República Federativa do Brasil
Ministério da Economia
Instituto Nacional da Propriedade Industrial

(11) BR 122020003135-2 B1

(22) Data do Depósito: 06/11/2012

(45) Data de Concessão: 06/07/2021

(54) Título: MÉTODO E DISPOSITIVO PARA DECODIFICAR DADOS DE VÍDEO E MEIO DE ARMAZENAMENTO NÃO TRANSITÓRIO LEGÍVEL POR COMPUTADOR

(51) Int.Cl.: H03M 7/40; H04N 19/10; H04N 19/13; H04N 19/176; H04N 19/18.

(52) CPC: H04N 19/10; H03M 7/4018; H03M 7/4037; H04N 19/13; H04N 19/176; (...).

(30) Prioridade Unionista: 05/11/2012 US 13/669.096; 07/02/2012 US 61/596.049; 08/11/2011 US 61/557.317; 19/01/2012 US 61/588.579; 20/11/2011 US 61/561.909.

(73) Titular(es): VELOS MEDIA INTERNATIONAL LIMITED.

(72) Inventor(es): LIWEI GUO; WEI-JUNG CHIEN; MARTA KARCZEWICZ.

(86) Pedido PCT: PCT US2012063717 de 06/11/2012

(87) Publicação PCT: WO 2013/070610 de 16/05/2013

(85) Data do Início da Fase Nacional: 14/02/2020

(62) Pedido Original do Dividido: BR112014011062-0 - 06/11/2012

(57) Resumo: MÉTODO E DISPOSITIVO PARA DECODIFICAR DADOS DE VÍDEO E MEIO DE ARMAZENAMENTO NÃO TRANSITÓRIO LEGÍVEL POR COMPUTADOR. Um codificador de vídeo é configurado para codificar uma sequência binária indicando uma posição de um último coeficiente significativo dentro de um bloco de vídeo. Um decodificador de vídeo é configurado para decodificar a sequência binária codificada. A sequência pode ser codificada utilizando codificação aritmética binária de contexto adaptativo (CABAC). Aos índices binários da sequência binária pode ser atribuído um contexto. O contexto pode ser determinado de acordo com uma função de mapeamento. Um contexto pode ser designado para um ou mais índices binários onde cada índice é associado com um tamanho de bloco diferente. O último índice binário de um bloco de vídeo 16x16 pode compartilhar um contexto com um último índice binário de um bloco de vídeo 32x32.

“MÉTODO E DISPOSITIVO PARA DECODIFICAR DADOS DE VÍDEO E MEIO DE ARMAZENAMENTO NÃO TRANSITÓRIO LEGÍVEL POR COMPUTADOR”

Pedido dividido do BR1120140110620, depositado em 06.11.2012.

PEDIDOS RELACIONADOS

[001] Esse pedido reivindica o benefício de:

Pedido Provisional dos Estados Unidos Nº 61/557.317, depositado em 8 de novembro de 2011;

Pedido Provisional dos Estados Unidos Nº 61/561.909, depositado em 20 de novembro de 2011;

Pedido Provisional dos Estados Unidos Nº 61/588.579, depositado em 19 de janeiro de 2012; e

Pedido Provisional dos Estados Unidos Nº 61/596.049, depositado em 7 de fevereiro de 2012, cada um dos quais é incorporado pelo presente documento mediante referência em sua totalidade respectiva.

CAMPO TÉCNICO

[002] Essa revelação se refere à codificação de vídeo, e mais especificamente às técnicas para codificar coeficientes de transformada.

ANTECEDENTES

[003] Capacidades de vídeo digital podem ser incorporadas em uma ampla gama de dispositivos, incluindo televisões digitais, sistemas digitais de difusão direta, sistemas de difusão sem fio, assistentes pessoais digitais (PDAs), computadores de mesa ou laptop, computadores tablet, leitores de livro eletrônico, câmeras digitais, dispositivos digitais de gravação, dispositivos de reprodução de mídia digital, dispositivo de videogame, consoles de videogame, telefones de rádio via satélite ou celulares, os assim chamados “smartphones”, dispositivos de teleconferência de vídeo, dispositivos de fluxo contínuo de vídeo, e semelhantes. Os dispositivos de

vídeo digital implementam técnicas de compactação de vídeo tais como aquelas descritas nos padrões definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificação Avançada de Vídeo (AVC), o padrão de Codificação de Vídeo de Alta Eficiência (HEVC) atualmente em desenvolvimento, e extensões de tais padrões. Os dispositivos de vídeo podem transmitir, receber, codificar, decodificar e/ou armazenar informação de vídeo digital mais especificamente mediante implementação de tais técnicas de compactação de vídeo.

[004] Técnicas de compactação de vídeo realizam predição espacial (imagem intra) e/ou predição temporal (imagem inter) para reduzir ou remover a redundância inerente nas sequências de vídeo. Para codificação de vídeo baseada em bloco, uma fatia de vídeo (isto é, um quadro de vídeo ou uma porção de um quadro de vídeo) pode ser dividida em blocos de vídeo, os quais também podem ser referidos como bloco de árvores, unidades de codificação (CUs) e/ou modos de codificação. Os blocos de vídeo em uma fatia codificada intra (I) de uma imagem são codificados utilizando a predição espacial com relação às amostras de referência em blocos vizinhos na mesma imagem. Os blocos de vídeo em uma fatia codificada inter (P ou B) de uma imagem podem usar predição espacial com relação às amostras de referência em blocos vizinhos na mesma imagem ou predição temporal com relação às amostras de referência em outras imagens de referência. As imagens podem ser referidas como quadros, e as imagens de referência podem ser referidas como quadros de referência.

[005] A predição espacial ou temporal resulta em um bloco preditivo para um bloco a ser codificado. Dados residuais representam diferenças de pixel entre o bloco original a ser codificado e o bloco preditivo. Um bloco codificado inter é codificado de acordo com um vetor de movimento que aponta para um bloco de amostras de referência formando o bloco preditivo, e os dados residuais indicando a

diferença entre o bloco codificado e o bloco preditivo. Um bloco codificado intra é codificado de acordo com um modo de codificação intra e os dados residuais. Para compactação adicional, os dados residuais podem ser transformados a partir do domínio de pixel para um domínio de transformada, resultando em coeficientes de transformada residual, os quais podem ser então quantizados. Os coeficientes de transformada quantizados, inicialmente arranjados em um arranjo bidimensional, podem ser escaneados para produzir um vetor unidimensional dos coeficientes de transformada, e codificação de entropia pode ser empregada para se obter adicionalmente compactação.

SUMÁRIO

[006] Em geral, essa revelação descreve técnicas para codificar dados de vídeo. Especificamente, essa revelação descreve técnicas para codificar coeficientes de transformada.

[007] Em um exemplo da revelação, um método de codificar coeficientes de transformada compreende obter uma sequência binária indicando uma posição de um último coeficiente significativo dentro de um bloco de coeficientes de transformada associados com um bloco de vídeo; determinar um contexto para um índice binário da sequência binária com base em um tamanho de bloco de vídeo, em que o contexto é atribuído a pelo menos dois índices binário, em que cada um dos pelo menos dois índices binários é associado com diferentes tamanhos de bloco de vídeo; e codificar a sequência binária utilizando codificação aritmética binária de contexto adaptativo (CABAC) com base ao menos em parte no contexto determinado.

[008] Em outro exemplo da revelação, um método de decodificar coeficientes de transformada compreende obter uma sequência binária codificada indicando uma posição de um último coeficiente significativo dentro de um bloco de coeficientes de transformada associados com um bloco de vídeo, em que a

sequência binária codificada é codificada utilizando CABAC; determinar um contexto para um índice binário da sequência binária codificada com base em um tamanho de bloco de vídeo, em que o contexto é atribuído a pelo menos dois índices binários, em que cada um dos pelo menos dois índices binários é associado com diferentes tamanhos de bloco de vídeo; e decodificar a sequência binária codificada utilizando CABAC com base ao menos em parte no contexto determinado.

[009] Em outro exemplo da revelação, um aparelho configurado para codificar coeficientes de transformada em um processo de codificação de vídeo compreende meio para obter uma sequência binária indicando uma posição de um último coeficiente significativo dentro de um bloco de coeficientes de transformada associados com um meio de bloco de vídeo para determinar um contexto para um índice binário da sequência binária com base em um tamanho de bloco de vídeo, em que o contexto é atribuído a pelo menos dois índices binários, em que cada um dos pelo menos dois índices binários é associado com diferentes tamanhos de bloco de vídeo; e meio para codificar a sequência binária utilizando CABAC com base ao menos em parte no contexto determinado.

[010] Em outro exemplo da revelação, um aparelho configurado para decodificar coeficientes de transformada em um processo de decodificação de vídeo compreende meio para obter uma sequência binária codificada indicando uma posição de um último coeficiente significativo dentro de um bloco de coeficientes de transformada associados com um bloco de vídeo, em que a sequência binária codificada é codificada utilizando CABAC; meio para determinar um contexto para um índice binário da sequência binária codificada com base em um tamanho de bloco de vídeo, em que o contexto é atribuído a pelo menos dois índices binários, em que cada um dos pelo menos dois índices binários é associado com diferentes tamanhos de bloco de vídeo; e meio para decodificar a sequência binária codificada utilizando CABAC com base ao menos em parte no contexto determinado.

[011] Em outro exemplo da revelação, um dispositivo compreende um codificador de vídeo configurado para obter uma sequência binária indicando uma posição do último coeficiente significativo dentro de um bloco de coeficientes de transformada associados com um bloco de vídeo; determinar um contexto para um índice binário da sequência binária com base em um tamanho de bloco de vídeo, em que o contexto é atribuído a pelo menos dois índices binários, em que cada um dos pelo menos dois índices binários é associado com diferentes tamanhos de blocos de vídeo; e codificar a sequência binária utilizando CABAC com base ao menos em parte no contexto determinado.

[012] Em outro exemplo da revelação, um dispositivo compreende um decodificador de vídeo configurado para obter uma sequência binária codificada indicando uma posição do último coeficiente significativo dentro de um bloco de coeficientes de transformada associados com um bloco de vídeo, em que a sequência binária codificada é codificada utilizando CABAC; determinar um contexto para um índice binário da sequência binária codificada com base em um tamanho de bloco de vídeo, em que o contexto é atribuído a pelo menos dois índices binários, em que cada um dos pelo menos dois índices binários é associado com diferentes tamanhos de bloco de vídeo; e decodificar a sequência binária codificada utilizando CABAC com base ao menos em parte no contexto determinado.

[013] Em outro exemplo da revelação, um meio de armazenamento legível por computador não transitório tem instruções nele armazenadas que a partir da execução fazem com que um dispositivo de codificação de vídeo obtenha uma sequência binária indicando uma posição de um último coeficiente significativo dentro de um bloco de coeficientes de transformada associados com um bloco de vídeo; determine um contexto para um índice binário da sequência binária com base em um tamanho de bloco de vídeo, em que o contexto é atribuído a pelo menos dois índices binários, em que cada um dos pelo menos dois índices binários é associado

com diferentes tamanhos de bloco de vídeo, e codifique a sequência binária utilizando CABAC com base ao menos em parte no contexto determinado.

[014] Em outro exemplo da revelação, um meio de armazenamento legível por computador não transitório tem instruções nele armazenadas que a partir da execução fazem com que um dispositivo de decodificação de vídeo obtenha uma sequência binária codificada indicando uma posição de um último coeficiente significativo dentro de um bloco de coeficientes de transformada associados com um bloco de vídeo, em que a sequência binária codificada é codificada utilizando CABAC; determine um contexto para um índice binário da sequência binária codificada com base em um tamanho de bloco de vídeo, em que o contexto é atribuído pelo menos a dois índices binários, em que cada um dos pelo menos dois índices binários é associado com diferentes tamanhos de bloco de vídeo; e decodifique a sequência binária codificada utilizando CABAC com base ao menos em parte no contexto determinado.

[015] Os detalhes de um ou mais exemplos são apresentados nos desenhos anexos e na descrição abaixo. Outras características, objetivos e vantagens serão evidentes a partir da descrição e desenhos, e a partir das reivindicações.

BREVE DESCRIÇÃO DOS DESENHOS

[016] A Figura 1 é um diagrama de blocos ilustrando um sistema exemplar de codificação e decodificação de vídeo.

[017] As Figuras 2A-2D ilustram ordens de varredura de valor de coeficiente exemplar.

[018] A Figura 3 ilustra um exemplo de um mapa de significância em relação a um bloco de valores de coeficiente.

[019] A Figura 4 é um diagrama de blocos ilustrando um codificador de vídeo exemplar que pode implementar as técnicas descritas nessa revelação.

[020] A Figura 5 é um diagrama de blocos ilustrando um codificador de

entropia exemplar que pode implementar as técnicas descritas nessa revelação.

[021] A Figura 6 é um fluxograma ilustrando um exemplo de codificação de um valor de sequência binária indicando a posição do último coeficiente significativo de acordo com as técnicas dessa revelação.

[022] A Figura 7 é um diagrama de blocos ilustrando um decodificador de vídeo exemplar que pode implementar as técnicas descritas nessa revelação.

[023] A Figura 8 é um diagrama de blocos ilustrando um decodificador de entropia exemplar que pode implementar as técnicas descritas nessa revelação.

[024] A Figura 9 é um fluxograma ilustrando um exemplo de decodificação de um valor de sequência binária indicando a posição de um último coeficiente significativo de acordo com as técnicas dessa revelação.

DESCRIÇÃO DETALHADA

[025] Em geral, essa revelação descreve as técnicas para codificação de dados de vídeo. Especificamente, essa revelação descreve as técnicas para codificar os coeficientes de transformada em um processo de codificação e/ou decodificação de vídeo. Na codificação de vídeo baseada em bloco, um bloco de coeficientes de transformada pode ser arranjo em um arranjo bidimensional (2D). Um processo de varredura pode ser realizado para rearranjar o arranjo bidimensional (2D) de coeficientes de transformada em um arranjo unidimensional (1D) ordenado, isto é, vetor, de coeficientes de transformada. Um ou mais elementos de sintaxe podem ser usados para indicar uma posição de um último coeficiente significativo (isto é, coeficiente diferente de zero) dentro do bloco de coeficientes de transformada com base em uma ordem de varredura. A posição do último coeficiente significativo pode ser usada por um codificador de vídeo para otimizar a codificação dos coeficientes de transformada. Similarmente, um decodificador de vídeo pode usar a posição do último coeficiente significativo para otimizar a decodificação dos coeficientes de transformada. Assim, é desejável codificar eficientemente o um ou

mais elementos de sintaxe que indicam uma posição de um último coeficiente significativo.

[026] Essa revelação descreve técnicas para codificar um ou mais elementos de sintaxe que indicam uma posição de um último coeficiente significativo. Em alguns exemplos, todos ou uma porção dos elementos de sintaxe que indicam a posição de um último coeficiente significativo podem ser codificados por entropia de acordo com qualquer uma das seguintes técnicas: Codificação de Comprimento Variável de Contexto Adaptativo (CAVLC), Codificação Aritmética Binária de Contexto Adaptativo (CABAC), Codificação de Entropia de Partição de Intervalo de Probabilidade (PIPE), ou semelhante. Nas técnicas de codificação de entropia que utilizam índices binários que também podem ser referidos como “binários” ou “índices binários” e atribuições de contexto, uma atribuição de contexto comum pode ser utilizada para binários para diferentes tamanhos de bloco de transformada (TU) e/ou diferentes componentes de cor. Dessa maneira, o número total de contextos pode ser reduzido. Mediante redução do número total de contextos um codificador de vídeo e/ou decodificador de vídeo pode mais eficientemente codificar os elementos de sintaxe que indicam uma posição de um último coeficiente significativo, uma vez que o número menor de contextos precisa ser armazenado.

[027] A Figura 1 é um diagrama de blocos ilustrando um sistema de codificação e decodificação de vídeo exemplar 10 que pode ser configurado para utilizar as técnicas para codificar coeficientes de transformada de acordo com os exemplos dessa revelação. Conforme mostrado na Figura 1, o sistema 10 inclui um dispositivo de origem 12 que transmite vídeo codificado para um dispositivo de destino 14 por intermédio de um canal de comunicação 16. Os dados de vídeo codificado podem ser armazenados em um meio de armazenamento 34 ou em um servidor de arquivo 36 e podem ser acessados pelo dispositivo de destino 14

conforme desejado. Quando armazenado em um meio de armazenamento ou servidor de arquivo, o codificador de vídeo 20 pode prover dados de vídeo codificado a outro dispositivo, tal como uma interface de rede, um disco compacto (CD), um queimador de disco de vídeo digital ou Blu-ray (DVD) ou dispositivo de instalação de estampagem, ou outros dispositivos, para armazenar os dados de vídeo codificados para o meio de armazenamento. Similarmente, um dispositivo separado do decodificador de vídeo 30, tal como uma interface de rede, leitor de CD ou DVD, ou semelhante, pode recuperar os dados de vídeo codificado a partir de um meio de armazenamento e providos os dados recuperados ao decodificador de vídeo 30.

[028] O dispositivo de origem 12 e o dispositivo de destino 14 podem compreender qualquer um de uma variedade de dispositivos, incluindo dispositivos de mesa, computadores notebook (isto é, laptop), computadores tablet, aparelhos de conversão de sinais, aparelhos telefônicos de mão, tais como os assim chamados telefones inteligentes, televisões, câmeras, dispositivos de exibição, aparelhos de reprodução de meios digitais, consoles de jogos de vídeo, ou semelhante. Em muitos casos, tais dispositivos podem ser equipados para comunicação sem fio. Portanto, o canal de comunicação 16 pode compreender um canal sem fio, um canal cabeado, ou uma combinação de canais sem fio e cabeados adequados para a transmissão de dados de vídeo codificado. Similarmente, o servidor de arquivo 36 pode ser acessado pelo dispositivo de destino 14 através de qualquer conexão de dados padrão, incluindo conexão da Internet. Isso pode incluir um canal sem fio (por exemplo, uma conexão Wi-Fi), uma conexão cabeada (por exemplo, DSL, modem a cabo, etc.), ou uma combinação dos dois, que é adequada para acessar os dados de vídeo codificado armazenados em um servidor de arquivo.

[029] Técnicas para codificar coeficientes de transformada, de acordo com os exemplos dessa revelação, podem ser aplicadas na codificação de vídeo em suporte de qualquer uma de uma variedade de aplicações de multimídia, tal como

transmissões de televisão pelo ar, transmissões de televisão a cabo, transmissões de televisão via satélite, transmissões de vídeo de fluxo contínuo, por exemplo, por intermédio da Internet, codificação de vídeo digital para armazenamento em um meio de armazenamento de dados, decodificação do vídeo digital armazenado em um meio de armazenamento de dados, ou outras aplicações. Em alguns exemplos, o sistema 10 pode ser configurado para suportar transmissão de vídeo de via única ou de duas vias para suportar aplicações tais como fluxo contínuo de vídeo, reprodução de vídeo, transmissão de vídeo, e/ou telefonia de vídeo.

[030] No exemplo da Figura 1, o dispositivo de origem 12 inclui uma fonte de vídeo 18, um codificador de vídeo 20, um modulador/demodulador 22 e um transmissor 24. No dispositivo de origem 12, a fonte de vídeo 18 pode incluir uma fonte tal como um dispositivo de captura de vídeo, tal como uma câmera de vídeo, um arquivo de vídeo contendo vídeo previamente capturado, uma interface de alimentação de vídeo para receber vídeo a partir de um provedor de conteúdo de vídeo, e/ou um sistema gráfico de computador para gerar dados gráficos de computador como o vídeo de origem, ou uma combinação de tais fontes. Como um exemplo, a fonte de vídeo 18 é uma câmera de vídeo, o dispositivo de origem 12 e o dispositivo de destino 14 podem formar os assim chamados telefones de câmera ou fontes de vídeo. Contudo, as técnicas descritas nessa revelação podem ser aplicáveis na codificação de vídeo em geral, e podem ser aplicadas em aplicações sem fio e/ou cabeadas, ou aplicação na qual os dados de vídeo codificado são armazenados em um disco local.

[031] O vídeo capturado, pré-capturado, ou gerado por computador pode ser codificado pelo codificador de vídeo 20. A informação de vídeo codificado pode ser modulada através de um modem 22, de acordo com um padrão de comunicação, tal como um protocolo de comunicação sem fio, e transmitida para o dispositivo de destino 14 através da interface de saída 24. Modem 22 pode incluir vários

misturadores, filtros, amplificadores ou outros componentes projetados para a modulação do sinal. O transmissor 24 pode incluir circuitos concebidos para transmissão de dados, incluindo amplificadores, filtros, e uma ou mais antenas.

[032] O vídeo capturado, pré-capturado, ou gerado por computador que é codificado pelo codificador de vídeo de 20 também pode ser armazenado num meio de armazenamento 34 ou um servidor de arquivos 36 para consumo posterior. O meio de armazenamento 34 pode incluir discos Blu-ray, DVDs, CD-ROMs, memória flash, ou qualquer outra mídia de armazenamento digital adequado para o armazenamento de vídeo codificado. O vídeo codificado armazenado no meio de armazenamento 34 pode então ser acessado por dispositivo de destino 14 para decodificação e reprodução.

[033] O servidor de arquivos 36 pode ser qualquer tipo de servidor capaz de armazenar vídeo codificado e transmitir esse vídeo codificado para o dispositivo de destino 14. Servidores de arquivo exemplares incluem um servidor de rede (por exemplo, para um sítio da rede), um servidor FTP, dispositivo de armazenamento ligados em rede (NAS), uma unidade de disco local, ou qualquer outro tipo de dispositivo capaz de armazenar dados de vídeo codificado e transmitir os mesmos para um dispositivo de destino. A transmissão de dados de vídeo codificado a partir do servidor de arquivo 36 pode ser uma transmissão de fluxo contínuo, uma transmissão de transferência baixada, ou uma combinação das duas. O servidor de arquivo 36 pode ser acessado pelo dispositivo de destino 14 através de qualquer conexão padrão de dados, incluindo uma conexão de Internet. Isso pode incluir um canal sem fio (por exemplo, uma conexão Wi-Fi), uma conexão cabeada (por exemplo, DSL, modem a cabo, Ethernet, USB, etc.), ou uma combinação das duas que seja adequada para acessar os dados de vídeo codificado armazenados em um servidor de arquivos.

[034] O dispositivo de destino 14, no exemplo da Figura 1, inclui um receptor

26, um modem 28, um decodificador de vídeo 30, e um dispositivo de exibição 32. O receptor 26 do dispositivo de destino 14 recebe informação através do canal 16, e o modem 28 demodula a informação para produzir um fluxo de bits demodulado para o decodificador de vídeo 30. A informação comunicada através do canal 16 pode incluir uma variedade de informação de sintaxe gerada pelo codificador de vídeo 20 para uso pelo decodificador de vídeo 30 na decodificação dos dados de vídeo. Tal sintaxe também pode ser incluída com os dados de vídeo codificados armazenados no meio de armazenamento 34 ou servidor de arquivo 36. Cada um do codificador de vídeo 20 e decodificador de vídeo 30 pode formar parte de um codificador-decodificador respectivo (CODEC) que é capaz de codificar ou decodificar os dados de vídeo.

[035] O dispositivo de exibição 32 pode ser integrado com, ou externo ao dispositivo de destino 14. Em alguns exemplos, o dispositivo de destino 14 pode incluir um dispositivo de exibição integrado e também ser configurado para estabelecer interface com um dispositivo de exibição externo. Em outros exemplos, o dispositivo de destino 14 pode ser um dispositivo de exibição. Em geral, o dispositivo de exibição 32 exibe os dados de vídeo decodificados a um usuário, e pode compreender qualquer um de uma variedade de dispositivos de exibição tal como um display de cristal líquido (LCD), um display de plasma, um display de diodo de emissão de luz orgânica (OLED), ou outro tipo de dispositivo de display.

[036] No exemplo da Figura 1, o canal de comunicação 16 pode compreender qualquer meio de comunicação sem fio ou cabeado, tal como um espectro de radiofrequência (RF) ou uma ou mais linhas físicas de transmissão, ou qualquer combinação de meios sem fio e cabeados. O canal de comunicação 16 pode formar parte de uma rede baseada em pacote, tal como uma rede de área local, uma rede de área remota, ou uma rede global tal como a Internet. O canal de comunicação 16 representa geralmente qualquer meio de comunicação adequado,

ou grupo de meios de comunicação diferentes, para transmitir dados de vídeo a partir do dispositivo de origem 12 para o dispositivo de destino 14, incluindo qualquer combinação adequada de meios cabeados ou sem fio. O canal de comunicação 16 pode incluir roteadores, comutadores, estações base ou qualquer outro equipamento que possa ser útil para facilitar a comunicação a partir do dispositivo de origem 12 para o dispositivo de destino 14.

[037] O codificador de vídeo 20 e o decodificador de vídeo 30 podem operar de acordo com um padrão de compactação de vídeo, tal como o padrão de Codificação de Vídeo de Alta Eficiência (HEVC) atualmente sendo desenvolvido pelo Joint Collaboration Team on Video Coding (JCT-VC) of ITU-T Video Coding Experts Group (VCEG) e ISO/IEC Motion Picture Experts Group (MPEG), e pode se ajustar ao Modelo de Teste (HM) HEVC. O codificador de vídeo 20 e o decodificador de vídeo 30 podem operar de acordo com o projeto recente do padrão HEVC, referido como “HEVC Working Draft 5” ou “WD5”, é descrito no documento JCTVC-G1103, Bross et all., “WD5: Working Draft 5 of High efficiency video coding (HEVC)”, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 e ISO/IEC JCT1/SC29/WG11, 7º Encontro: Genebra, CH, novembro de 2012. Adicionalmente, outro projeto de trabalho recente de HEVC, Working Draft 7 é descrito no documento HCTVC-11003, Bross et al., “High Efficiency Video Coding (HEVC) Text Specification Draft 7”, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 e ISO/IEC JCT1/SC29/WG11, 9º Encontro: Genebra, Suíça, 27 de abril de 2012 a 7 de maio de 2012. Alternativamente, o codificador de vídeo 20 e o decodificador de vídeo 30 podem operar de acordo com outros padrões patenteados ou da indústria, tal como o padrão ITU-T H.264, referido alternativamente como MPEG-4, Parte 10, Codificação Avançada de Vídeo (AVC), ou extensões de tais padrões. As técnicas dessa revelação, contudo, não são limitadas a qualquer padrão de codificação específico. Outros exemplos incluem MPEG-2 e ITU-T H.263.

[038] Embora não mostrado na figura 1, em alguns aspectos, o codificador de vídeo 20 e o decodificador de vídeo 30 podem ser individualmente integrados com um codificador e decodificador de áudio, e podem incluir unidades apropriadas MUX-DEMUX, ou outro hardware e software, para lidar com a codificação de áudio e vídeo em um fluxo de dados comum ou fluxos de dados separados. Se for o caso, as unidades MUX-DEMUX podem estar de acordo com o protocolo multiplexador ITU H.223, ou outros protocolos, como o Protocolo de Datagrama de Usuário (UDP).

[039] O codificador de vídeo 20 e o decodificador de vídeo 30 podem ser individualmente implementados como qualquer um de uma variedade de circuitos de codificador adequado, tal como um ou mais processadores, incluindo os microprocessadores, processadores de sinais digitais (DSPs), circuitos integrados de aplicação específica (ASICs), conjunto de porta programável no campo (FPGAs), lógica discreta, software, hardware, firmware ou qualquer combinação desses. Quando as técnicas são aplicadas parcialmente em software, um dispositivo pode armazenar instruções de software num meio de leitura por computador adequado, não transitório e executar as instruções em hardware utilizando um ou mais processadores para executar as técnicas desta revelação. Cada codificador de vídeo 20 e decodificador de vídeo 30 pode ser incluído em um ou mais transdutores ou decodificadores, cada um dos quais pode ser integrado como parte de um codificador/decodificador combinada (CODEC) num respectivo dispositivo.

[040] O codificador de vídeo 20 pode implementar qualquer uma ou todas as técnicas dessa revelação para codificar coeficientes de vídeo em um processo de codificação de vídeo. Similarmente, o codificador de vídeo 30 pode implementar qualquer uma ou todas essas técnicas para codificar coeficientes de transformada em um processo de codificação de vídeo. Um codificador de vídeo, conforme descrito nessa revelação pode se referir a um codificador de vídeo ou a um decodificador de vídeo. Similarmente, uma unidade de codificação de vídeo pode se

referir a um codificador de vídeo ou a um decodificador de vídeo. Similarmente, a codificação de vídeo pode se referir a uma codificação de vídeo ou decodificação de vídeo.

[041] O codificador de vídeo 20 e o decodificador de vídeo 30 da Figura 1 representam exemplos de codificadores de vídeo configurados para obter uma sequência binária indicando uma posição de um último coeficiente significativo dentro de um bloco de vídeo, determinar um contexto para um índice binário da sequência binária com base em um tamanho de bloco de vídeo, em que o contexto é atribuído a pelo menos dois índices binários, em que cada um dos pelo menos dois índices binários é associado com diferentes tamanhos de bloco de vídeo; e codificar a sequência binária utilizando codificação aritmética binária de contexto adaptativo (CABAC) com base ao menos em parte no contexto determinado.

[042] Para codificação de vídeo de acordo com o padrão HEVC atualmente em desenvolvimento, um quadro de vídeo pode ser dividido em unidades de codificação. Uma unidade de codificação (CU) geralmente se refere a uma região de imagem que serve como uma unidade básica a qual as várias ferramentas de codificação são aplicadas para compactação de vídeo. Uma CU normalmente tem um componente de luminância, denotado como Y, e dois componentes de crominância, denotados como U e V. Dependendo do formato de amostragem de vídeo, o tamanho dos componentes U e V, em termos do número de amostras, pode ser o mesmo que ou diferente do tamanho do componente Y. Uma CU é tipicamente quadrada, e pode ser considerada como similar a um assim chamado macro bloco, por exemplo, sob outros padrões de codificação de vídeo tal como ITU-T H.264. A codificação de acordo com alguns dos aspectos atualmente propostos do padrão HEVC em desenvolvimento será descrita nesse pedido com o propósito de ilustração. Contudo, as técnicas descritas nessa revelação podem ser úteis para outros processos de codificação de vídeo, tais como aqueles definidos de acordo

com H.264 ou outro padrão ou processos de codificação de vídeo, patenteados. Os esforços de padronização HEVC se baseiam em um modelo de um dispositivo de codificação de vídeo referido como Modelo de Teste (HM) HEVC. O HM presume várias capacidades dos dispositivos de codificação de vídeo em relação aos dispositivos de acordo com, por exemplo, ITU-T H.264/AVC. Por exemplo, enquanto H.264 proporciona nove modos de codificação de predição intra, HM proporciona tantos quantos 34 modos de codificação de predição intra.

[043] Uma seqüência de vídeo, normalmente inclui uma série de imagens ou imagens de vídeo. Um grupo de imagens (GOP) compreende geralmente uma série de uma ou mais das imagens de vídeo. Um GOP pode incluir dados de sintaxe em um cabeçalho do GOP, um cabeçalho de uma ou mais das imagens, ou em outro lugar, que descreve uma série de imagens incluídas no GOP. Cada fatia de uma imagem pode incluir dados de sintaxe de fatia que descrevem um modo de codificação para a respectiva fatia. O codificador de vídeo 20 normalmente opera em blocos de vídeo dentro fatias de vídeo individuais, para codificar os dados de vídeo. Um bloco de vídeo pode corresponder a um nó de codificação dentro de uma CU. Os blocos de vídeo podem ter tamanhos fixados ou variados, e podem variar em tamanho de acordo com um padrão de codificação especificado.

[044] De acordo com HM, uma CU pode incluir uma ou mais unidades de predição (PUs) e/ou uma ou mais unidades de transformada (TUs). Dados de sintaxe dentro de um fluxo de bits podem definir uma unidade de codificação maior (LCU), a qual é a maior CU em termos do número de pixels. Em geral, uma CU tem um propósito semelhante a o de um macro bloco de H.264, exceto que uma CU não tem uma distinção de tamanho. Assim, uma CU pode ser dividida em sub-CUs. EM geral, referências nessa revelação a uma CU podem se referir a uma unidade de codificação maior de uma imagem ou uma sub-CU de uma LCU. Uma LCU pode ser dividida em sub-CUs, e cada sub-CU pode ser dividida adicionalmente em sub-CUs.

Os dados de sintaxe para um fluxo de bits podem definir um número máximo de vezes em que uma LCU pode ser dividida, referida como profundidade de CU. Consequentemente, um fluxo de bits também pode definir uma unidade de codificação menor (SCU). Essa revelação também utiliza o termo “bloco” ou “porção” para se referir a qualquer uma de uma CU, PU, ou TU. Em geral, “porção” pode se referir a qualquer subconjunto de um quadro de vídeo.

[045] Uma LCU pode ser associada com uma estrutura de dados de quadtree. Em geral, uma estrutura de dados de quadtree inclui um nó por CU, onde um nó raiz corresponde à LCU. Se uma CU for dividida em quatro sub-CUs, o nó correspondendo à CU inclui 4 nós folha, cada um dos quais corresponde a uma das sub-CUs. Cada nó da estrutura de dados de quadtree pode prover dados de sintaxe para a CU correspondente. Por exemplo, um nó na quadtree pode incluir um sinalizador de divisão, indicando se a CU correspondendo ao nó é dividida em sub-CUs. Os elementos de sintaxe para uma CU podem ser definidos de forma recursiva, e podem depender de se a CU é dividida em sub-CUs. Se uma CU não for dividida adicionalmente, ela é referida como uma CU folha. Nessa revelação, 4 sub-CUs de uma CU folha também serão referidas como CUs folha embora não haja divisão explícita da Cu folha. Por exemplo, se uma CU em tamanho 16x16 não for dividida adicionalmente, as quatro sub-Cus 8x8 também serão referidas como CUs folha embora a CU 16x16 nunca tenha sido dividida.

[046] Além disso, as TUs das CUs folha também podem ser associadas com as estruturas de dados de quadtree respectivas. Isto é, uma CU folha pode incluir uma quadtree indicando como a CU folha é dividida em TUs. Essa revelação se refere à quadtree indicando como uma LCU é dividida como uma quadtree de CU e a quadtree indicando como uma CU folha é dividida em TUs como uma quadtree de TU. O nó raiz de uma quadtree de TU geralmente corresponde a uma CU folha, enquanto que o nó raiz de uma quadtree de CU geralmente corresponde a uma

LCU. As TUs da quadtree de TU que não são divididas são referidas como TUs folha.

[047] Uma CU folha pode incluir uma ou mais unidades de predição (PUs). Em geral, uma PU representa toda ou uma porção da CU correspondente, e pode incluir os dados para recuperar uma amostra de referência para a PU. Por exemplo, quando a PU é codificada no modo inter, a PU pode incluir os dados definindo um vetor de movimento para a PU. Os dados definindo o vetor de movimento podem descrever, por exemplo, um componente horizontal do vetor de movimento, um componente vertical do vetor de movimento, uma resolução para o vetor de movimento (por exemplo, exatidão de um quarto de pixel ou exatidão de um oitavo de pixel), um quadro de referência para o qual o vetor de movimento aponta, e/ou uma lista de referência (por exemplo, lista 0 ou lista 1) para o vetor de movimento. Os dados para a CU folha definindo a PU(s) também pode descrever, por exemplo, a divisão da CU em uma ou mais PUs. Os modos de divisão podem diferir dependendo de se a CU não é codificada, é codificada no modo de predição intra, ou codificada no modo de predição inter. Para codificação intra, uma PU pode ser tratada da mesma forma que uma unidade de transformada de folha descrita abaixo.

[048] Como exemplo, o HM suporta predição em vários tamanhos de PU. Assumindo-se que o tamanho de uma CU particular é 2Nx2N, o HM suporta predição intra em tamanhos de PU de 2Nx2N ou NxN, e predição inter em tamanhos simétricos PU de 2Nx2N, 2NxN, Nx2N, ou NxN. O HM também suporta divisão assimétrica para predição inter em tamanhos PU de 2NxnU, 2NxnD, nLx2N e nRx2N. Na divisão assimétrica, uma direção de uma CU não está dividida, enquanto a outra direção é dividida em 25% e 75%. A porção da CU correspondendo à porção de 25% é indicada por um “n” seguido por uma indicação de “acima”, “abaixo”, “esquerda” ou “direita”. Assim, por exemplo, “2NxnU” refere-se a uma CU 2Nx2N dividida horizontalmente com uma PU de 2Nx0,5N na parte superior e uma PU de

2Nxl,5 N na parte inferior.

[049] Nessa revelação, “NxN” e “N por N” pode ser utilizado indiferentemente para designar as dimensões de pixels de um bloco de vídeo, em termos de dimensões vertical e horizontal, por exemplo, 16x16 pixels ou 16 por 16 pixels. Em geral, um bloco de 16x16 terá 16 pixels em uma direção vertical (Y = 16) e 16 pixels na direção horizontal (x = 16). Da mesma forma, um bloco NxN geralmente tem N pixels na direção vertical e N pixels na direção horizontal, onde N representa um valor inteiro não negativo. Os pixels de um bloco podem ser dispostos em filas e colunas. Além disso, os blocos não necessitam, necessariamente, ter o mesmo número de pixels na direção horizontal como na direção vertical. Por exemplo, os blocos podem compreender nxm pixels, em que M não é necessariamente igual a N.

[050] Para codificar um bloco (por exemplo, uma unidade de predição de dados de vídeo), um preditor para o bloco é primeiramente derivado. O preditor, também referido como um bloco preditivo pode ser derivado quer seja através de predição intra (I) (isto é, predição espacial) ou predição inter (P ou B) (isto é, predição temporal). Portanto, algumas unidades de predição podem ser codificadas intra (I) utilizando predição espacial com relação às amostras de referência em blocos de referência adjacentes no mesmo quadro (ou fatia), e outras unidades de predição podem ser codificadas inter (P) de forma unidirecional ou codificadas inter (B) de forma bidirecional com relação aos blocos das amostras de referência em outros quadros previamente codificados (ou fatias). Em todo caso, as amostras de referência podem ser usadas para formar um bloco preditivo para um bloco a ser codificado.

[051] A partir da identificação de um bloco preditivo, a diferença entre o bloco de dados de vídeo original e seu bloco preditivo é determinada. Essa diferença pode ser referida como os dados residuais de predição, e indica as diferenças de

pixel entre os valores de pixel no bloco e os valores codificados e de pixel no bloco preditivo selecionado para representar o bloco codificado. Para obter melhor compactação, os dados residuais de predição podem ser transformados, por exemplo, utilizando uma transformada discreta de cosseno (DCT), uma transformada de número inteiro, uma transformada Karhunen-Loeve (K-L), ou outra transformada.

[052] Os dados residuais em um bloco de transformada, tal como uma TU, podem ser arranjados em um arranjo bidimensional (2D) de valores de diferença de pixel residindo no domínio de pixel, espacial. Uma transformada converte os valores residuais de pixel em um arranjo bidimensional de coeficientes de transformada em um domínio de transformada, tal como um domínio de frequência. Para compactação adicional, os coeficientes de transformada podem ser quantizados antes da codificação de entropia. Um codificador de entropia aplica então codificação de entropia, tal como CAVLC, CABAC, PIPE, ou semelhante, aos coeficientes quantizados de transformada.

[053] Para codificar por entropia um bloco de coeficientes de transformada quantizados, um processo de varredura é normalmente realizado de tal modo que o arranjo bidimensional (2D) de coeficientes de transformada quantizados em um bloco é rearranjado, de acordo com uma ordem de varredura específica, para um arranjo unidimensional (1D) ordenado, isto é, vetor de coeficientes de transformada. A codificação de entropia é então aplicada ao vetor dos coeficientes de transformada. A varredura dos coeficientes de transformada em uma unidade de transformada serializa o arranjo 2D dos coeficientes de transformada para o codificador de entropia. Um mapa de significância pode ser gerado para indicar as posições dos coeficientes significativos (isto é, diferentes de zero). A varredura pode ser aplicada aos níveis de varredura de coeficientes significativos (isto é, diferentes de zero), e/ou para codificar sinais dos coeficientes significativos.

[054] Em HEVC, a informação de posição dos coeficientes de transformada

significativos (por exemplo, o mapa de significância) é primeiramente codificada para uma TU para indicar a localização do último coeficiente diferente de zero na ordem de varredura. O mapa de significância e a informação de nível (os valores absolutos e sinais dos coeficientes) são codificados para cada coeficiente em uma ordem de varredura inversa.

[055] Depois de quaisquer transformadas para produzir coeficientes de transformação, o codificador de vídeo 20 pode realizar quantização dos coeficientes de transformação. Quantização geralmente refere-se a um processo em que os coeficientes de transformada são quantizados para possivelmente reduzir a quantidade de dados utilizados para representar os coeficientes, proporcionando a compactação adicional. O processo de quantização pode reduzir a profundidade do bit associado com alguns ou todos os coeficientes. Por exemplo, um valor de n bits pode ser arredondado para um valor de m bits durante quantização, onde n é maior do que m. Em alguns exemplos, o codificador de vídeo 20 pode utilizar uma ordem de varredura predefinida para explorar os coeficientes de transformada quantizados para produzir um vetor serializado que pode ser codificado por entropia. Em outros exemplos, o codificador de vídeo 20 pode realizar uma varredura adaptativa.

[056] As Figuras 2A-2D ilustram algumas ordens de varredura, exemplares, diferentes. Outras ordens de varredura definidas, ou ordens de varredura adaptativas (que mudam) também podem ser usadas. A Figura 2A ilustra uma ordem de varredura em ziguezague, a Figura 2B ilustra uma ordem de varredura horizontal, a Figura 2C ilustra uma ordem de varredura vertical, e a Figura 2D ilustra uma ordem de varredura diagonal. Combinações dessas ordens de varredura também podem ser definidas e usadas. Em alguns exemplos, as técnicas dessa revelação podem ser especificamente aplicáveis durante a codificação de um assim chamado mapa de significância no processo de codificação de vídeo.

[057] Um ou mais elementos de sintaxe podem ser definidos para indicar

uma posição de um último coeficiente significativo (isto é, coeficiente diferente de zero), o qual pode depender da ordem de varredura associada com um bloco de coeficientes. Por exemplo, um elemento de sintaxe pode definir uma posição de coluna de um último coeficiente significativo dentro de um bloco de valores de coeficiente e outro elemento de sintaxe pode definir uma posição de fileira do último coeficiente significativo dentro de um bloco de valores de coeficiente.

[058] A Figura 3 ilustra um exemplo de um mapa de significância relativo a um bloco de valores de coeficiente. O mapa de significância é mostrado à direita, em que sinalizadores de um bit identificam os coeficientes no bloco de vídeo na esquerda que são significativos, isto é, diferentes de zero. Em um exemplo, dado um conjunto de coeficientes significativos (por exemplo, definidos por um mapa de significância) e uma ordem de varredura, uma posição de um último coeficiente significativo pode ser definida. No padrão HEVC emergente, os coeficientes de transformada podem ser agrupados em uma amostra. A amostra pode compreender uma TU inteira, ou em alguns casos, TUs podem ser subdivididas em amostras menores. O mapa de significância e a informação de nível (valor absoluto e sinal) são codificados para coeficiente em uma amostra. Em um exemplo, uma amostra consiste em 16 coeficientes consecutivos na ordem de varredura inversa (por exemplo, diagonal, horizontal ou vertical) para uma TU de 4x4 e para uma TU de 8x8. Para as TUs de 16x16 e 32x32, os coeficientes dentro de um sub-bloco de 4x4 são tratados como uma amostra. Os elementos de sintaxe são codificados e sinalizados para representar a informação de nível de coeficiente dentro de uma amostra. Em um exemplo, todos os símbolos são codificados em uma ordem de varredura inversa. As técnicas dessa revelação podem aperfeiçoar a codificação de um elemento de sintaxe usado para definir essa posição do último coeficiente significativo de um bloco de coeficientes.

[059] Como um exemplo, as técnicas dessa revelação podem ser usadas

para codificar a posição do último coeficiente significativo de um bloco de coeficientes (por exemplo, uma TU ou uma amostra de uma TU). Então, após codificar a posição do último coeficiente significativo, a informação de nível e sinal pode ser codificada. A codificação da informação de nível e sinal pode processar de acordo com uma abordagem de cinco passagens mediante codificação dos símbolos a seguir na ordem de varredura inversa (por exemplo, para uma TU ou para uma amostra da TU):

[060] significant_coeff_flag (abbr. sigMapFlag): esse sinalizador pode indicar a significância de cada coeficiente em uma amostra. Um coeficiente com um valor de um ou maior é considerado como sendo significativo.

coeff_abs_level_greater1_flag (abbr. gr1Flag): esse sinalizador pode indicar se o valor absoluto do coeficiente é maior do que um para os coeficientes diferentes de zero (isto é, coeficientes com *MapFlag* como 1).

coeff_abs_level_greater2_flag (abbr. gr2Flag): esse sinalizador pode indicar se o valor absoluto do coeficiente é maior do que dois para os coeficientes com um valor absoluto maior do que um (isto é, coeficientes com *grlFlag* como 1).

coeff_sign_flag (abbr. signFlag): esse sinalizador pode indicar a informação de sinal para os coeficientes diferentes de zero. Por exemplo, um zero para esse sinalizador indica um sinal positivo, enquanto que um 1 indica um sinal negativo.

coeff_abs_level_remain (abbr. levelRem): é o valor absoluto restante de um nível de coeficiente de transformada. Para esse sinalizador, o valor absoluto do coeficiente-x é codificado (abs(nível)-x) para cada coeficiente com a amplitude maior do que x o valor de x depende dos atuais de gr1Flag e gr2Flag.

[061] Dessa maneira, os coeficientes de transformada para uma TU ou para uma amostra de uma TU podem ser codificados. Em todo caso, as técnicas dessa revelação, que dizem respeito à codificação de um elemento de sintaxe usado para definir a posição do último coeficiente significativo de um bloco de coeficientes, pode

também ser usada com outros tipos de técnicas para por fim codificar a informação de nível e sinal dos coeficientes de transformada. A abordagem de cinco passagens para significância de codificação, informação de nível e sinal é apenas uma técnica exemplar que pode ser usada após a codificação da posição do último coeficiente significativo de um bloco, como apresentado nessa revelação.

[062] Após a varredura, os coeficientes de transformada quantizados para formar um vetor unidimensional, o codificador de vídeo 20 pode codificar por entropia o vetor unidimensional dos coeficientes de transformada. O codificador de vídeo 20 também pode codificar por entropia os elementos de sintaxe associados com os dados de vídeo codificado para uso pelo decodificador de vídeo 30 na decodificação dos dados de vídeo. A codificação por entropia pode ser realizada de acordo com uma das seguintes técnicas: CAVLC, CABAC, codificação aritmética binária de contexto adaptativo baseada em sintaxe (SBAC), codificação PIPE ou outra metodologia de codificação de entropia. Para realizar CAVLC, o codificador de vídeo 20 pode selecionar um código de comprimento variável para um símbolo a ser transmitido. As palavras-chave na codificação de comprimento variável (VLC) podem ser construídas de tal modo que códigos relativamente mais curtos correspondem aos símbolos mais prováveis, enquanto que códigos mais longos correspondem aos símbolos menos prováveis. Desse modo, o uso de VLC pode atingir uma economia de bits, por exemplo, utilizando palavras-chave de comprimento igual para cada símbolo a ser transmitido.

[063] As técnicas de codificação de entropia dessa revelação são descritas especificamente como sendo aplicáveis à CABAC, embora as técnicas também possam ser aplicáveis à CAVLC, SBAC, PIPE ou outras técnicas. Deve-se observar que PIPE usa princípios similares àqueles da codificação aritmética.

[064] Em geral, símbolos de dados de codificação utilizando CABAC podem envolver uma ou mais das seguintes etapas:

(1) Binarização: Se um símbolo a ser codificado tiver um valor não binário, ele é mapeado para uma sequência de assim chamados “binários”. Cada binário pode ter um valor de “0” ou “1”.

(2) Atribuição de Contexto: A cada binário (no modo regular) é atribuído um contexto. Um modelo de contexto determina como um contexto para um determinado binário é calculado com base na informação disponível para o binário, tal como valores de símbolos previamente codificados ou número binário.

(3) Codificação de binário: Os binários são codificados com um codificador aritmético. Para codificar um binário, o codificador aritmético requer como uma entrada uma probabilidade do valor do binário, isto é, uma probabilidade de que o valor do binário seja igual a “0”, e uma probabilidade de que o valor do binário seja igual a “1”. A possibilidade (estimada) de cada contexto é representada por um valor inteiro denominado “estado de contexto”. Cada contexto tem um estado, e assim o estado (isto é, probabilidade estimada) é o mesmo para os binários atribuídos a um contexto, e difere entre os contextos.

(4) Atualização de estados: A probabilidade (estado) para um contexto selecionado é atualizada com base no valor codificado efetivo do binário (por exemplo, se o valor do binário for “1”, a probabilidade de “1’s” é aumentada).

[065] O que se segue é uma técnica de binarização exemplar dos elementos de sintaxe de último coeficiente significativo que podem ser realizados pelo codificador de vídeo 20. Os elementos de sintaxe do último coeficiente significativo podem incluir uma posição de fileira e coluna de um último coeficiente significativo dentro de um bloco bidimensional (isto é, uma coordenada x e uma coordenada y). Para um bloco 8x8, há oito diferentes possibilidades para a última posição de um coeficiente dentro de uma coluna ou fileira, isto é, 0, 1..., 7. Oito binários diferentes são usados para representar essas oito posições de fileira ou coluna. Por exemplo, bin0=1 pode indicar que o coeficiente na fileira ou coluna 0 é o último coeficiente

significativo. Nesse exemplo, se bin0=0, então o coeficiente na posição 0 não é o último coeficiente. Outro binário igual a 1 pode indicar a posição do último coeficiente significativo. Por exemplo, bin1=1 pode indicar que a posição 1 é o último coeficiente significativo. Como outro exemplo, binX=1 pode indicar que a posição X é o último coeficiente significativo. Conforme descrito acima, cada binário pode ser codificado mediante dois métodos diferentes: (1) codificar o binário com um contexto e (2) codificar o binário usando o modo de desvio (sem um contexto).

[066] A Tabela 1 mostra uma binarização exemplar de uma posição de um último coeficiente significativo onde alguns binários são codificados com um contexto e outros são codificados utilizando um modo de desvio. O exemplo na Tabela 1 proporciona uma binarização exemplar do último coeficiente significativo para uma TU 32x32. A segunda coluna da Tabela 1 proporciona valores de prefixo unário truncado correspondente para possíveis valores da posição de um último coeficiente significativo dentro de uma TU de tamanho T definido pelo comprimento de prefixo unário truncado máximo de $2log_2(T)$-1. A terceira coluna da Tabela 1 provê um sufixo de comprimento fixo correspondente para cada prefixo unário truncado. Com a finalidade de brevidade, a Tabela 1 inclui X valores que indicam um valor de bit de um ou zero. Deve-se observar que os valores X mapeiam unicamente cada valor compartilhando um prefixo unário truncado de acordo com um código de comprimento fixo. A magnitude do componente de última posição na Tabela 1 pode corresponder a um valor de coordenada x e/ou um valor de coordenada y. Deve-se observar que a binarização do último coeficiente significativo para uma TU 4x4, 8x8 e 16x16 pode ser definido de uma maneira similar à binarização de uma TU 32x32 descrita com relação à Tabela 1.

Tabela 1. Binarização para uma TU de tamanho 32x32, onde X significa 1 ou 0.

Magnitude do último componente de	Unário truncado (modelo de contexto)	Binário fixo (desvio)	*f ... value*

posição			
0	1	-	0
1	01	-	0
2	001	-	0
3	0001	-	0
4-5	00001	X	0-1
6-7	000001	X	0-1
8-11	0000001	XX	0-3
12-15	00000001	XX	0-3
16-23	000000001	XXX	0-7
24-31	000000000	XXX	0-7

[067] Conforme descrito acima, os símbolos de dados de codificação utilizando CABAC também podem envolver atribuição de contexto. Em um exemplo, a modelagem de contexto pode ser usada para codificação aritmética da porção de sequência unária truncada da sequência binária, enquanto que a modelagem de contexto não é usada para codificação aritmética das sequências binárias fixas da sequência binária (isto é, a sequência binária fixa é ignorada). No caso onde sequências unárias truncadas são codificadas utilizando modelagem de contexto, um contexto pode ser atribuído a cada um dos índices binários de uma sequência binária.

[068] Há várias formas nas quais os contextos podem ser atribuídos a cada índice binário de uma sequência binária. O número de atribuições de contexto para uma sequência binária representando a posição do último coeficiente significativo pode ser igual ao número de índices binários ou comprimento de uma sequência unária truncada para possíveis tamanhos de TU e componentes de cor. Por exemplo, se os possíveis tamanhos de um componente luma são 4x4, 8x8, 16x16 e 32x32, o número de atribuições de contexto para uma dimensão pode ser igual a 60 (isto é, 4+8+16+32), quando nenhum dos binários for ignorado. Similarmente, para cada componente croma com tamanhos possíveis de 4x4, 8x8 e 16x16, o número de atribuições de contexto pode ser igual a 28 (isto é, 4+8+16), quando nenhum dos binários for ignorado. Assim, um número máximo de atribuições de contexto pode ser igual a 116 (isto é, 60+28+28) para cada dimensão quando a posição do último

coeficiente significativo é especificada utilizando a coordenada x e também a coordenada y. As atribuições de contexto exemplares abaixo supõem que alguns binários serão ignorados de acordo com o esquema de binarização descrito com relação à Tabela 1. Contudo, deve-se observar que as técnicas de atribuição de contexto aqui descritas podem ser aplicáveis aos vários esquemas de binarização. Adicionalmente, mesmo quando for suposto que alguns binários serão ignorados, ainda há várias formas nas quais os contextos podem ser atribuídos aos binários de uma sequência binária representando a posição do último coeficiente significativo.

[069] Em alguns casos pode ser desejável reduzir o número total de contextos relativos ao número exigido de atribuições de contexto. Dessa maneira, um codificador ou decodificador pode não precisar armazenar e manter tantos contextos. Contudo, quando o número de contextos é reduzido, a exatidão da predição também pode ser reduzida, por exemplo, se os contextos forem compartilhados para dois binários com probabilidades diferentes. Além disso, um contexto específico pode ser atualizado mais frequentemente, o que pode afetar a probabilidade estimada de um valor binário. Isto é, a codificação de um binário utilizando um contexto atribuído pode envolver atualização de um contexto. Assim, um binário subsequente atribuído ao contexto pode ser codificado utilizando o contexto atualizado. Adicionalmente, deve-se observar que em alguns exemplos os modelos de contexto podem ser inicializados para um nível de fatia, de tal modo que os valores dos binários dentro de uma fatia podem não afetar a codificação dos binários dentro de uma fatia subsequente, embora os binários sejam atribuídos ao mesmo contexto. A revelação descreve técnicas para otimizar as atribuições de contexto, de tal modo que um número de contextos pode ser reduzido enquanto mantendo a exatidão para probabilidades estimadas. Em um exemplo, as técnicas de atribuição de contexto descritas aqui incluem técnicas onde índices de binários individuais são atribuídos o mesmo contexto.

[070] As Tabelas 2-13 abaixo ilustram atribuições de contexto para os índices de binário de uma sequência binária representando a posição de um último coeficiente significativo dentro de uma TU. Deve se observar que alguns binários nas Tabelas 2-13 (por exemplo, binários 5-17 de um bloco 8x8 não há contextos atribuídos. Isso porque se supõe que esses binários serão codificados utilizando-se um modo de desvio, como descrito acima. Deve também ser observado que os valores na Tabela 2-13 representam um índice de um contexto. Nas Tabelas 2-13 quando diferentes binários têm o mesmo valor de índice de contexto eles compartilham o mesmo contexto. O mapeamento de um índice de contexto para um contexto efetivo pode ser definido de acordo com um padrão de codificação de vídeo. As Tabelas 2-13 ilustram como um contexto pode ser comumente atribuído aos binários.

[071] A Tabela 2 ilustra uma possível indexação de contexto para cada binário de diferentes tamanhos de TU para as binarizações exemplares, providas com relação à Tabela 1 acima. No exemplo na Tabela 2, binários adjacentes podem compartilhar os mesmos contextos. Por exemplo, os binários 2 e 3 de uma TU 8x8 compartilham o mesmo contexto.

Tabela 2: Atribuição de contexto para codificação de última posição

Índice Binário	0	1	2	3	4	5	6	7	8
TU 4x4	0	1	2						
TU 8x8	3	4	5	5	6				
TU 16x16	7	8	9	9	10	10	11		
TU 32x32	12	13	14	14	15	15	16	16	17

[072] Cada uma das Tabelas 3-6 ilustra exemplos adicionais de atribuições de contexto de acordo com as seguintes regras:

1. Os primeiros K binários não estão compartilhando contexto, onde K>1. K poderia ser diferente para cada tamanho de TU.

2. A um contexto somente podem ser atribuídos binários consecutivos. Por

exemplo, binário 3 - binário 5 poderiam utilizar contexto 5. Porém, binário 3 e binário 5, utilizando contexto 5 e binário 4 utilizando contexto 6 não são permitidos.

3. O último N binário, N>=0, de tamanhos de TU diferentes pode compartilhar o mesmo contexto.

4. O número de binários que compartilham o mesmo contexto aumenta com tamanhos de TU.

[073] As regras 1-4 acima podem ser particularmente úteis para a binarização provida na Tabela 1. Contudo, as regras 1-4 podem ser igualmente úteis para outros esquemas de binarização e as atribuições de contexto efetivas podem ser ajustadas de acordo com o esquema de binarização que é implementado.

Tabela 3: Exemplo de binários de última posição de acordo com as Regras 1-4

Índice Binário	0	1	2	3	4	5	6	7	8
TU 4x4	0	1	2						
TU 8x8	3	4	5	6	7				
TU 16x16	8	9	10	11	11	12	12		
TU 32x32	13	14	14	15	16	16	16	16	17

Tabela 4: Exemplo de binários de última posição de acordo com as Regras 1-4

Índice Binário	0	1	2	3	4	5	6	7	8
TU 4x4	0	1	2						
TU 8x8	3	4	5	6	6				
TU 16x16	8	9	10	11	11	12	12		
TU 32x32	13	14	14	15	16	16	16	16	17

Tabela 5: Exemplo de binários de última posição de acordo com as Regras 1-4

Índice Binário	0	1	2	3	4	5	6	7	8
TU 4x4	0	1	2						

TU 8x8	3	4	5	6	7				
TU 16x16	8	9	10	11	11	12	12		
TU 32x32	13	14	14	15	16	16	16	12	12

Tabela 6: Exemplo de binários de última posição de acordo com as Regras 1-4

Índice Binário	0	1	2	3	4	5	6	7	8
TU 4x4	0	1	2						
TU 8x8	3	4	5	6	7				
TU 16x16	8	9	10	10	11	11	12		
TU 32x32	13	14	14	15	15	15	16	16	16

[074] As Tabelas 7-8 abaixo proporcionam atribuições de contexto exemplares onde os últimos binários a partir de diferentes tamanhos de blocos compartilham o mesmo contexto, o que pode otimizar adicionalmente o número de contextos. Em um exemplo, mapeamento direto pode ser usado para determinar como os conceitos são compartilhados entre os últimos binários de dois ou mais tamanhos de bloco. Por exemplo, para um bloco A e para um bloco B com tamanhos M e N, respectivamente, o contexto do enésimo binário do bloco A pode usar o mesmo contexto que o enésimo binário do bloco B.

Tabela 7: Exemplo de binários de última posição onde os tamanhos de bloco compartilham os mesmos contextos

Índice Binário	0	1	2	3	4	5	6	7	8
TU 4x4	12	13	14						
TU 8x8	12	13	14	14	15				
TU 16x16	12	13	14	14	15	15	16		
TU 32x32	12	13	14	14	15	15	16	16	17

[075] A Tabela 8 mostra outro exemplo onde os binários de última posição a partir de alguns tamanhos de bloco compartilham os contextos entre si. Nesse caso, as TUs de tamanho 8x8 e 16x16 compartilham os mesmos contextos.

Tabela 8: Exemplo de binários de última posição onde os tamanhos de bloco (8x8 e 16x16) compartilham os mesmos contextos

Índice Binário	0	1	2	3	4	5	6	7	8
TU 4x4	0	1	2						
TU 8x8	7	8	9	9	10				
TU 16x16	7	8	9	9	10	10	11		
TU 32x32	12	13	14	14	15	15	16	16	17

[076] Em outro exemplo, o mapeamento de contextos para binários de última posição a partir de diferentes tamanhos de blocos podem ser derivados utilizando uma função f(.). Por exemplo, o enésimo binário no tamanho de bloco A pode compartilhar os mesmos contextos com o binário de ordem m no tamanho de bloco B, onde m é uma função de (m= f(n)). Por exemplo, a função pode ser linear, isto é, m = n*a + b, onde a e b são parâmetros da função linear. A Tabela 9 mostra um exemplo onde a = 1, b = 1, A = uma TU 8x8 e B = uma TU 16x16.

Tabela 9: Exemplo de binários de última posição com contextos compartilhados com base em uma função linear

Índice Binário	0	1	2	3	4	5	6	7	8
TU 4x4	0	1	2						
TU 8x8	8	9	9	10	10				
TU 16x16	7	8	9	9	10	10	11		
TU 32x32	12	13	14	14	15	15	16	16	17

[077] Deve-se observar que ao aplicar a equação acima em certos casos, devido à operação de número inteiro, pode haver arredondamento envolvido. Por exemplo, 7*0,5 = 3.

[078] De acordo com exemplo seguinte, o mapeamento a partir da posição n no tamanho de bloco 8x8 para um local m em um bloco 4x4 pode ser calculado com a seguinte equação:

$$m = f(n) = n>>1, \text{ que significa } a = 0,5,\ b = 0,\ A=8x8,\ B=4x4 \qquad (1)$$

[079] O mapeamento a partir do local n em um bloco 16x16 para um local m em um bloco 4x4 pode ser calculado com a seguinte equação:

m = f(n) = n>>2, que significa a = 0,5, b = 0, A=16x16, B=4x4 (2)

[080] Conforme descrito acima, as Equações (1) e (2) são apenas dois exemplos que podem ser usados para implementar um mapeamento entre blocos de tamanhos diferentes. As Equações (1) e (2) podem ser referidas como funções de mapeamento. Deve-se observar que o “>>” nas Equações (1) e (2) pode representar uma operação de deslocamento definida de acordo com um padrão de codificação de vídeo, tal como HEVC. Adicionalmente, outras equações podem ser usadas para se obter o mesmo mapeamento ou mapeamentos diferentes.

[081] A Tabela 10 provê uma atribuição de contexto exemplar para o último coeficiente significativo para as TUs 4x4, 8x8 e 16x16 de acordo com as Equações (1) e (2).

Tabela 10: Mapeamento de contexto para unidades de transformada de tamanho diferente

Índice Binário	0	1	2	3	4	5	6	7	8
TU 4x4	0	1	2						
TU 8x8	0	0	1	1	2				
TU 16x16	0	0	0	0	1	1	1		

[082] A Tabela 11 proporciona um exemplo da atribuição de contexto da Tabela 10 onde diferentes valores de índice de contexto são usados (isto é, 15-17 em vez de 0-2). Conforme descrito acima, os valores dos índices de contexto nas Tabelas 3-12 não pretendem limitar os contextos efetivos atribuídos a um índice binário.

Tabela 11: Mapeamento de contexto para unidades de transformada de tamanho diferente

Índice Binário	0	1	2	3	4	5	6	7	8

TU 4x4	15	16	17						
TU 8x8	15	15	16	16	17				
TU 16x16	15	15	15	15	16	16	16		

[083] Deve-se observar que o mapeamento dos contextos na Tabela 11 é equivalente à função de mapeamento seguinte:

ctx_index = (n >>k) + 15 (3)

onde ctx_index é o índice do contexto;

n = Índice binário

k = log2TrafoDimension-2;

log2TrafoDimension = log2 (largura) para última posição na dimensão x;

log2TrafoDimension = log2 (altura) para última posição na dimensão y.

[084] Em alguns casos a função definida em (1)-(3) pode ser usada por um dispositivo de codificação para elaborar uma série de tabelas que podem ser armazenadas em memória e utilizadas para atribuições de contexto de consulta. Em alguns casos, as tabelas podem ser predeterminadas com base nas equações e regras aqui descritas e armazenadas tanto no codificador de vídeo 20 como no decodificador de vídeo 30.

[085] Adicionalmente, em alguns exemplos, as funções (1)-(3) definidas acima podem ser aplicadas seletivamente para atribuir contextos para binários específicos. Dessa maneira, a diferentes binários podem ser atribuídos um contexto com base em regra diferente. Em um exemplo, as funções, tais como aquelas descritas acima, podem se aplicar apenas para um índice binário (isto é, um valor de n) que é menor do que um limiar Th1, e/ou maior do que um limiar Th2. A Tabela 12 mostra um exemplo onde as técnicas de mapeamento descritas acima são aplicadas seletivamente com base no valor do índice binário, isto é, n > Th2 = 2.

Tabela 12: Exemplo de binários de última posição com contextos compartilhados com base em uma função linear e um limiar

Índice Binário	0	1	2	3	4	5	6	7	8

TU 4x4	0	1	2						
TU 8x8	3	4	9	10	10				
TU 16x16	7	8	9	9	10	10	11		
TU 32x32	12	13	14	14	15	15	16	16	17

[086] Em outro exemplo, o valor de limiar para aplicar as técnicas aos índices binários pode ser diferente para os tamanhos diferentes de bloco, diferentes tipos de quadro, diferentes componentes de cor (Y,U,V), e/ou outra informação secundária. Esse limiar pode ser predefinido de acordo com um padrão de codificação de vídeo ou pode ser sinalizado utilizando sintaxe de alto nível. Por exemplo, o limiar pode ser sinalizado em um conjunto de parâmetros de sequência (SPS), um conjunto de parâmetros de imagem (PPS), um conjunto de parâmetros de adaptação (APS), e/ou um cabeçalho de fatia.

[087] Em outro exemplo, uma função de mapeamento pode ser diferente para os tamanhos diferentes de bloco, os diferentes tipos de quadro, os diferentes componentes de cor (Y, U e V), e/ou outra informação secundária. A função de mapeamento pode ser predefinida de acordo com um padrão de codificação de vídeo ou pode ser sinalizada utilizando sintaxe de alto nível. Por exemplo, a função de mapeamento pode ser sinalizada em um SPS, um PPS, um APS e/ou um cabeçalho de fatia.

[088] Em outro exemplo, as técnicas de mapeamento de função e mapeamento direto descritas acima podem ser aplicadas adaptativamente com base em componentes de cor, tipo de quadro, parâmetros de quantização (QP) e/ou outra informação secundária. Por exemplo, as técnicas de mapeamento de função ou de mapeamento direto podem ser empregadas apenas aos componentes croma. As regras para essa capacidade de adaptação podem ser predefinidas ou podem ser sinalizadas utilizando sintaxe de alto nível. Por exemplo, as regras para capacidade de adaptação podem sinalizadas em um SPS, um PPS, um APS e/ou um cabeçalho de fatia.

[089] Em outro exemplo, binários de última posição para componentes croma e luma podem compartilhar os mesmos conceitos. Isso pode se aplicar para qualquer tamanho de bloco, por exemplo, 4x4, 8x8, 16x16 ou 32x32. A Tabela 13 mostra um exemplo onde os contextos são compartilhados para binários de última posição de componentes luma e croma para uma TU 4x4.

Tabela 13: Exemplos de binários de última posição para componentes luma e croma em uma TU 4x4

Índice Binário	0	1	2	3
Luma TU 4x4	0	1	2	
Chroma TU 4x4	0	1	2	

[090] A Figura 4 é um diagrama de blocos ilustrando um exemplo de um codificador de vídeo 20 que pode usar as técnicas para codificar coeficientes de transformada conforme descrito nessa revelação. Por exemplo, o codificador de vídeo 20 representa um exemplo de um codificador de vídeo configurado para obter uma sequência binária indicando uma posição de um último coeficiente significativo dentro de um bloco de vídeo; determinar um contexto para um índice binário da sequência binária com base em um tamanho de bloco de vídeo, em que o contexto é atribuído a pelo menos dois índices binários, em que cada um dos pelo menos dois índices binários é associado com diferentes tamanhos de bloco de vídeo; e codificar a sequência binária utilizando CABAC com base ao menos em parte no contexto determinado. O codificador de vídeo 20 será descrito no contexto de codificação HEVC para fins de ilustração, porém sem limitação dessa revelação em relação a outros padrões ou métodos de codificação que podem exigir varredura dos coeficientes de transformada. O codificador de vídeo 20 pode realizar codificação intra e inter das CUs dentro dos quadros de vídeo. Codificação intra se baseia em predição espacial para reduzir ou remover a redundância espacial em dados de vídeo dentro de um determinado quadro de vídeo. Codificação inter se baseia em predição temporal para reduzir ou remover a redundância temporal entre um quadro

atual e quadros previamente codificados de uma sequência de vídeo. Modo intra (modo I) pode se referir a qualquer um dos vários modos de compactação de vídeo de base espacial. Os modos inter tal como predição unidirecional (modo P) ou predição bidirecional (modo B) pode se referir a qualquer um dos vários modos de compactação de vídeo de base temporal.

[091] Conforme mostrado na Figura 4, o codificador de vídeo 20 recebe um bloco de vídeo atual dentro de um quadro de vídeo para ser codificado. No exemplo da Figura 4, o codificador de vídeo 20 inclui um módulo de seleção de modo 40, um módulo de estimação de movimento 42, um módulo de compensação de movimento 44, um módulo de predição intra 46, um armazenador temporário de quadro de referência 64, um somador 50, um módulo de transformada 52, um módulo de quantização 54, e um módulo de codificação de entropia 56. O módulo de transformada 52 ilustrado na Figura 4 é o módulo que aplica a transformada atual ou combinações de transformada a um bloco de dados residuais, e não deve ser confundido com um bloco de coeficientes de transformada, o qual também pode ser referido como uma unidade de transformada (TU) de uma CU. Para reconstrução de bloco de vídeo, o codificador de vídeo 20 também inclui um módulo de quantização inversa 58, um módulo de transformada inversa 60, e um somador 62. Um filtro de desblocagem (não mostrado na Figura 4) também pode ser incluído para filtrar os limites de bloco para remover artefatos de blocagem a partir do vídeo reconstruído. Se desejado, o filtro de desblocagem tipicamente filtraria a saída do somador 62.

[092] Durante o processo de codificação, o codificador de vídeo 20 recebe o quadro de vídeo ou fatia a ser codificada. O quadro ou fatia pode ser dividido em múltiplos blocos de vídeo, por exemplo, unidades de codificação maiores (LCUs). O módulo de estimação de movimento 42 e o módulo de compensação de movimento 44 realizam codificação inter preditiva do bloco de vídeo recebido em relação a um ou mais blocos em um ou mais quadros de referência para prover compactação

temporal. O módulo de predição intra 46 pode realizar codificação preditiva intra do bloco de vídeo recebido em relação a um ou mais blocos adjacentes no mesmo quadro ou fatia que o bloco a ser codificado para proporcionar compactação espacial.

[093] O módulo de seleção de modo 40 pode selecionar um dos modos de codificação, intra ou inter, por exemplo, com base em resultados de erro (isto é, distorção) para cada modo, e fornece o bloco predito intra ou inter resultante (por exemplo, uma unidade de predição (PU)) ao somador 50 para gerar dados residuais de bloco e ao somador 62 para reconstruir o bloco codificado para uso em um quadro de referência. O somador 62 combina o bloco predito com os dados transformados de forma inversa, quantizados de forma inversa a partir do módulo de transformada inversa 60 para o bloco para reconstruir o bloco codificado, conforme descrito em maior detalhe abaixo. Alguns quadros de vídeo podem ser designados como quadros I, onde todos os blocos em um quadro I são codificados em um modo de predição intra. Em alguns casos, o módulo de predição intra 46 pode realizar codificação de predição intra de um bloco em um quadro P ou B, por exemplo, quando a busca de movimento realizada pelo módulo de estimação de movimento 42 não resulta em uma predição suficiente do bloco.

[094] O módulo de estimação de movimento 42 e o módulo de compensação de movimento 44 podem ser altamente integrados, mas são ilustrados separadamente para fins conceptuais. A estimação de movimento (ou busca de movimento) é o processo de gerar vetores de movimento, os quais estimam o movimento para os blocos de vídeo. Um vetor de movimento, por exemplo, pode indicar o deslocamento de uma unidade de predição em um quadro atual em relação a uma amostra de referência de um quadro de referência. O módulo de estimação de movimento 42 calcula um vetor de movimento para uma unidade de predição de um quadro codificado inter, mediante comparação da unidade de predição com as

amostras de referência de um quadro de referência armazenado no armazenador temporário de quadro de referência 64. Uma amostra de referência pode ser um bloco que é encontrado para combinar estreitamente com a porção da CU incluindo a PU sendo codificada em termos de diferença de pixel, o que pode ser determinado pela soma da diferença absoluta (SAD), soma das diferenças levadas ao quadro (SSD), ou outra métrica de diferença. A amostra de referência pode ocorrer em qualquer lugar dentro de um quadro de referência ou fatia de referência, e não necessariamente em um limite de bloco (por exemplo, unidade de codificação) do quadro ou fatia de referência. Em alguns exemplos, a amostra de referência pode ocorrer em uma posição de pixel fracionário.

[095] O módulo de estimação de movimento 42 envia o vetor de movimento calculado para o módulo de codificação de entropia 56 e para o módulo de compensação de movimento 44. A porção do quadro de referência identificada por um vetor de movimento pode ser referida como uma amostra de referência. O módulo de compensação de movimento 44 pode calcular um valor de predição para uma unidade de predição de uma CU atual, por exemplo, mediante recuperação da amostra de referência identificada por um vetor de movimento para a PU.

[096] O módulo de predição intra 46 pode intra predizer o bloco recebido, como uma alternativa para a predição inter realizada pelo módulo de estimação de movimento 42 e pelo módulo de compensação de movimento 44. O módulo de predição intra 46 pode predizer o bloco recebido em relação aos blocos adjacentes, previamente codificados, por exemplo, os blocos acima, acima e à direita, acima e à esquerda, ou à esquerda do bloco atual, supondo uma ordem de codificação da esquerda para a direita, de cima para baixo, para os blocos. O módulo de predição intra 46 pode ser configurado com uma variedade de diferentes modos de predição intra. Por exemplo, o módulo de predição intra 46 pode ser configurado com certo número de modos de predição direcional, por exemplo, 34 modos de predição

direcional, com base no tamanho da CU sendo codificada.

[097] O módulo de predição intra 46 pode selecionar um modo de predição intra mediante, por exemplo, cálculo dos valores de erro para os vários módulos de predição intra e selecionando um modo que produza o valor de erro mais baixo. Os modos de predição direcional podem incluir funções para combinar valores de pixels espacialmente vizinhos e aplicar os valores combinados a uma ou mais posições de pixel em uma PU. Quando os valores para todas as posições de pixel na PU tiverem sido calculados, o módulo de predição intra 46 pode calcular um valor de erro para o modo de predição com base nas diferenças de pixel entre a PU e o bloco recebido a ser codificado. O módulo de predição intra 46 pode continuar a testar os modos de predição intra até que um modo de predição intra que produza um valor de erro aceitável seja descoberto. O módulo de predição intra 46 pode então enviar a PU para o somador 50.

[098] O codificador de vídeo 20 forma um bloco residual mediante subtração dos dados de predição calculados pelo módulo de compensação de movimento 44 ou módulo de predição intra 46 a partir do bloco de vídeo original sendo codificado. O somador 50 representa o componente ou componentes que realizam essa operação de subtração. O bloco residual pode corresponder a uma matriz bidimensional de valores de diferença de pixel, onde o número de valores no bloco residual é o mesmo que o número de pixels na PU correspondendo ao bloco residual. Os valores no bloco residual podem corresponder às diferenças, por exemplo, erro, entre valores de pixels co-localizados na PU e no bloco original a ser codificado. As diferenças podem ser diferenças croma ou luma dependendo do tipo de bloco que seja codificado.

[099] O módulo de transformada 52 pode formar uma ou mais unidades de transformada (TUs) a partir do bloco residual. O módulo de transformada 52 seleciona uma transformada dentre uma pluralidade de transformadas. A

transformada pode ser selecionada com base em uma ou mais características de codificação, tal como tamanho de bloco, modo de codificação ou semelhante. O módulo de transformada 52 aplica então a transformada selecionada à TU, produzindo um bloco compreendendo um arranjo bidimensional de coeficientes de transformada.

[0100] O módulo de transformada 52 pode enviar os coeficientes de transformada resultantes para o módulo de quantização 54. O módulo de quantização 54 pode então quantizar os coeficientes de transformada. O módulo de codificação de entropia 56 pode então realizar uma varredura dos coeficientes de transformada quantizados na matriz de acordo com um modo de varredura. Essa revelação descreve o módulo de codificação de entropia 56 como realizando a varredura. Contudo, deve ser entendido que, em outros exemplos, outros módulos de processamento tal como o módulo de quantização 54, poderiam realizar a varredura.

[0101] O módulo de quantização inversa 58 e o módulo de transformada inversa 60 aplicam quantização inversa e transformação inversa, respectivamente, para reconstruir o bloco residual no domínio de pixel, por exemplo, para uso posterior como um bloco de referência. O módulo de compensação de movimento 44 pode calcular um bloco de referência mediante ajuste do bloco residual para um bloco preditivo de um dos quadros do armazenador temporário de quadro de referência 64. O armazenador temporário de quadro de referência 64 algumas vezes é referido como um armazenador temporário de imagem decodificada (DPB). O módulo de compensação de movimento 44 também pode aplicar um ou mais filtros de interpolação ao bloco residual reconstruído para calcular valores de pixel de sub-número inteiro para uso na estimação de movimento. O somador 62 adiciona o bloco residual reconstruído ao bloco de predição de movimento compensado produzido pelo módulo de compensação de movimento 44 para produzir um bloco de vídeo

reconstruído para armazenamento no armazenador temporário de quadro de referência 64. O bloco de vídeo reconstruído pode ser usado pelo módulo de estimação de movimento 42 e pelo módulo de compensação de movimento 44 como um bloco de referência para codificar inter um bloco em um quadro de vídeo subsequente.

[0102] Quando os coeficientes de transformada são submetidos à varredura para o arranjo unidimensional, o módulo de codificação de entropia 56 pode aplicar codificação de entropia tal como CAVLC, CABAC, SBAC, PIPE, ou outra metodologia de codificação de entropia aos coeficientes. Em alguns casos, o módulo de codificação de entropia 56 pode ser configurado para realizar outras funções de codificação, além da codificação de entropia. Por exemplo, o módulo de codificação de entropia 56 pode ser configurado para determinar valores de padrão de bloco codificado (CBP) para as CU e PU. Além disso, em alguns casos, o módulo de codificação de entropia 56 pode realizar codificação de extensão de execução dos coeficientes. Após a codificação de entropia pelo módulo de codificação de entropia 56, o vídeo codificado resultante pode ser transmitido a outro dispositivo, tal como o decodificador de vídeo 30, ou arquivado para posterior transmissão ou recuperação.

[0103] De acordo com as técnicas dessa revelação, o módulo de codificação de entropia 56 pode selecionar o contexto usado para codificar elementos de sintaxe com base, por exemplo, nas atribuições de contexto descritas acima com relação à Tabela 2-13 e qualquer combinação do seguinte: uma direção de predição intra para os modos de predição intra, uma posição de varredura do coeficiente correspondendo aos elementos de sintaxe, tipo de bloco, tipo de transformada, e/ou outras propriedades de sequência de vídeo.

[0104] Em um exemplo, o módulo de codificação de entropia 56 pode codificar a posição de um último coeficiente significativo utilizando a técnica de binarização adotada em HEVC descrito acima com relação à Tabela 1. Em outros

exemplos, o módulo de codificação de entropia 56 pode codificar a posição de um último coeficiente significativo utilizando outras técnicas de binarização. Em um exemplo, uma palavra-chave para a posição de um último coeficiente significativo pode incluir um prefixo de código unário truncado seguido por um sufixo de código de comprimento fixo. Em um exemplo, cada magnitude da última posição pode usar a mesma binarização para todos os possíveis tamanhos de TU, exceto quando a última posição é igual ao tamanho de TU menos 1. Essa exceção se deve às propriedades da codificação unária truncada. Em um exemplo, a posição de um último coeficiente significativo dentro de um coeficiente de transformada retangular pode ser especificado mediante especificação de valor de coordenada x e um valor de coordenada y. Em outro exemplo, um bloco de coeficiente de transformada pode estar na forma de um vetor 1xN e a posição do último coeficiente significativo dentro do vetor pode ser especificada por um único valor de posição.

[0105] A Figura 5 é um diagrama de blocos que ilustra um módulo de codificação de entropia exemplar que poderia implementar as técnicas descritas nessa revelação. Em um exemplo, o módulo de codificação de entropia 56 ilustrado na Figura 5 pode ser um codificador CABAC. O módulo de codificação de entropia exemplar 56 pode incluir um módulo de binarização 502, um módulo de codificação aritmética 510, o qual inclui um mecanismo de codificação de desvio 504 e um mecanismo de codificação regular 508, e um módulo de modelagem de contexto 506. O módulo de codificação de entropia 56 recebe elementos de sintaxe, tal como um ou mais elementos de sintaxe representando a posição do último coeficiente de transformada significativo dentro de coeficientes de bloco de transformada e codifica o elemento de sintaxe em um fluxo de bits. Os elementos de sintaxe podem incluir um elemento de sintaxe especificando uma coordenada x da posição de um último coeficiente significativo dentro de um bloco de coeficiente de transformada e um elemento de sintaxe especificando uma coordenada y da posição do último

coeficiente significativo dentro de um bloco de coeficiente de transformada.

[0106] O módulo de binarização 502 recebe um elemento de sintaxe e produz uma sequência de binários (isto é, sequência binária). Em um exemplo, o módulo de binarização 502 recebe os elementos de sintaxe representando a última posição de um coeficiente significativo dentro de um bloco de coeficientes de transformada e produz uma sequência de binários de acordo com o exemplo descrito acima com relação à Figura 1. O módulo de codificação aritmética 510 recebe uma sequência binária a partir do módulo de binarização 502 e realiza a codificação aritmética na sequência binária. Conforme mostrado na Figura 5, um módulo de codificação aritmética 510 pode receber valores binários a partir de um percurso de desvio ou do percurso de codificação regular. Consistente com o processo CABAC descrito acima, no caso onde o módulo de codificação aritmética 510 recebe os valores binários a partir de um percurso de desvio, o mecanismo de codificação de desvio 504 pode realizar a codificação aritmética nos valores binários sem utilizar um contexto designado para um valor binário. Em um exemplo, o mecanismo de codificação de desvio 504 pode assumir probabilidades iguais para possíveis valores de um binário.

[0107] No caso onde o módulo de codificação aritmética 510 recebe os valores binários através do percurso regular, o módulo de modelagem de contexto 506 pode prover uma variável de contexto (por exemplo, um estado de contexto), de tal modo que o mecanismo de codificação regular 508 pode realizar a codificação aritmética com base nas atribuições de contexto providas pelo módulo de modelagem de contexto 506. Em um exemplo, o módulo de codificação aritmética 510 pode codificar uma porção de prefixo de uma sequência de bits utilizando atribuição de contexto e pode codificar uma porção de sufixo de uma sequência de bits sem utilizar atribuições de contexto. As atribuições de contexto podem ser definidas de acordo com os exemplos descritos acima com relação às Tabelas 2-13.

Os modelos de contexto podem ser armazenados em memória. O módulo de modelagem de contexto 506 pode incluir uma série de tabelas indexadas e/ou utilizar funções de mapeamento para determinar um contexto e uma variável de contexto para um binário específico. Após codificar um valor binário, o mecanismo de codificação regular 508, pode atualizar um contexto com base nos valores binários efetivos e produzir o valor binário codificado como parte de um fluxo de bits. Dessa maneira, o módulo de codificação de entropia é configurado para codificar um ou mais elementos de sintaxe com base nas técnicas de atribuição de contexto aqui descritas.

[0108] A Figura 6 é um fluxograma ilustrando um método exemplar para determinar um contexto para um valor de sequência binária indicando a posição de um último coeficiente significativo de acordo com as técnicas dessa revelação. O método descrito na Figura 6 pode ser realizado mediante qualquer um dos codificadores de vídeo exemplares ou codificadores de entropia descritos aqui. Na etapa 602, uma sequência binária indicando a posição de um último coeficiente de transformada significativo dentro de um bloco de vídeo é obtida. A sequência binária pode ser definida de acordo com o esquema de binarização descrito com relação à Tabela 1. Na etapa 604, um contexto é determinado para um valor binário da sequência binária. Um contexto pode ser atribuído a um binário com base nas técnicas aqui descritas. O contexto pode ser determinado por um codificador de entropia ou vídeo acessando uma tabela de consulta ou realizando a função de mapeamento. O contexto pode ser usado para derivar uma variável de contexto específica para um binário específico. Uma variável de contexto pode ser um valor binário de 7 bits indicando uma de 64 possíveis probabilidades (estados) e um estado de maior probabilidade (por exemplo, “1” ou “0”). Conforme descrito acima, em alguns casos, os binários podem compartilhar contextos de acordo com as funções de mapeamento e Tabelas 2-13 descritas acima. Na etapa 606, um valor

binário é codificado utilizando um processo de codificação aritmética que utiliza uma variável de contexto, tal como CABAC. Deve-se observar que quando os binários compartilham contextos, o valor de um binário pode afetar o valor de uma variável de contexto usada para codificar um binário subsequente de acordo com técnicas de codificação de contexto adaptativo. Por exemplo, se um binário específico for “1” um binário subsequente pode ser codificado com base em uma probabilidade aumentada de ser 1. Dessa maneira, codificação de entropia da sequência binária pode incluir atualizar um estado de contexto de um modelo de contexto. Adicionalmente, deve-se observar que em alguns exemplos os modelos de contexto podem ser inicializados para um nível de fatia, de tal modo que os valores dos binários dentro da fatia podem não afetar a codificação dos binários dentro de uma fatia subsequente.

[0109] A Figura 7 é um diagrama de blocos ilustrando um exemplo de um decodificador de vídeo 30 que pode usar técnicas para codificar os coeficientes de transformada como descrito nessa revelação. Por exemplo, o decodificador de vídeo 30 representa um exemplo de um decodificador de vídeo configurado para obter uma sequência binária codificada indicando uma posição de um último coeficiente significativo dentro de um bloco de vídeo, em que a sequência binária codificada é codificada utilizando CABAC; determinar um contexto para um índice binário da sequência binária codificada com base em um tamanho de bloco de vídeo, em que o contexto é atribuído a pelo menos dois índices binários, em que cada um dos pelo menos dois índices binários é associado com diferentes tamanhos de bloco de vídeo; e decodificar a sequência binária codificada utilizando CABAC com base ao menos em parte no contexto determinado.

[0110] No exemplo da Figura 7, o decodificador de vídeo 30 inclui um módulo de decodificação de entropia 70, um módulo de compensação de movimento 72, um módulo de predição intra 74, um módulo de quantização inversa 76, um

módulo de transformada inversa 78, um armazenador temporário de quadro de referência 82 e um somador 80. O decodificador de vídeo 30 pode, em alguns exemplos, realizar uma passagem de decodificação geralmente recíproca em relação à passagem de codificação descrita com relação ao codificador de vídeo 20.

[0111] O módulo de decodificação de entropia 70 realiza um processo de decodificação de entropia no fluxo de bits codificado para recuperar um arranjo unidimensional de coeficientes de transformada. O processo de decodificação de entropia utilizado depende da codificação de entropia utilizada pelo codificador de vídeo 20 (por exemplo, CABAC, CAVLC, etc.). O processo de codificação de entropia usado pelo codificador pode ser sinalizado no fluxo de bits codificado ou pode ser um processo predeterminado.

[0112] Em alguns exemplos, o módulo de decodificação de entropia 70 (ou o módulo de quantização inversa 76) pode realizar a varredura dos valores recebidos utilizando uma varredura espelhando o modo de varredura usado pelo módulo de codificação de entropia 56 (ou o módulo de quantização 54) do codificador de vídeo 20. Embora a varredura dos coeficientes possa ser realizada no módulo de quantização inversa 76, a varredura será descrita com propósitos de ilustração como sendo realizada pelo módulo de decodificação de entropia 70. Além disso, embora mostrado como módulos funcionais separados para facilidade de ilustração, a estrutura e a funcionalidade do módulo de decodificação de entropia 70, do módulo de quantização inversa 76, e de outros módulos do decodificador de vídeo 30 podem ser altamente integradas entre si.

[0113] O módulo de quantização inversa 76 quantiza de forma inversa, isto é, desquantiza, os coeficientes de transformada quantizados providos no fluxo de bits e decodificados pelo módulo de decodificação de entropia 70. O processo de quantização inversa pode incluir um processo convencional, por exemplo, similar aos processos propostos para HEVC ou definidos pelo padrão de decodificação H.264.

O processo de quantização inversa pode incluir o uso de um parâmetro de quantização QP calculado pelo codificador de vídeo 20 para a CU para determinar um grau de quantização e, similarmente, um grau de quantização inversa que deve ser aplicado. O módulo de quantização inversa 76 pode quantizar de forma inversa os coeficientes de transformada quer seja antes ou após os coeficientes serem convertidos a partir de um arranjo unidimensional para um arranjo bidimensional.

[0114] O módulo de transformada inversa 78 aplica uma transformada inversa aos coeficientes de transformada quantizada inversa. Em alguns exemplos, o módulo de transformada inversa 78 pode determinar uma transformada inversa com base na sinalização a partir do codificador de vídeo 20, ou mediante dedução da transformada a partir de uma ou mais características de codificação tal como tamanho de bloco, modo de codificação ou semelhante. Em alguns exemplos, o módulo de transformada inversa 78 pode determinar uma transformada para aplicar ao bloco atual com base em uma transformada sinalizada no nó raiz de uma quadtree para uma LCU incluindo o bloco atual. Alternativamente, a transformada pode ser sinalizada na raiz de uma quadtree de TU para uma CU de nó folha na quadtree LCU. Em alguns exemplos, o módulo de transformada inversa 78 pode aplicar uma transformada inversa em cascata, na qual o módulo de transformada inversa 78 aplica duas ou mais transformadas inversas aos coeficientes de transformada do bloco atual sendo decodificado.

[0115] O módulo de predição intra 74 pode gerar os dados de predição para um bloco atual de um quadro atual com base em um modo de predição intra sinalizado e dados a partir dos blocos previamente decodificados do quadro atual. O módulo de compensação de movimento 72 pode recuperar o vetor de movimento, direção de predição de movimento e índice de referência a partir do fluxo de bits codificado. A direção de predição de referência indica se o modo de predição inter é unidirecional (por exemplo, um quadro P) ou bidirecional (um quadro B). O índice de

referência indica em qual quadro de referência o vetor de movimento candidato se baseia. Com base na direção de predição de movimento recuperada, o índice de quadro de referência, e vetor de movimento, o módulo de compensação de movimento 72 produz um bloco de movimento compensado para a porção atual. Esses blocos de movimento compensado essencialmente recriam o bloco preditivo usado para produzir os dados residuais.

[0116] O módulo de compensação de movimento 72 pode produzir os blocos de movimento compensado, possivelmente realizando interpolação com base nos filtros de interpolação. Identificadores para os filtros de interpolação a serem usados para estimação de movimento com precisão de subpixel podem ser incluídos nos elementos de sintaxe. O módulo de compensação de movimento 72 pode usar filtros de interpolação conforme utilizados pelo codificador de vídeo 20 durante a codificação do bloco de vídeo para calcular os valores interpolados para pixels de sub-inteiros de um bloco de referência. O módulo de compensação de movimento 72 pode determinar os filtros de interpolação usados pelo codificador de vídeo 20 de acordo com a informação de sintaxe recebida e utilizar os filtros de interpolação para produzir blocos preditivos.

[0117] Adicionalmente, o módulo de compensação de movimento 72 e o módulo de predição intra 74, em um exemplo HEVC, podem utilizar alguma da informação de sintaxe (por exemplo, provida por uma quadtree) para determinar os tamanhos das LCUs usadas para codificar quadro(s) da sequência de vídeo codificado. O módulo de compensação de movimento 72 e o módulo de predição intra 74 também podem utilizar informação de sintaxe para determinar informação dividida que descreve como cada CU de um quadro da sequência de vídeo codificado é dividido (e similarmente, como as sub-CUs são divididas). A informação de sintaxe também pode incluir modos indicando como cada divisão é codificada (por exemplo, predição intra ou inter, e para predição intra um modo de codificação

de predição intra), um ou mais quadros de referência (e/ou listas de referência contendo identificadores para os quadros de referência) para cada PU codificada inter, e outra informação para decodificar a sequência de vídeo codificada.

[0118] O somador 80 combina os blocos residuais com os blocos de predição correspondentes gerados pelo módulo de compensação de movimento 72 ou pelo módulo de predição intra 74 para formar os blocos decodificados. Se desejado, o um filtro de desblocagem também pode ser aplicado para filtrar os blocos decodificados para remover os artefatos de blocagem. Os blocos de vídeo decodificados são então armazenados no armazenador de quadro de referência 82 o qual proporciona blocos de referência para compensação de movimento subsequente e também produz vídeo decodificado para apresentação em um dispositivo de exibição (tal como o dispositivo de exibição 32 da Figura 1). O armazenador temporário de quadro de referência 82 também pode ser referido como um DPB.

[0119] A Figura 8 é um diagrama de blocos que ilustra um módulo de decodificação de entropia exemplar 70 que pode implementar as técnicas descritas nessa revelação. O módulo de decodificação de entropia 70 recebe um fluxo de bits codificado por entropia e decodifica os elementos de sintaxe a partir do fluxo de bits. Em um exemplo, os elementos de sintaxe podem representar a posição do último coeficiente de transformada significativo dentro de coeficientes de bloco de transformada. Os elementos de sintaxe podem incluir um elemento de sintaxe especificando uma coordenada x da posição de um último coeficiente significativo dentro de um bloco de coeficiente de transformada e um elemento de sintaxe especificando uma coordenada y da posição de um último coeficiente significativo dentro de um bloco de coeficiente de transformada. Em um exemplo, o módulo de decodificação de entropia 70 ilustrado na Figura 8 pode ser um decodificador CABAC. O módulo de decodificação de entropia exemplar 70 na Figura 8 inclui um

módulo de decodificação aritmética 702, o qual pode incluir um mecanismo de decodificação de desvio 704 e um mecanismo de decodificação regular 706. O módulo de decodificação de entropia exemplar 70 inclui também unidade de modelagem de contexto 708 e módulo de binarização inversa 710. O módulo de decodificação de entropia exemplar 70 pode realizar as funções recíprocas do módulo de codificação de entropia exemplar 56 descrito com relação à Figura 5. Dessa maneira, o módulo de decodificação de entropia 70 pode realizar a decodificação de entropia com base nas técnicas de atribuição de contexto aqui descritas.

[0120] O módulo de decodificação aritmética 702 recebe um fluxo de bits codificado. Como mostrado na Figura 8, o módulo de decodificação aritmética 702 pode processar os valores binários codificados de acordo com o percurso de desvio ou percurso de codificação regular. Uma indicação de se um valor binário codificado deve ser processado de acordo com um percurso de desvio ou um percurso regular pode ser sinalizado no fluxo de bits com sintaxe de nível superior. Consistente com o processo CABAC descrito acima, no caso onde o módulo de decodificação aritmética 702 recebe os valores binários a partir de um percurso de desvio, o mecanismo de decodificação de desvio 704 pode realizar decodificação aritmética nos valores binários sem utilizar um contexto atribuído a um valor binário. Em um exemplo, o mecanismo de decodificação de desvio 704 pode supor probabilidades iguais para possíveis valores de um binário.

[0121] No caso onde o módulo de decodificação aritmética 702 recebe valores binários através do percurso regular, o módulo de modelagem de contexto 708 pode prover uma variável de contexto, de tal modo que o mecanismo de decodificação regular 706 pode realizar decodificação aritmética com base nas atribuições de contexto providas pelo módulo de modelagem de contexto 708. As atribuições de contexto podem ser definidas de acordo com os exemplos descritos

acima com relação às Tabelas 2-13. Os modelos de contexto podem ser armazenados em memória. O módulo de modelagem de contexto 708 pode incluir uma série de tabelas indexadas e/ou utilizar funções de mapeamento para determinar um contexto e uma porção de contexto variável de fluxo de bits codificado. Após decodificar um valor binário, o mecanismo de decodificação regular 706, pode atualizar um contexto com base nos valores binários decodificados. Adicionalmente, o módulo de binarização inversa 710 pode realizar uma binarização inversa em um valor binário e usar uma função de equiparação binária para determinar se um valor binário é válido. O módulo de binarização inversa 710 também pode atualizar o módulo de modelagem de contexto com base na determinação de equiparação. Assim, o módulo de binarização inversa 710 produz elementos de sintaxe de acordo com uma técnica de decodificação de contexto adaptativo. Dessa maneira, o módulo de decodificação de entropia 70 é configurado para decodificar um ou mais elementos de sintaxe com base nas técnicas de atribuição de contexto aqui descritas.

[0122] A Figura 9 é um fluxograma ilustrando um método exemplar para determinar um valor indicando a posição de um último coeficiente significativo dentro de um coeficiente de transformada a partir de uma sequência binária de acordo com as técnicas dessa revelação. O método descrito na Figura 9 pode ser realizado mediante qualquer um dos decodificadores de vídeo exemplares ou unidades de decodificação de entropia aqui descritas. Na etapa 902, um fluxo de bits codificado é obtido. O fluxo de bits codificado pode ser codificado de acordo com um processo de codificação CABAC ou outro processo de codificação de entropia. Na etapa 904, um contexto é determinado para uma porção da sequência binária codificada. Um contexto pode ser atribuído a um binário codificado com base nas técnicas aqui descritas. O contexto pode ser determinado por um decodificador de entropia ou vídeo acessando uma tabela de consulta ou realizando a função de mapeamento. O

contexto pode ser determinado com base em sintaxe de nível superior provida no fluxo de bits codificado. O contexto pode ser usado para derivar uma variável de contexto específica para um binário codificado específico. Conforme descrito acima, uma variável de contexto pode ser um valor binário de 7 bits indicando uma de 64 possíveis probabilidades (estados) e um estado mais provável (por exemplo, “1” ou “0”) e em alguns casos, os binários podem compartilhar os contextos. Na etapa 906, uma sequência binária é decodificada utilizando um processo de decodificação aritmética que utiliza uma variável de contexto, tal como CABAC. Uma sequência binária pode ser decodificada em uma base de binário por binário em que um modelo de contexto é atualizado após a decodificação de cada binário. O fluxo de bits decodificado pode incluir elementos de sintaxe que são usados adicionalmente para decodificar os coeficientes de transformada associados com os dados de vídeo codificados. Dessa maneira, a atribuição dos contextos para binários específicos utilizando as técnicas descritas acima pode prover decodificação eficiente de dados de vídeo de decodificação.

[0123] Em um ou mais exemplos, as funções descritas podem ser implementadas em hardware, software, firmware ou qualquer combinação dos mesmos. Se implementada em software, as funções podem ser armazenadas no, ou transmitidas através do, como uma ou mais instruções ou código, meio legível por computador e executadas por uma unidade de processamento baseado em hardware. Meios legíveis por computador podem incluir meios de armazenamento legíveis por computador, os quais correspondem a um meio tangível; tal como os meios de armazenamento de dados, ou meios de comunicação incluindo qualquer meio que facilita a transferência de um programa de computador de um local para outro, por exemplo, de acordo com um protocolo de comunicação. Dessa maneira, meios legíveis por computador geralmente podem corresponder a (1) meios de armazenamento tangíveis legíveis por computador os quais são não transitórios ou

(2) um meio de comunicação tal com um sinal ou onda portadora. Os meios de armazenamento de dados podem ser quaisquer meios disponíveis que podem ser acessados por um ou mais computadores ou um ou mais processadores para recuperar instruções, código e/ou estruturas de dados para implementação das técnicas descritas nessa revelação. Um produto de programa de computador pode incluir um meio legível por computador.

[0124] Como um exemplo, e não como limitação, tais meios de armazenamento legíveis por computador podem compreender RAM, ROM, EEPROM, CD-ROM ou qualquer outro meio de armazenamento de dados de estado sólido, ótico ou magnético, incluindo armazenamento de disco ótico, armazenamento de disco magnético, ou outros dispositivos de armazenamento magnético, ou qualquer outro meio que possa ser usado para armazenar código de programa desejado na forma de instruções ou estruturas de dados e que possam ser acessados por um computador. Além disso, qualquer conexão é propriamente denominada meio legível por computador. Por exemplo, se as instruções forem transmitidas a partir de um sítio de rede, servidor, ou outra fonte remota utilizando um cabo coaxial, cabo de fibras óticas, par de fios torcidos, linha de assinante digital (DSL), ou tecnologias sem fio tal como infravermelho, rádio e micro-ondas, então o cabo axial, cabo de fibras óticas, par de fios torcidos, DSL, ou tecnologias sem fio tais como infravermelho, rádio e micro-ondas são incluídos na definição de meio. Deve ser entendido, contudo, que meios de armazenamento tangíveis legíveis por computador e meios de armazenamento de dados não incluem conexões, ondas portadoras, sinais ou outros meios transientes, mas em vez disso são dirigidos aos meios de armazenamento não transientes, tangíveis. Disco magnético e disco ótico, conforme aqui usado, incluem disco compacto (CD), disco a laser, disco ótico, disco digital versátil (DVD), disquete e disco Blu-ray, onde discos magnéticos normalmente reproduzem os dados magneticamente, enquanto que os discos óticos reproduzem

os dados oticamente com laseres. Combinações dos mencionados acima também devem ser incluídas no escopo de meios legíveis por computador.

[0125] Instruções podem ser executadas por um ou mais processadores, tal como um ou mais processadores de sinal digital (DSPs), microprocessadores de uso geral, circuitos integrados de aplicação específica (ASICs), arranjos lógicos programáveis no campo (FPGAs), ou outros circuitos lógicos integrados ou discretos equivalentes. Consequentemente, o termo ”processador”, conforme aqui usado pode se referir a qualquer uma da estrutura precedente ou qualquer outra estrutura adequada para implementação das técnicas aqui descritas. Além disso, em alguns aspectos, a funcionalidade aqui descrita pode ser provida dentro de módulos de hardware e/ou software, dedicados configurados para codificar e decodificar, ou incorporados em um codec combinado. Além disso, as técnicas poderiam ser completamente implementadas em um ou mais circuitos ou elementos lógicos.

[0126] As técnicas dessa revelação podem ser implementadas em uma ampla variedade de dispositivos ou equipamentos, incluindo um aparelho telefônico sem fio, um circuito integrado (IC) ou um conjunto de ICs (por exemplo, um conjunto de chip). Vários componentes, módulos ou unidades são descritos nessa revelação para enfatizar os aspectos funcionais de dispositivos configurados para realizar as técnicas reveladas, mas não necessariamente requerem a realização por diferentes unidades de hardware. Mais propriamente, conforme descrito acima, as várias unidades podem ser combinadas em uma unidade de hardware de codec ou providas por um grupo de unidades de hardware inter operativas, incluindo um ou mais processadores conforme descrito acima, em conjunto com software e/ou firmware adequado.

[0127] Vários exemplos foram descritos. Esses e outros exemplos estão dentro do escopo das reivindicações a seguir.

REIVINDICAÇÕES

1. Método para decodificar dados de vídeo compreendendo:

decodificar uma primeira sequência binária e uma segunda sequência binária a partir de um fluxo de bits codificado usando uma codificação aritmética binária adaptável ao contexto, processo CABAC, o processo CABAC compreendendo decodificar a primeira sequência binária com base em um modelo de contexto e decodificar por desvio a segunda sequência binária; e

determinar um valor indicando a posição de um último coeficiente significativo dentro de um bloco de vídeo de tamanho T com base na primeira sequência binária e na segunda sequência binária,

em que a primeira sequência binária é definida por um esquema de codificação unário truncado, e

em que a segunda sequência binária é definida por um esquema de codificação de comprimento fixo que inclui determinar um comprimento da segunda sequência binária com base em um valor da primeira sequência binária, **CARACTERIZADO** pelo fato de que o esquema de codificação unário truncado, pelo qual a primeira sequência binária é determinada, é definido por um comprimento máximo de bit definido por 2log2(T)-1.

2. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que T é igual a 32, em que o valor indicando a posição de um último coeficiente significativo é igual a 8, e em que a primeira sequência binária tem um comprimento de bit de 7.

3. Método, de acordo com qualquer uma das reivindicações 1 e 2, **CARACTERIZADO** pelo fato de que a decodificação de desvio da segunda sequência binária compreende a decodificação dos binários da segunda sequência binária usando um estado de probabilidade igual.

4. Método, de acordo com qualquer uma das reivindicações 1 a 3, **CARACTERIZADO** pelo fato de que o valor que indica a posição do último coeficiente significativo compreende um dentre um valor que indica uma coordenada x do último coeficiente significativo ou um valor que indica uma coordenada y do último coeficiente significativo.

5. Dispositivo para decodificar dados de vídeo, o dispositivo

compreendendo:

meios para decodificar uma primeira sequência binária e uma segunda sequência binária a partir de um fluxo de bits codificado usando uma codificação aritmética binária adaptável ao contexto, processo CABAC, o processo CABAC compreendendo decodificar a primeira sequência binária com base em um modelo de contexto e decodificar por desvio a segunda sequência binária; e

meios para determinar um valor indicando a posição de um último coeficiente significativo dentro de um bloco de vídeo de tamanho T com base na primeira sequência binária e na segunda sequência binária,

em que a primeira sequência binária é definida por um esquema de codificação unário truncado, e

em que a segunda sequência binária é definida por um esquema de codificação de comprimento fixo que inclui a determinação do comprimento da segunda sequência binária com base em um valor da primeira sequência binária, **CARACTERIZADO** pelo fato de que o esquema de codificação unário truncado, pelo qual a primeira sequência binária é determinada, é definido por um comprimento máximo de bit definido por 2log2(T)-1.

6. Dispositivo, de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que T é igual a 32, em que o valor indicando a posição de um último coeficiente significativo é igual a 8, e em que a primeira sequência binária tem um comprimento de 7.

7. Dispositivo, de acordo com qualquer uma das reivindicações 5 e 6, **CARACTERIZADO** pelo fato de que os meios de decodificação compreendem os meios de decodificação de desvio de binários da segunda sequência binária usando um estado de probabilidade igual.

8. Dispositivo, de acordo com qualquer uma das reivindicações 5 a 7, **CARACTERIZADO** pelo fato de que o valor indicando a posição do último coeficiente significativo compreende um dentre um valor indicando uma coordenada x do último coeficiente significativo ou um valor indicando uma coordenada y do último coeficiente significativo.

9. Meio de armazenamento não transitório legível por computador **CARACTERIZADO** pelo fato de que compreende instruções armazenadas no

mesmo em que, quando executado por um computador, faz com que o computador execute os passos do método conforme definido em qualquer uma das reivindicações 1 a 4.

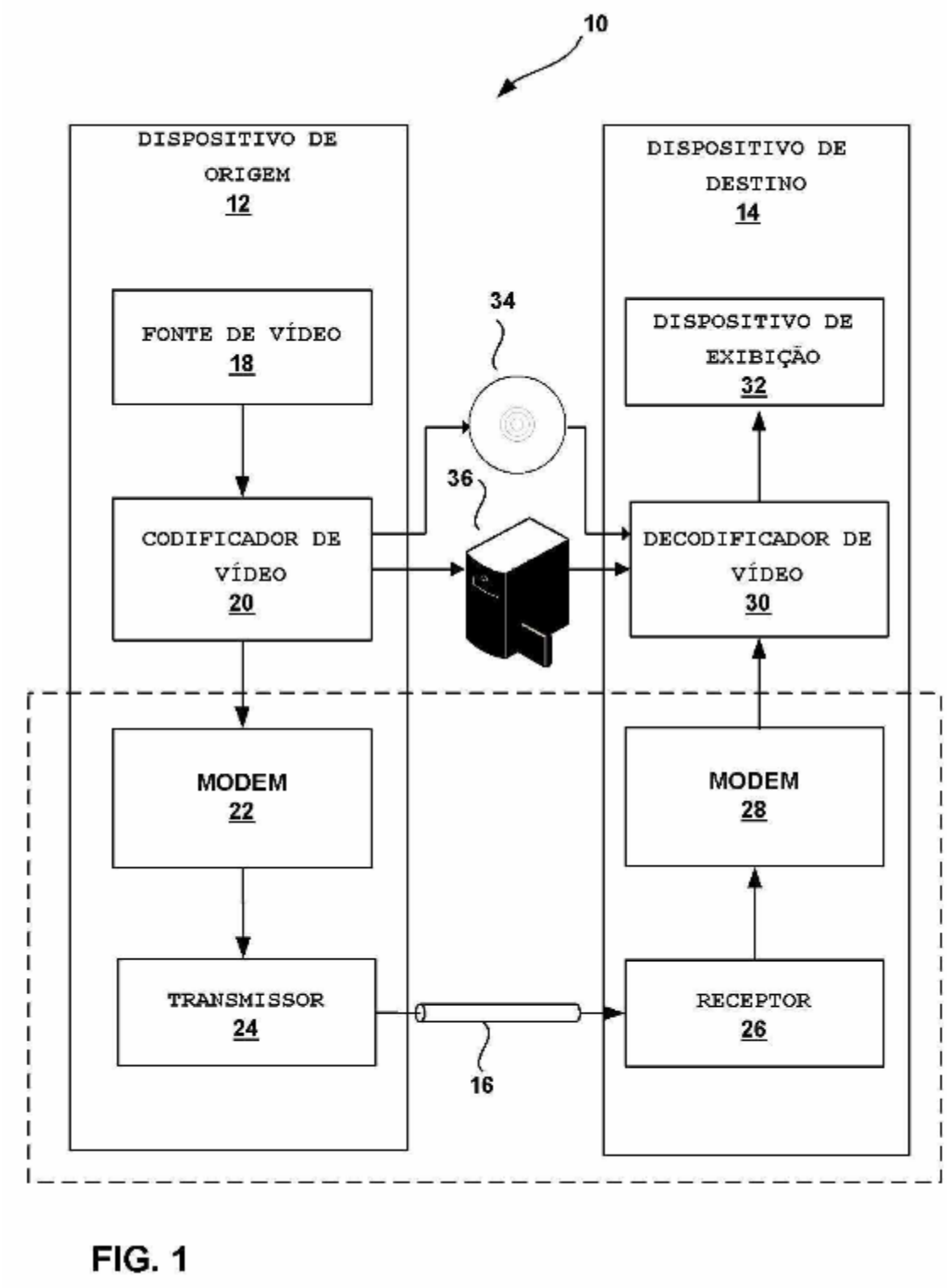

10
DISPOSITIVO DE ORIGEM
12
DISPOSITIVO DE DESTINO
14
FONTE DE VÍDEO
18
34
DISPOSITIVO DE EXIBIÇÃO
32
36
CODIFICADOR DE VÍDEO
20
DECODIFICADOR DE VÍDEO
30
MODEM
22
MODEM
28
TRANSMISSOR
24
RECEPTOR
26
16

FIG. 1

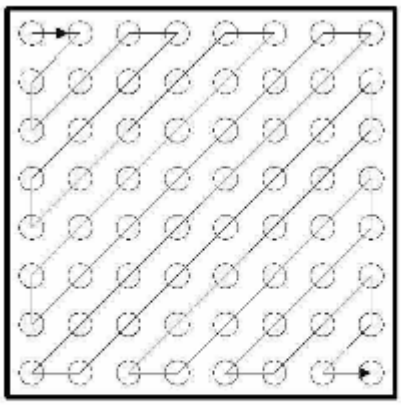


FIG. 2A

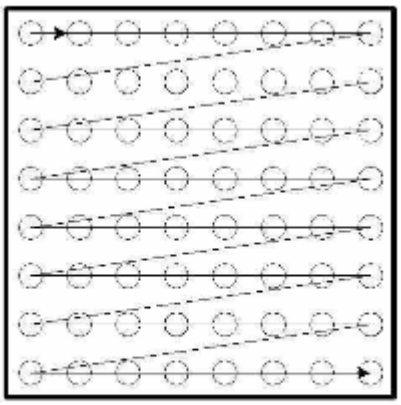


FIG. 2B

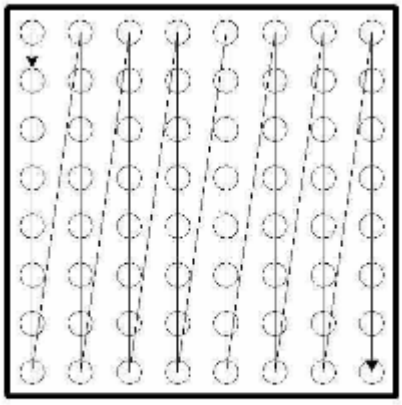


FIG. 2C

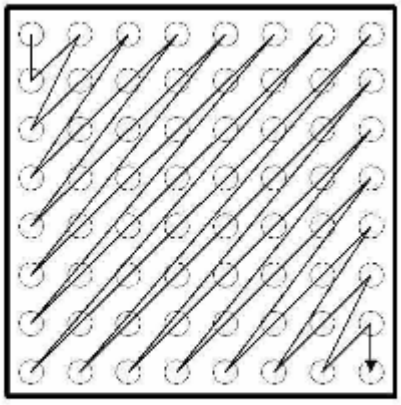


FIG. 2D

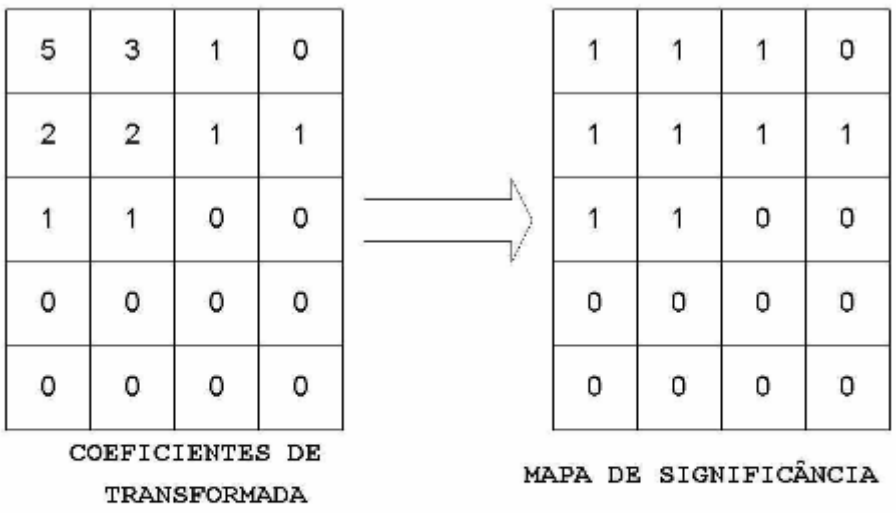

5 3 1 0
2 2 1 1
1 1 0 0
0 0 0 0
0 0 0 0
COEFICIENTES DE TRANSFORMADA
1 1 1 0
1 1 1 1
1 1 0 0
0 0 0 0
0 0 0 0
MAPA DE SIGNIFICÂNCIA

FIG. 3

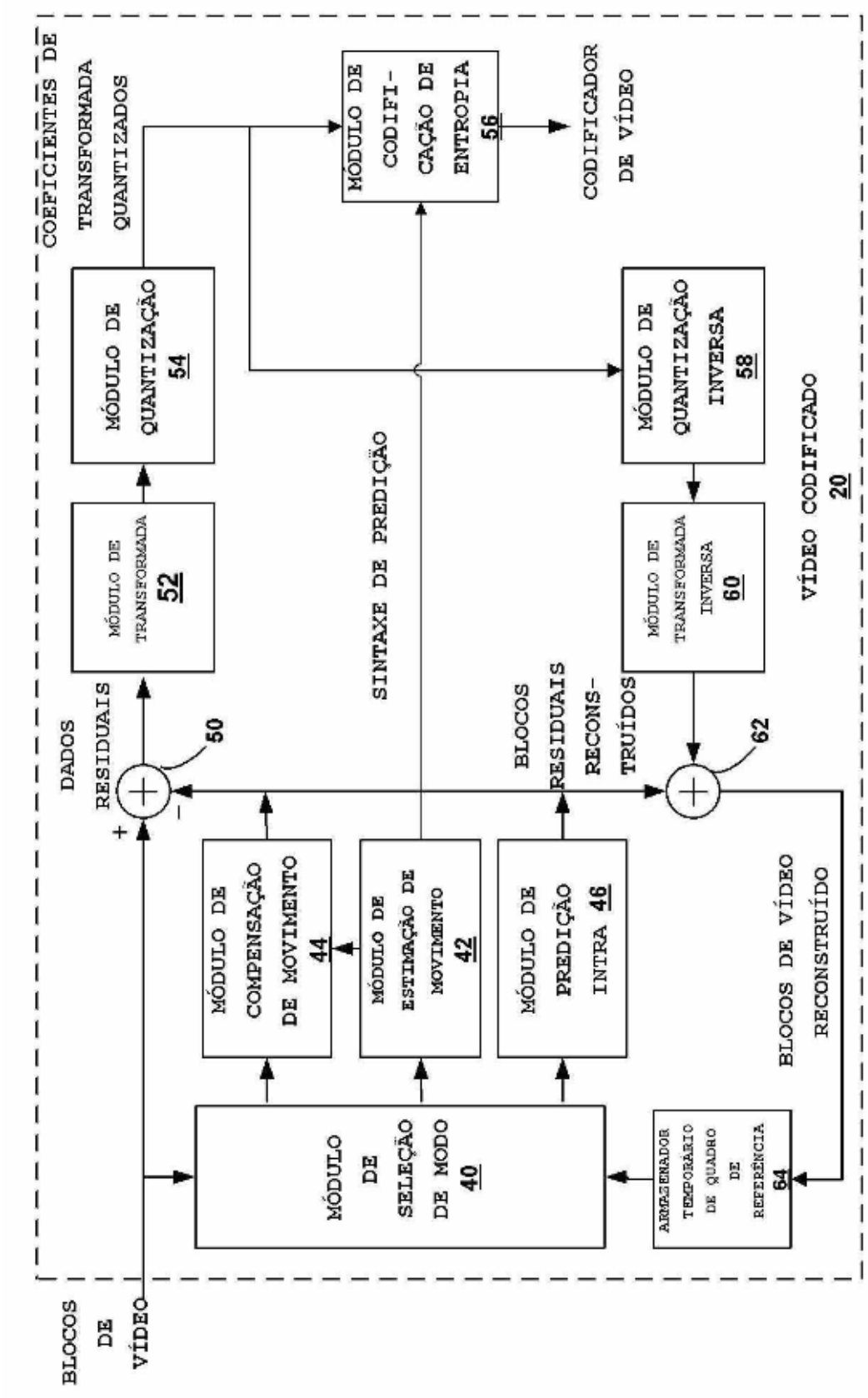

BLOCOS DE VÍDEO
COEFICIENTES DE TRANSFORMADA QUANTIZADOS
MÓDULO DE QUANTIZAÇÃO 54
MÓDULO DE TRANSFORMADA 52
MÓDULO DE CODIFI-CAÇÃO DE ENTROPIA 56
CODIFICADOR DE VÍDEO
MÓDULO DE QUANTIZAÇÃO INVERSA 58
MÓDULO DE TRANSFORMADA INVERSA 60
SINTAXE DE PREDIÇÃO
DADOS RESIDUAIS
50
62
BLOCOS RESIDUAIS RECONS-TRUÍDOS
VÍDEO CODIFICADO 20
MÓDULO DE COMPENSAÇÃO DE MOVIMENTO 44
MÓDULO DE ESTIMAÇÃO DE MOVIMENTO 42
MÓDULO DE PREDIÇÃO INTRA 46
MÓDULO DE SELEÇÃO DE MODO 40
ARMAZENADOR TEMPORÁRIO DE QUADRO DE REFERÊNCIA 64
BLOCOS DE VÍDEO RECONSTRUÍDO

FIG. 4

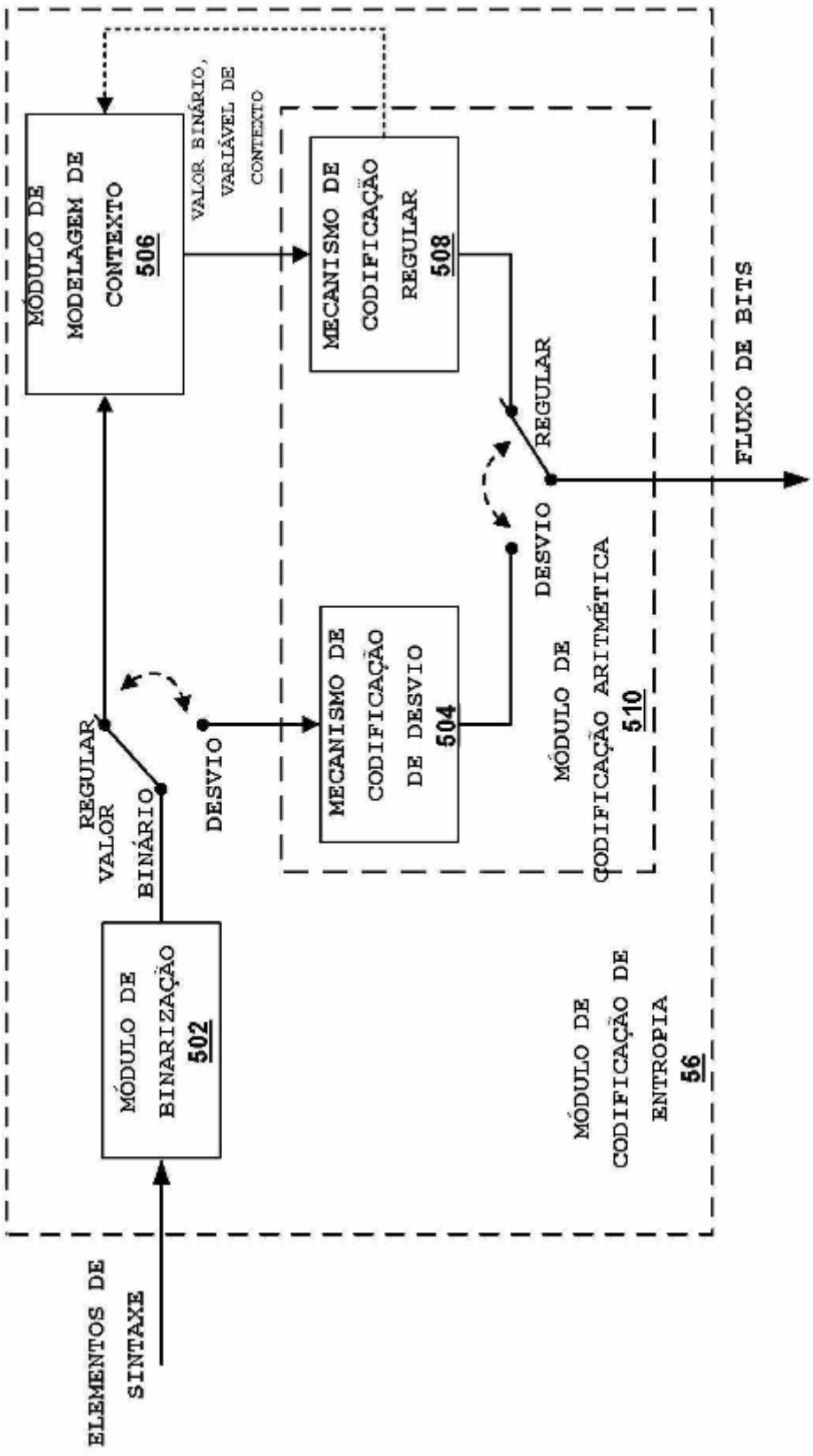

ELEMENTOS DE SINTAXE
MÓDULO DE BINARIZAÇÃO 502
REGULAR
VALOR BINÁRIO
DESVIO
MÓDULO DE MODELAGEM DE CONTEXTO 506
VALOR BINÁRIO, VARIÁVEL DE CONTEXTO
MECANISMO DE CODIFICAÇÃO DE DESVIO 504
MECANISMO DE CODIFICAÇÃO REGULAR 508
DESVIO
REGULAR
MÓDULO DE CODIFICAÇÃO ARITMÉTICA 510
MÓDULO DE CODIFICAÇÃO DE ENTROPIA 56
FLUXO DE BITS

FIG. 5

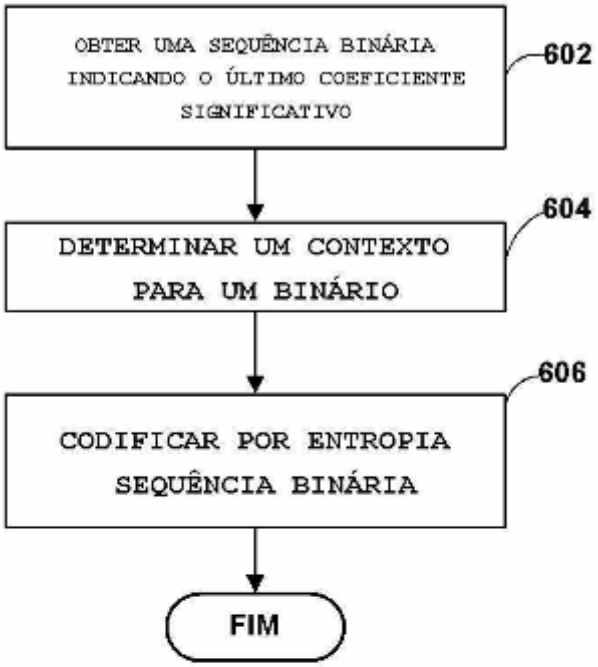

OBTER UMA SEQUÊNCIA BINÁRIA INDICANDO O ÚLTIMO COEFICIENTE SIGNIFICATIVO
602
DETERMINAR UM CONTEXTO PARA UM BINÁRIO
604
CODIFICAR POR ENTROPIA SEQUÊNCIA BINÁRIA
606
FIM

FIG. 6

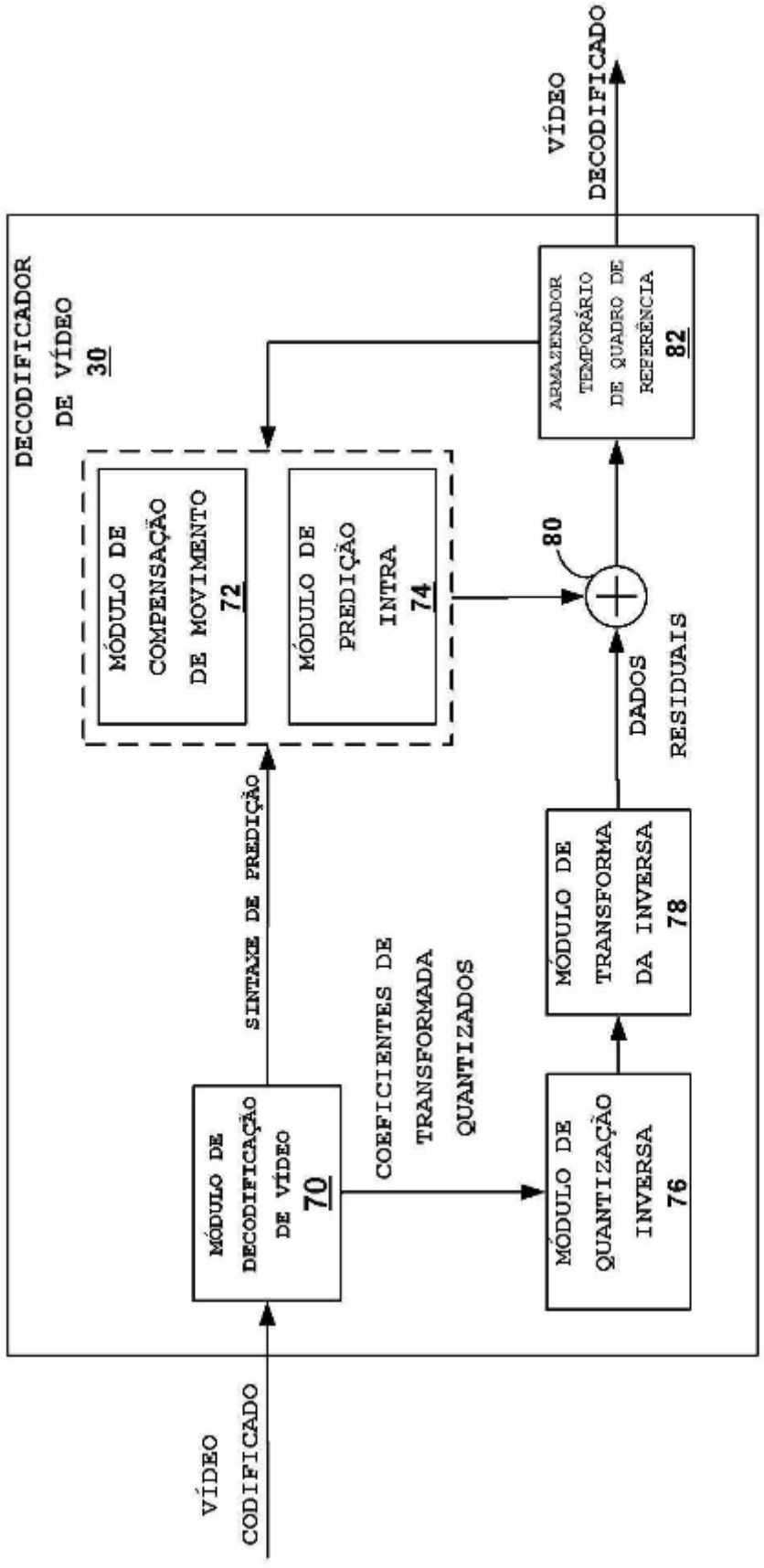

VÍDEO CODIFICADO
DECODIFICADOR DE VÍDEO 30
MÓDULO DE DECODIFICAÇÃO DE VÍDEO 70
SINTAXE DE PREDIÇÃO
COEFICIENTES DE TRANSFORMADA QUANTIZADOS
MÓDULO DE QUANTIZAÇÃO INVERSA 76
MÓDULO DE TRANSFORMA DA INVERSA 78
DADOS RESIDUAIS
80
MÓDULO DE COMPENSAÇÃO DE MOVIMENTO 72
MÓDULO DE PREDIÇÃO INTRA 74
ARMAZENADOR TEMPORÁRIO DE QUADRO DE REFERÊNCIA 82
VÍDEO DECODIFICADO

FIG. 7

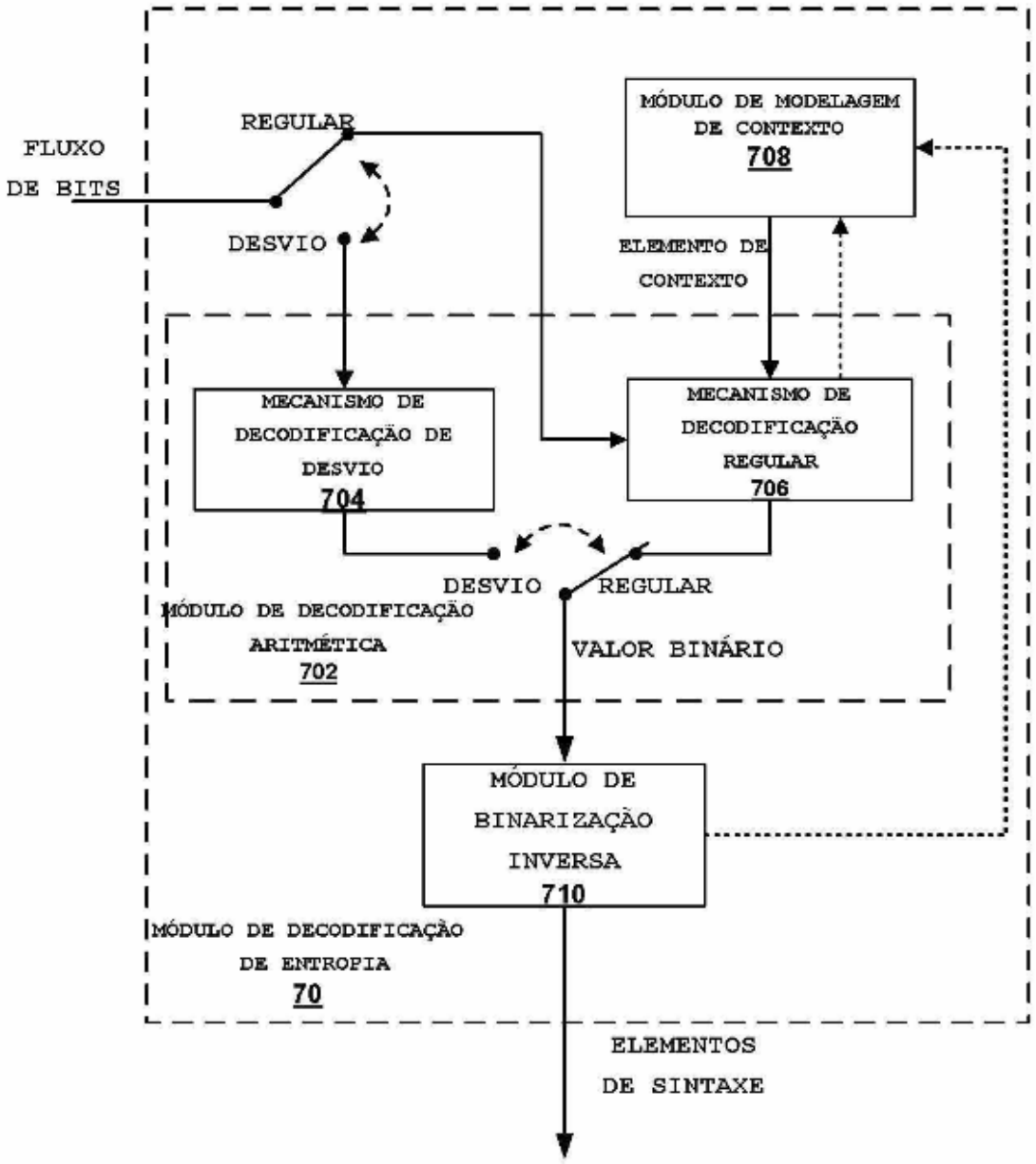
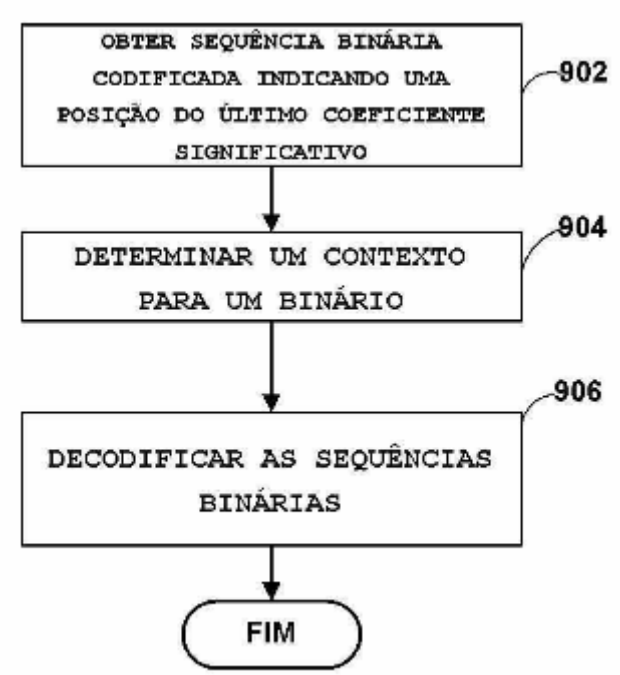

FLUXO
DE BITS
REGULAR
DESVIO
MÓDULO DE MODELAGEM
DE CONTEXTO
708
ELEMENTO DE
CONTEXTO
MECANISMO DE
DECODIFICAÇÃO DE
DESVIO
704
MECANISMO DE
DECODIFICAÇÃO
REGULAR
706
DESVIO
REGULAR
MÓDULO DE DECODIFICAÇÃO
ARITMÉTICA
702
VALOR BINÁRIO
MÓDULO DE
BINARIZAÇÃO
INVERSA
710
MÓDULO DE DECODIFICAÇÃO
DE ENTROPIA
70
ELEMENTOS
DE SINTAXE

FIG. 8


OBTER SEQUÊNCIA BINÁRIA CODIFICADA INDICANDO UMA POSIÇÃO DO ÚLTIMO COEFICIENTE SIGNIFICATIVO
902
DETERMINAR UM CONTEXTO PARA UM BINÁRIO
904
DECODIFICAR AS SEQUÊNCIAS BINÁRIAS
906
FIM