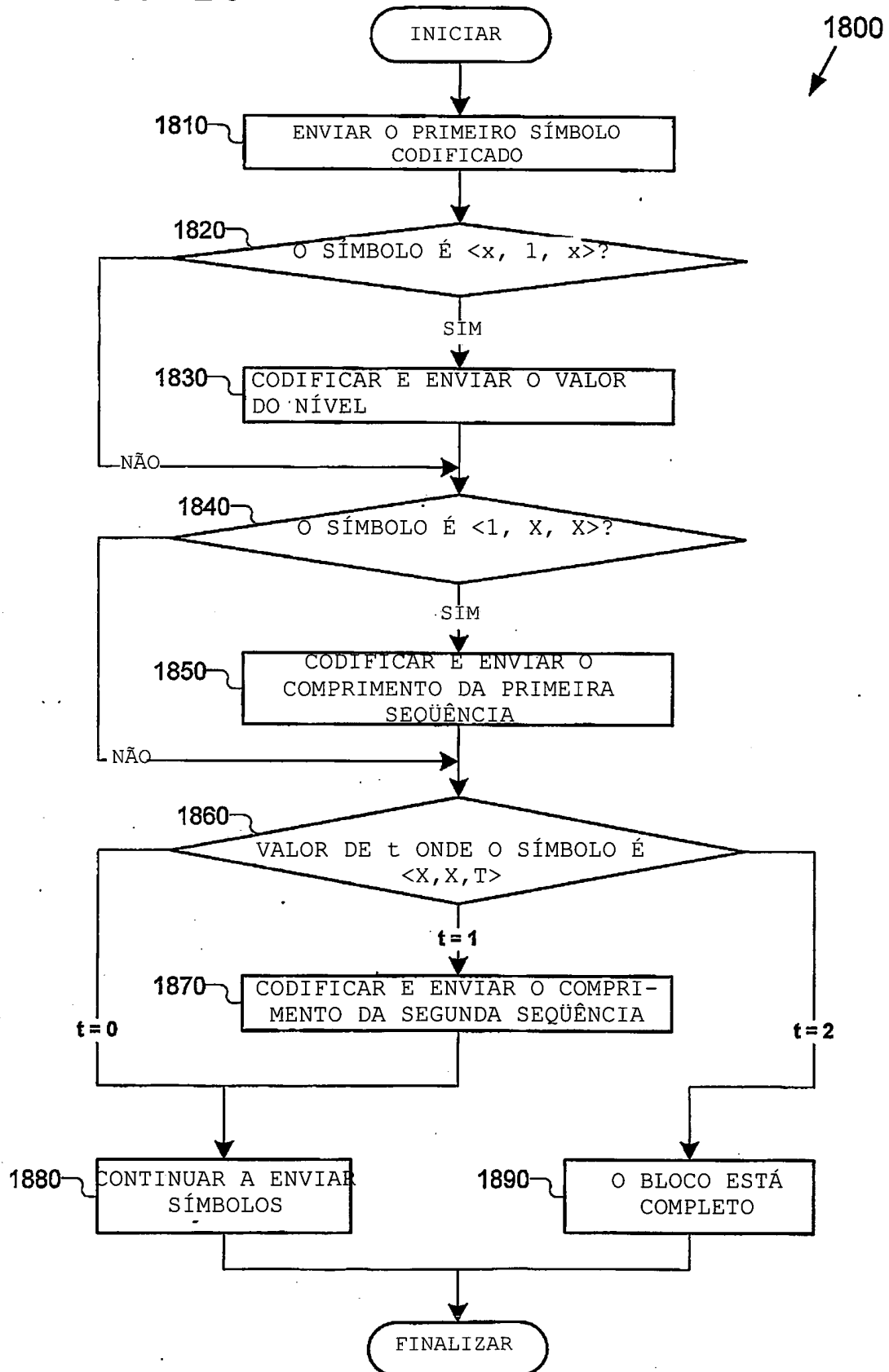


FIG. 19

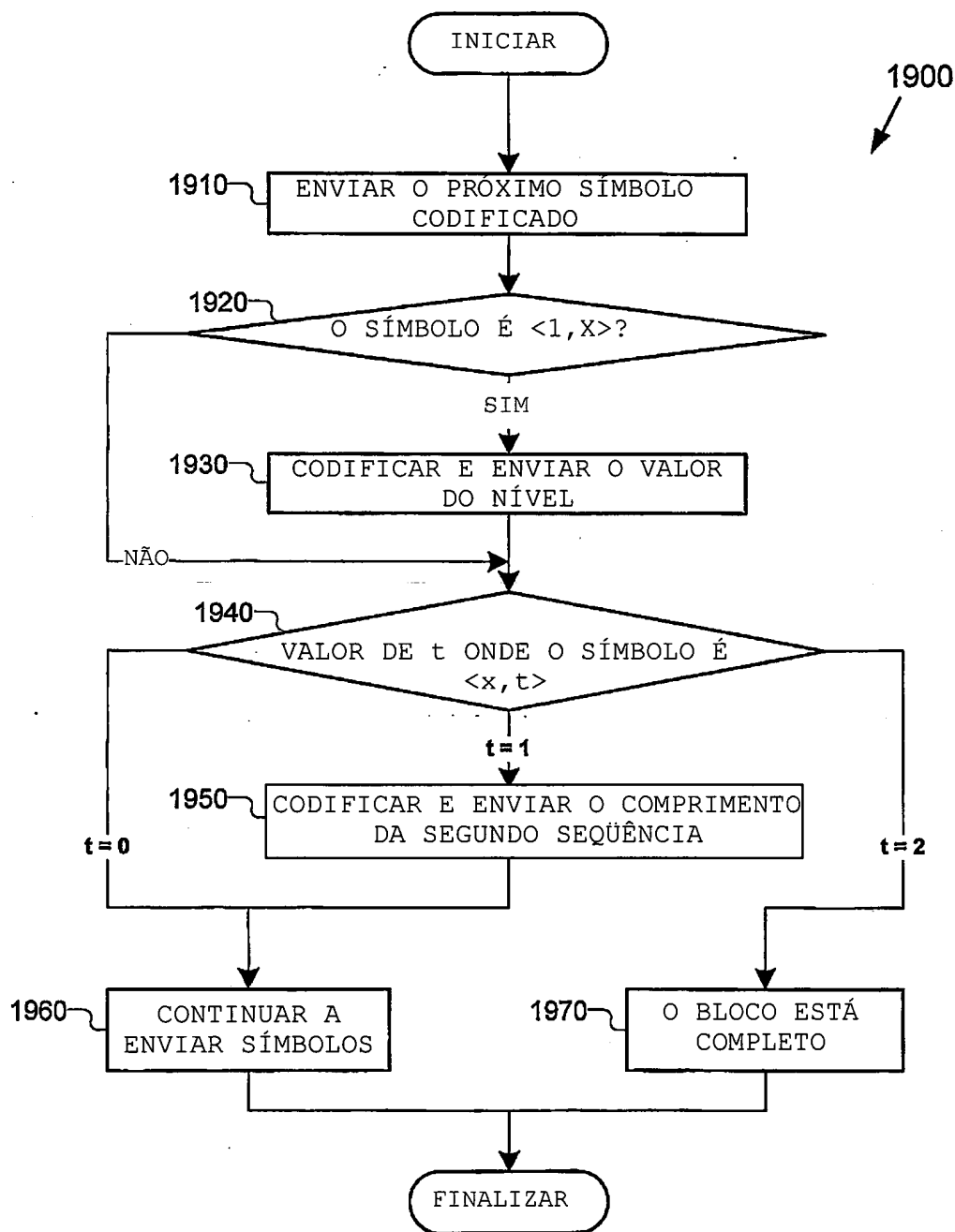


FIG. 20

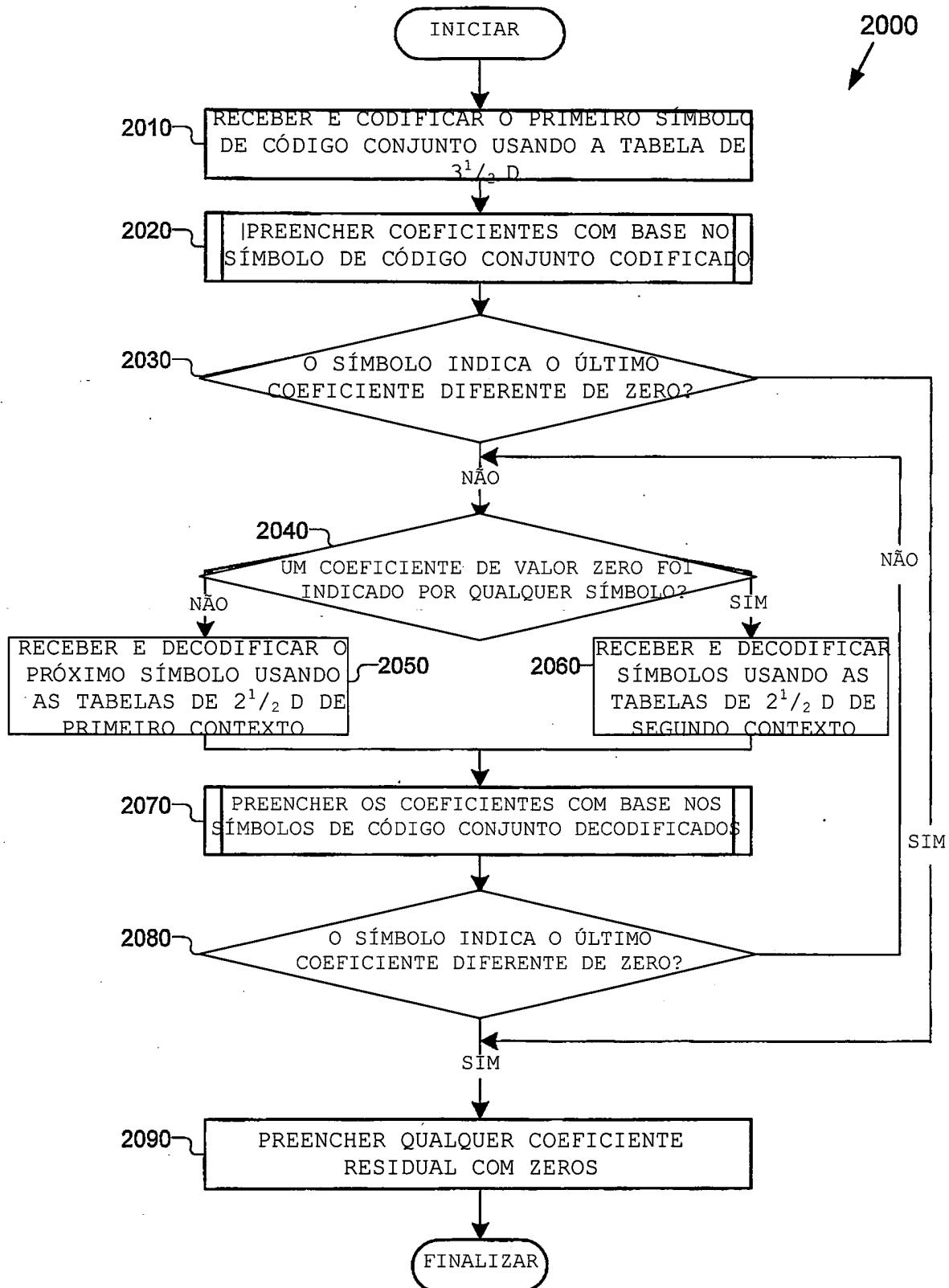


FIG. 21

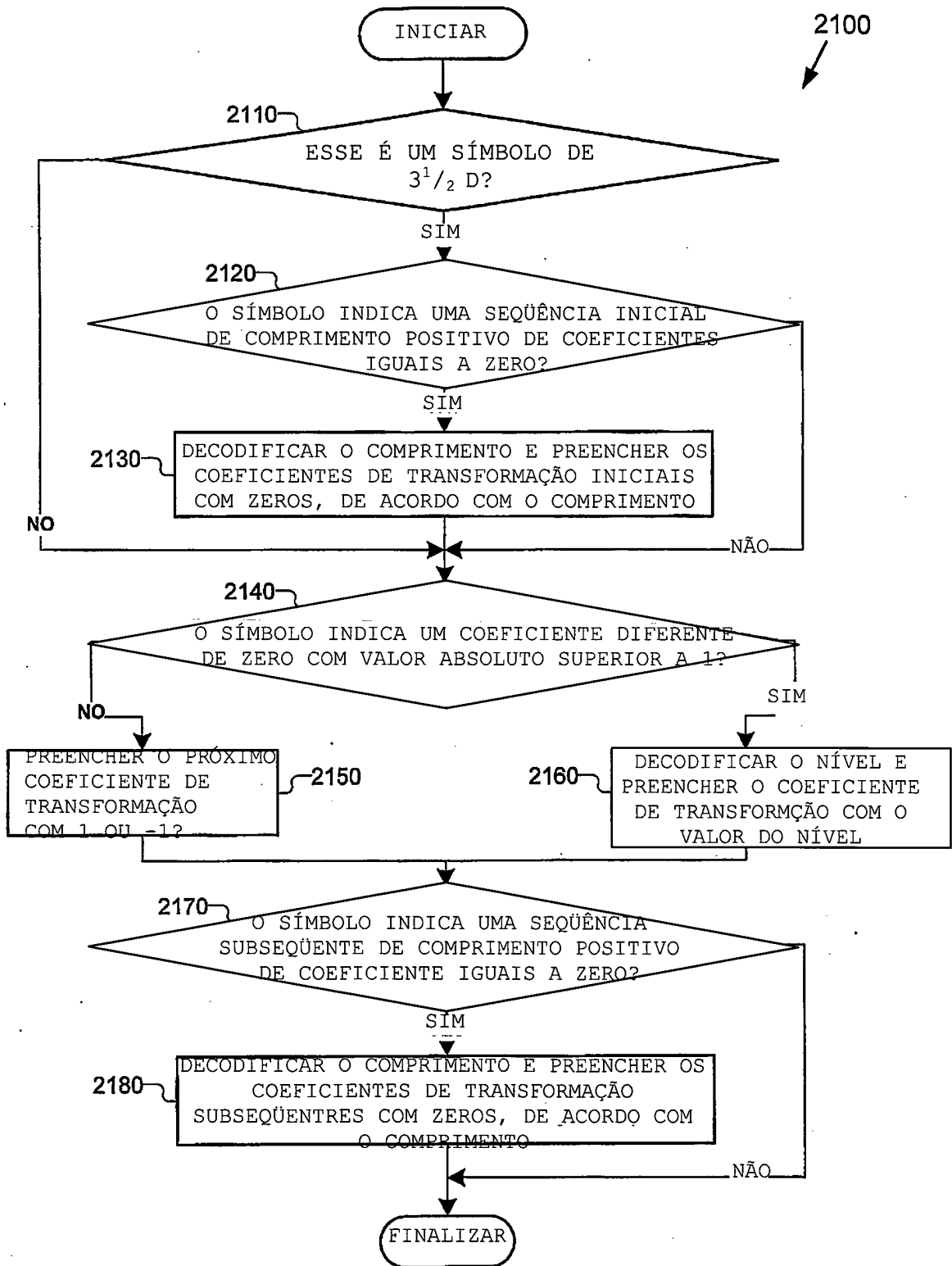
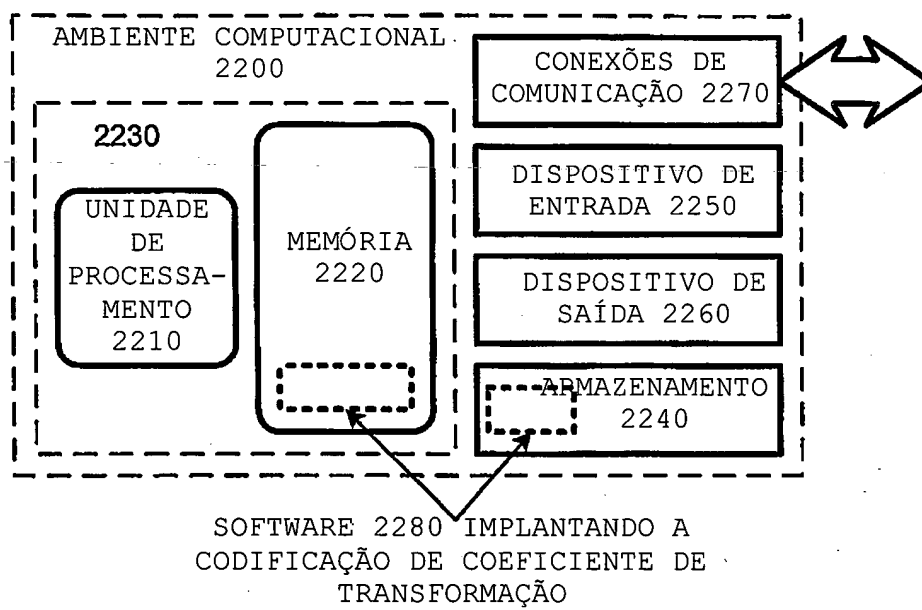


FIG. 22



P1061425-9-3

RESUMO

"CODIFICAÇÃO E DECODIFICAÇÃO EFICIENTE DE BLOCOS DE TRANSFORMAÇÃO".

Trata-se de um codec de mídia digital baseado em blocos de transformação que codifica de modo mais eficiente coeficientes de transformação através da codificação conjunta de coeficientes diferentes de zero junto com sucessões subseqüentes de coeficientes de valor igual a zero. Quando o coeficiente diferente de zero é o último de seu bloco, um último indicador é substituído pelo valor da sucessão no símbolo para aquele coeficiente. Coeficientes iniciais diferentes de zero são indicados em um símbolo especial que codifica em conjunto o coeficiente diferente de zero junto com as sucessões iniciais e subseqüentes de valores iguais a zero. O codec ensaja a codificação múltipla de contextos reconhecendo interrupções nas sucessões de coeficientes diferentes de zero em cada um dos lados da dita interrupção separadamente. O codec também reduz o tamanho da tabela de código indicando em cada símbolo se um coeficiente diferente de zero apresenta valor superior a 1 e se sucessões de zero apresentam valor positivo, codificando separadamente o nível dos coeficientes e o comprimento das sucessões fora dos símbolos.