



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

(11) BR 112016021142-1 B1



(22) Data do Depósito: 13/03/2015

(45) Data de Concessão: 20/09/2022

(54) Título: MÉTODO E DISPOSITIVO DE DECODIFICAÇÃO DE DADOS DE VÍDEO, MÉTODO E DISPOSITIVO DE CODIFICAÇÃO DE DADOS DE VÍDEO, E, MEMÓRIA LEGÍVEL POR COMPUTADOR

(51) Int.Cl.: H04N 19/176; H04N 19/70; H04N 19/134; H04N 19/126; H04N 19/14; (...).

(30) Prioridade Unionista: 10/10/2014 US 62/062,637; 12/03/2015 US 14/656,482; 14/03/2014 US 61/953,573; 18/04/2014 US 61/981,645.

(73) Titular(es): QUALCOMM INCORPORATED.

(72) Inventor(es): LI ZHANG; JIANLE CHEN; MARTA KARCZEWICZ; JOEL SOLE ROJALS; WOO-SHIK KIM.

(86) Pedido PCT: PCT US2015020543 de 13/03/2015

(87) Publicação PCT: WO 2015/138962 de 17/09/2015

(85) Data do Início da Fase Nacional: 13/09/2016

(57) Resumo: PARÂMETROS DE QUANTIZAÇÃO PARA CODIFICAÇÃO COM CONVERSÃO NO ESPAÇO DE CORES. A presente invenção se refere, em geral, à codificação de blocos de vídeo, usando um processo de conversão no espaço de cores. Um codificador de vídeo, tal como um codificador de vídeo ou um decodificador de vídeo, pode determinar quanto ao uso de conversão no espaço de cores para codificar os dados de vídeo. Em resposta à determinação de usar conversão no espaço de cores, o codificador de vídeo pode quantizar os dados de um primeiro componente de cor dos dados de vídeo, usando um primeiro desvio de um primeiro parâmetro de quantização (QP) e quantizar os dados de um segundo componente de cor dos dados de vídeo usando um desvio de um segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e o segundo QP é diferente do primeiro QP. O codificador de vídeo pode ainda quantizar de forma inversa dados do primeiro componente de cor usando o primeiro desvio e quantizar de forma inversa dados do segundo componente de cor usando o segundo desvio.

“MÉTODO E DISPOSITIVO DE DECODIFICAÇÃO DE DADOS DE VÍDEO,
MÉTODO E DISPOSITIVO DE CODIFICAÇÃO DE DADOS DE VÍDEO, E,
MEMÓRIA LEGÍVEL POR COMPUTADOR”

[0001] Este pedido de patente reivindica o benefício do Pedido de Patente provisório US n° 61/953,573, depositado em 14 de março de 2014, Pedido de Patente provisório US n° 61/981,645, depositado em 18 de abril de 2014 e Pedido de Patente provisório US 62/062,637, depositado em 10 de outubro de 2014, cujo conteúdo total é aqui incorporado por referência.

CAMPO TÉCNICO

[0002] Esta invenção se refere à codificação de vídeo e, mais especificamente, codificação de vídeo usando conversão de espaço de cores.

FUNDAMENTOS DA INVENÇÃO

[0003] As capacidades de vídeo digital podem ser incorporadas em uma ampla gama de dispositivos, incluindo televisões digitais, sistemas de transmissão direta digital, sistemas de transmissão sem fio, assistentes pessoais digitais (PDAs), laptop ou computadores, tablets, leitores de e-book, câmeras digitais, dispositivos de gravação de vídeo digital, reprodutores de mídia digital, dispositivos de vídeo game, consoles de vídeo game, telefones celulares ou telefones por rádio satélite, os assim chamados “smartphones”, dispositivos de videoconferência, dispositivos de transferência de vídeo e similares. Os dispositivos de vídeo digital implementam as técnicas de codificação 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), padrão de codificação de vídeo de alta eficiência (HEVC) atualmente sob desenvolvimento e extensões de tais padrões. Os dispositivos de vídeo podem

transmitir, receber, codificar, decodificar e/ou armazenar informação de vídeo digital de modo mais eficiente através da implementação de tais técnicas de codificação de vídeo.

[0004] Técnicas de codificação de vídeo incluem predição espacial (intraimagem) e/ou predição temporal (interimagem) para reduzir ou remover a redundância inerente nas sequências de vídeo. Para a codificação de vídeos em blocos, uma parte do vídeo (por exemplo, um quadro de vídeo ou uma parte de um quadro de vídeo) pode ser dividida em blocos de vídeo, os quais também podem ser referidos como treeblocks, unidades de codificação (CUs) e/ou nós de codificação. Os blocos de vídeo em uma fatia (I) intracodificada de uma imagem são codificados usando a predição espacial com relação às amostras de referência nos blocos próximos na mesma imagem. Os blocos de vídeo em uma fatia intercodificada (P ou B) de uma imagem podem usar a predição espacial com referência a amostras de referência nos blocos próximos na mesma imagem ou predição temporal com referência a 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.

[0005] A predição espacial ou predição temporal resultam em um bloco de predição para um bloco a ser codificado. Os dados residuais representam as diferenças de pixel entre o bloco original a ser codificado e o bloco de predição. Um bloco intercodificado é codificado de acordo com um vetor de movimento que aponta para um bloco de amostras de referência formando o bloco de predição e os dados residuais indicando a diferença entre o bloco codificado e o bloco de predição. Um bloco intracodificado é codificado de acordo com um modo de intracodificação e os dados residuais. Para compressão adicional, os dados

residuais podem ser transformados do domínio de pixel em um domínio de transformada, resultando em coeficientes de transformada residuais, os quais podem ser então quantizados. Os coeficientes de transformada quantizados, inicialmente dispostos em uma ordem bidimensional, podem ser digitalizados de modo a produzir um vetor unidimensional de coeficientes de transformada e a codificação de entropia pode ser aplicada para alcançar ainda mais compressão.

SUMÁRIO DA INVENÇÃO

[0006] Em geral, esta revelação descreve técnicas para codificar os blocos de vídeo usando um processo de conversão no espaço de cores. Um codificador de vídeo, tal como um codificador de vídeo ou um decodificador de vídeo, pode determinar se usa a conversão no espaço de cores para codificar os dados de vídeo. Em resposta a determinar se usa a conversão no espaço de cores para codificar os dados de vídeo, o codificador de vídeo pode quantizar os dados de um primeiro componente de cor dos dados de vídeo usando um primeiro desvio de um primeiro parâmetro de quantização (QP) e quantizar os dados de um segundo componente de cor dos dados de vídeo usando um segundo desvio de um segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio do segundo QP é diferente do primeiro desvio do primeiro QP. O codificador de vídeo também pode determinar se os dados de vídeo foram codificados usando a conversão no espaço de cores. Em resposta a determinar se os dados de vídeo foram codificados usando a conversão no espaço de cores, o codificador de vídeo pode quantizar de forma inversa os dados do primeiro componente de cor dos dados de vídeo usando o primeiro desvio do primeiro QP e quantizar de forma inversa os dados do segundo componente de cor dos

dados de vídeo usando o segundo desvio do segundo QP. Em resposta a determinar que os dados de vídeo não foram codificados usando a conversão no espaço de cores, o codificador de vídeo pode quantizar de forma inversa os dados de vídeo usando o primeiro QP e o segundo QP sem o primeiro e o segundo desvio.

[0007] Em um exemplo, um método para decodificação de dados de vídeo é descrito. O método compreende determinar se os dados de vídeo foram codificados usando a conversão no espaço de cores. O método também inclui, em resposta a determinar se os dados de vídeo foram codificados usando a conversão no espaço de cores, a quantização inversa de dados de um primeiro componente de cor dos dados de vídeo usando um primeiro desvio de um primeiro parâmetro de quantização (QP) e a quantização inversa de dados de um segundo componente de cor dos dados de vídeo usando um segundo desvio de um segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio do segundo QP é diferente do primeiro desvio do primeiro QP.

[0008] Em outro exemplo, um dispositivo inclui uma memória configurada para armazenar dados de vídeo e um ou mais processadores configurados para determinar se os dados de vídeo foram codificados usando a conversão no espaço de cores. Além disso, um ou mais processadores também são configurados para, em resposta a determinar se os dados de vídeo foram codificados usando a conversão no espaço de cores, quantizar de forma inversa os dados de um primeiro componente de cor dos dados de vídeo usando um primeiro desvio de um primeiro parâmetro de quantização (QP) e quantizar de forma inversa os dados de um segundo componente de cor dos dados de vídeo usando um segundo desvio de um segundo QP, em que o segundo componente de cor

é diferente do primeiro componente de cor, e em que o segundo desvio do segundo QP é diferente do primeiro desvio do primeiro QP.

[0009] Em outro exemplo, um dispositivo inclui meios para determinar se os dados de vídeo foram codificados usando a conversão no espaço de cores. O dispositivo também inclui, em resposta a determinar se os dados de vídeo foram codificados usando a conversão no espaço de cores, meios para uma quantização inversa de dados de um primeiro componente de cor dos dados de vídeo usando um primeiro desvio de um primeiro parâmetro de quantização (QP) e meios para uma quantização inversa de dados de um segundo componente de cor dos dados de vídeo usando um segundo desvio de um segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio do segundo QP é diferente do primeiro desvio do primeiro QP.

[0010] Em outro exemplo, um meio de armazenamento legível por computador compreendendo instruções que, quando executadas, causam um ou mais processadores a determinar se os dados de vídeo foram codificados usando a conversão no espaço de cores. As instruções, quando executadas, também fazem o processador, em resposta a determinar se os dados de vídeo foram codificados usando a conversão no espaço de cores, quantizar de forma inversa os dados de um primeiro componente de cor dos dados de vídeo usando um primeiro desvio de um primeiro parâmetro de quantização (QP) e quantizar de forma inversa os dados de um segundo componente de cor dos dados de vídeo usando um segundo desvio de um segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio do segundo QP é diferente do primeiro desvio do primeiro QP.

[0011] Em outro exemplo, um método para codificar dados de vídeo é descrito. O método compreende determinar se usa a conversão no espaço de cores para codificar os dados de vídeo. O método também inclui, em resposta a determinar se usa a conversão no espaço de cores para codificar os dados de vídeo, quantizar os dados de um primeiro componente de cor dos dados de vídeo usando um primeiro desvio de um primeiro parâmetro de quantização (QP) e quantizar os dados de um segundo componente de cor dos dados de vídeo usando um segundo desvio de um segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio do segundo QP é diferente do primeiro desvio do primeiro QP.

[0012] Em outro exemplo, um dispositivo inclui uma memória configurada para armazenar dados de vídeo e um ou mais processadores configurados para determinar se usa a conversão no espaço de cores para codificar os dados de vídeo. Um ou mais processadores também são configurados para, em resposta a determinar se usa a conversão no espaço de cores para codificar os dados de vídeo, quantizar os dados de um primeiro componente de cor dos dados de vídeo usando um primeiro desvio de um primeiro parâmetro de quantização (QP) e quantizar os dados de um segundo componente de cor dos dados de vídeo usando um segundo desvio de um segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio do segundo QP é diferente do primeiro desvio do primeiro QP.

[0013] Em outro exemplo, um dispositivo inclui meios para determinar se usa a conversão no espaço de cores para codificar os dados de vídeo. O dispositivo também inclui, em resposta a determinar se usa a conversão no espaço de cores para codificar os dados de vídeo, meios

para quantizar os dados de um primeiro componente de cor dos dados de vídeo usando um primeiro desvio de um primeiro parâmetro de quantização (QP) e meios para quantizar os dados de um segundo componente de cor dos dados de vídeo usando um segundo desvio de um segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio do segundo QP é diferente do primeiro desvio do primeiro QP.

[0014] Em outro exemplo, um meio de armazenamento legível por computador compreende instruções que, quando executadas, fazem com que um ou mais processadores determinem se usam a conversão no espaço de cores para codificar os dados de vídeo. As instruções também fazem com que um ou mais processadores, em resposta a determinar se usam a conversão no espaço de cores para codificar os dados de vídeo, quantizem os dados de um primeiro componente de cor dos dados de vídeo usando um primeiro desvio de um primeiro parâmetro de quantização (QP) e quantizem os dados de um segundo componente de cor dos dados de vídeo usando um segundo desvio de um segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio do segundo QP é diferente do primeiro desvio do primeiro QP.

[0015] Os detalhes de um ou mais exemplos da revelação são apresentados nos desenhos anexos e na descrição abaixo. Outras características, objetivos e vantagens da revelação ficarão aparentes a partir da descrição e dos desenhos, e a partir das reivindicações.

BREVE DESCRIÇÃO DOS DESENHOS

[0016] A FIG. 1 é um diagrama de blocos que ilustra um sistema de codificação e decodificação de vídeo que pode usar as técnicas descritas nesta revelação.

[0017] A FIG. 2 é um diagrama de blocos que ilustra um codificador de vídeo exemplificador que pode implementar as técnicas descritas nesta revelação.

[0018] A FIG. 3 é um diagrama de blocos que ilustra um decodificador de vídeo exemplificador que pode implementar as técnicas descritas nesta revelação.

[0019] A FIG. 4 é um diagrama conceitual que ilustra os modos de predição 35 HEVC de acordo com uma ou mais técnicas da atual revelação.

[0020] A FIG. 5 é um diagrama conceitual que ilustra candidatos a vetor de movimento espacial próximo para consolidação e modos de predição de vetor de movimento avançado (AMVP) de acordo com uma ou mais técnicas da atual revelação.

[0021] A FIG. 6 é um diagrama conceitual que ilustra um exemplo de cópia de intrabloco (BC) de acordo com uma ou mais técnicas da atual revelação.

[0022] A FIG. 7 é um diagrama conceitual que ilustra um exemplo de um bloco alvo e uma amostra de referência para um intrabloco 8x8, de acordo com uma ou mais técnicas da atual revelação.

[0023] A FIG. 8 é um fluxograma que ilustra uma técnica de codificação de acordo com uma ou mais técnicas da atual revelação.

[0024] FIG. 9 é um fluxograma que ilustra uma técnica de decodificação de acordo com uma ou mais técnicas da atual revelação.

DESCRIÇÃO DETALHADA DA INVENÇÃO

[0025] Em alguns exemplos, esta revelação se refere à codificação de conteúdo da tela, em que o formato de amostragem de croma elevado 4:4:4 é usado. Em alguns exemplos, esta revelação também se aplica para extensões de faixa (RCEx), incluindo o suporte para

profundidade de bits elevada, (mais de 8 bits), formato de amostragem de croma elevada 4:4:4. Em alguns exemplos, esta revelação também se aplica a outros formatos coloridos, como o formato de amostragem de croma de 4:2:2. Mais especificamente, nesta revelação, muitas técnicas diferentes que se referem à conversão no espaço de cores são descritas.

[0026] Os padrões de codificação de vídeo incluem ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 ou ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (também conhecido como ISO/IEC MPEG-4 AVC), incluindo as suas extensões de codificação escalável de vídeo (SVC) e codificação em múltiplas vistas de vídeo (MVC).

[0027] O projeto de um novo padrão de codificação de vídeo, a saber, codificação de vídeo de alta eficiência (HEVC), foi finalizado pela Colaboração conjunta em codificação de vídeo (JCT-VC) do Grupo de Especialistas em Codificação de Vídeo (VCEG) do ITU-T e Grupo de Especialistas em Imagem em Movimento (MPEG) do ISO/IEC.

[0028] Em HEVC, a maior unidade de codificação em uma fatia é chamada um tree block de codificação (CTB). Um CTB contém um quad-tree de nós os quais são as unidades de codificação. O tamanho de um CTB pode variar de 16x16 a 64x64 no perfil principal de HEVC, embora os tamanhos de CTB 8x8 possam ser suportados. A unidade de codificação (CU) poderia ser do mesmo tamanho que o CTB e tão pequena quanto 8x8. Cada CU é codificada com um módulo. Quando uma CU é intercodificada, a CU também pode ser dividida nas duas unidades de predição (PUs) ou se tornar uma única PU quando divisões adicionais não se aplicam. Quando duas PUs estão presentes em uma CU, cada PU pode ser retângulos de tamanho médio ou dois retângulos com um tamanho igual a 1/4 ou 3/4 do tamanho da CU.

[0029] Quando a CU é intercodificada, um conjunto de informação de movimentos está presente para cada PU. Além disso, cada PU é codificada com um modo de interpredição único para derivar o conjunto de informação de movimento. Em HEVC, os menores tamanhos de PU são 8x4 e 4x8.

[0030] HEVC especifica quatro unidades de transformada (TUs) com tamanhos de 4x4, 8x8, 16x16 e 32x32 ao código de predição residual. Um CTB também pode ser dividido de maneira recorrente em 4 ou mais TUs. TUs usam funções baseadas em número inteiro que são similares às funções de transformada discreta de cosseno (DCT). Além disso, os blocos de transformada luminância 4x4 que pertencem a uma região intracodificada são transformadas usando uma transformada inteira que é derivada de uma função de transformada discreta de seno (DST). A crominância usa os mesmos tamanhos de TU que a luminância.

[0031] No padrão HEVC atual, para o componente luminância de cada unidade de predição (PU), um método de intrapredição é usado com 33 modos de predição angular (indexados a partir de 2 a 34), um modo DC (indexado com 1), e um modo planar (indexado com 0), conforme descrito abaixo com referência à FIG. 4.

[0032] Além dos 35 modos intra acima, mais um modo, nomeado 'I-PCM', também é empregado por HEVC. No modo I-PCM, a codificação de predição, transformada, quantização e entropia são ultrapassadas enquanto que as amostras de predição são codificadas por um número predefinido de bits. O principal objetivo do modo I-PCM é lidar com a situação quando o sinal não pode ser codificado de forma eficiente por outros modos.

[0033] No padrão HEVC atual, existem dois modos de predição disponíveis. Um tal modo é o modo de mesclagem

(skip é considerado um caso especial de mesclagem) e o segundo modo é o modo de predição de vetor de movimento avançado (AMVP) para a unidade de predição (PU).

[0034] Em ambos os modos AMVP e de mesclagem, uma lista de candidatos a vetor de movimento (MV) é mantida para os preditores de vetor de movimento múltiplos. O(s) vetore(s) de movimento, assim como os índices de referência no modo de mesclagem, da PU atual são gerados ao se tomar um candidato da lista de candidatos de MV.

[0035] A lista de candidatos de MV podem conter até 5 candidatos para o modo de mesclagem e apenas dois candidatos para o modo AMVP. Um candidato de mesclagem pode conter um conjunto de informação de movimento, por exemplo, vetores de movimento correspondendo a ambas as listas de imagem de referência (lista 0 e lista 1) e os índices de referência. Se um candidato de mesclagem é identificado por um índice de mesclagem, as imagens de referência são usadas para a predição dos blocos atuais, assim como os vetores de movimento associado são determinados. No entanto, sob o modo AMVP para cada direção de predição potencial de cada lista 0 ou lista 1, um índice de referência precisa ser explicitamente sinalizado, junto com o índice de MVP para a lista de candidato de MV uma vez que o candidato de AMVP contém apenas um vetor de movimento. No modo de AMVP, os vetores de movimento preditos também podem ser refinados.

[0036] um candidato de mesclagem corresponde a um conjunto completo de informação de movimento enquanto que um candidato de AMVP pode conter apenas um vetor de movimento para um índice de referência e direção de predição específica. Os candidatos para ambos os modos são derivados similarmente dos mesmos blocos próximos espaciais e temporais.

[0037] Conforme descrito abaixo com referência à FIG. 5, os candidatos de MV espacial são derivados a partir dos blocos próximos mostrados na FIG. 5, para um PU específico (PU0). No entanto, os métodos que geram os candidatos a partir dos blocos diferem para os modos de mesclagem e AMVP.

[0038] No modo de mesclagem, até quatro candidatos de MV especial podem ser derivados com as ordens mostradas na FIG. 5(a) com números. A ordem é como segue: esquerda (0), acima (1), acima à direita (2), abaixo à esquerda (3) e acima à esquerda (4), conforme mostrado na FIG. 5(a).

[0039] No modo AMVP, os blocos próximos são divididos em dois grupos: um grupo à esquerda incluindo o bloco 0 e 1, e um grupo acima incluindo os blocos 2, 3 e 4 conforme mostrados na FIG. 5(b). Para cada grupo, o potencial candidato em um bloco próximo que se refere à mesma imagem de referência como aquela indicada pelo índice de referência assinalado tem a maior prioridade de ser escolhido para formar o candidato final do grupo. É possível que todos os blocos próximos não contenham um vetor de movimento que aponta para a mesma imagem de referência. Deste modo, se tal candidato não puder ser encontrado, o primeiro candidato disponível será escalado para formar o candidato final, assim as diferenças temporais de distância podem ser compensadas.

[0040] Conforme descrito abaixo com referência à FIG. 6, a Cópia Intrabloco (BC) foi incluída no SCC. Um exemplo de Intra BC é mostrado na FIG. 6, em que a CU atual é predita a partir de um bloco decodificado da imagem/fatia atual. O tamanho do bloco Intra BC atual pode ser tão grande quanto o tamanho de uma CU, que varia de 8x8 a

64x64, embora em algumas aplicações, algumas limitações possam se aplicar.

[0041] Em alguns exemplos, um processo de transformada no espaço de cores in-loop para os sinais residuais pode ser usado para as sequências em formato de crominância 4:4:4. Este método transforma os sinais de erro de predição em formato de crominância RGB/YUV naqueles em um espaço de cores sub-ótimo. Com esta etapa adicional, a correlação entre os componentes coloridos também poderia ser reduzida. A matriz de transformada é derivada dos valores de amostra de pixel para cada unidade de codificação por uma decomposição-valor-singular (SVD). A transformada no espaço de cores se aplica ao erro de predição de ambos os modos intra e inter.

[0042] Em alguns exemplos, quando o processo de transformada no espaço de cores é aplicado ao modo inter, o residual é primeiramente convertido em um domínio diferente com a matriz de transformada derivada. Depois da conversão no espaço de cores, as etapas de codificação convencionais, tais como DCT/DST, quantização e codificação de entropia são realizadas em ordem.

[0043] Em alguns exemplos, quando o processo de transformada no espaço de cores é aplicado ao modo intra, a predição e o bloco atual são primeiramente convertidos em um domínio diferente com a matriz de transformada derivada, respectivamente. Depois da conversão no espaço de cores, o residual entre o bloco atual e o seu preditor também é transformado com DCT/DST, quantizado e codificado em entropia.

[0044] Na operação avançada, a matriz de transformada no espaço de cores é aplicada aos três planos G, B e R como segue:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} G \\ B \\ R \end{bmatrix} = \begin{bmatrix} P \\ Q \\ S \end{bmatrix}$$

[0045] Os valores resultantes são cortados dentro da faixa da especificação HEVC porque, em alguns exemplos, os valores são aumentados em até $\Lambda/3$ vezes. Na operação inversa, uma matriz de transformada no espaço de cores é aplicada aos três componentes P' , Q' e R' como a seguir,

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^t \begin{bmatrix} P' \\ Q' \\ S' \end{bmatrix} = \begin{bmatrix} G' \\ B' \\ R' \end{bmatrix}.$$

[0046] Conforme descrito abaixo com referência à FIG. 7, uma matriz de transformada pode ser derivada a partir dos valores de amostragem de referência. As diferentes amostras de referência podem ser usadas para o intracaso e o intercaso. Para o caso de um bloco intracodificado, um bloco alvo e amostras de referência são mostrados na FIG. 7. Neste exemplo, o bloco alvo consiste de amostras sombreadas de 8×8 e as amostras de referência são amostras pontilhadas e listradas.

[0047] Para o caso de um bloco intercodificado, as amostras de referência para a derivação da matriz são as mesmas que as amostras de referência para a compensação de movimento. De modo a realizar a operação de troca, as amostras de referência no bloco AMP são subamostradas de modo que o número de amostras se torne a energia de dois. Por exemplo, o número de amostras de referência em um bloco 12×16 é reduzido por $2/3$.

[0048] De acordo com as técnicas da revelação atual, o processo de transformada no espaço de cores pode ser aplicado para cada CU. Deste modo, não há necessidade de sinalizar se o processo de transformada é executado ou não. Além disso, ambos os lados codificador e decodificador

derivam da matriz de transformada com o mesmo método para evitar a sobrecarga para sinalizar a matriz de transformada.

[0049] De acordo com as técnicas da revelação atual, os processos de transformada no espaço de cores, tal como as matrizes de transformada no espaço de cores, são usados. Uma tal matriz é a matriz de transformada YCbCr, a qual é:

Avançada:

Inversa:

$$\begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1172 & -0.3942 & 0.5114 \\ 0.5114 & -0.4645 & -0.0469 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \\ = \begin{bmatrix} 1 & 0 & 1.5397 \\ 1 & -0.1831 & -0.4577 \\ 1 & 1.8142 & 0 \end{bmatrix} \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix}$$

[0050] Outra matriz é a matriz de transformada YCoCg, a qual é:

Avançada:

Inversa:

$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 0 & -1/2 \\ -1/4 & 1/2 & -1/4 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \\ \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix}$$

[0051] Outra tal matriz é a matriz YCoCg-R, a qual é uma versão revisada da matriz YCoCg a qual ativa os componentes Co e Cg através de um fator dos dois. Ao usar essa técnica de elevação, a transformada avançada e inversa poderia ser alcançada através das equações que seguem:

Avançada:

$$\begin{aligned} Co &= R - B \\ t &= B + \lfloor Co/2 \rfloor \\ Cg &= G - t \\ Y &= t + \lfloor Cg/2 \rfloor \end{aligned}$$

Inversa:

$$\begin{aligned} t &= Y - \lfloor Cg/2 \rfloor \\ G &= Cg + t \\ B &= t - \lfloor Co/2 \rfloor \\ R &= B + Co \end{aligned}$$

[0052] Nas equações e matrizes acima, as transformações avançadas podem ser realizadas durante o processo de codificação (por exemplo, através de um codificador de vídeo) e as transformações inversas podem ser realizadas no processo de decodificação (por exemplo, através de um decodificador de vídeo).

[0053] Na codificação de vídeo tradicional, supõe-se que as imagens tenham um tom contínuo e sejam espacialmente estáveis. Com base nestas suposições, várias ferramentas foram desenvolvidas (por exemplo, transformada com base em bloco, filtragem, etc.) as quais mostram bom desempenho para vídeos com conteúdo natural. No entanto, em certas aplicações (por exemplo, um computador remoto, monitores de trabalho colaborativos e monitores sem fio), o conteúdo do monitor gerado pelo computador pode ser o conteúdo dominante a ser comprimido. Este tipo de conteúdo tende a ter um discreto tom e apresentar linhas agudas e limites de objeto de alto contraste. A suposição da uniformidade e do tom contínuo pode não mais se aplicar e

assim as técnicas de codificação de vídeo tradicionais podem não funcionar de forma eficiente.

[0054] A codificação de modo palheta de cores pode ser usada para superar as deficiências acima. Exemplos de técnicas de codificação de palhetas de cores são descritos no Pedido de Patente Provisório Norte-Americano No. 61/810,649, depositado em 10 de abril de 2013. Para cada CU, uma palheta pode ser derivada, a qual inclui os valores de pixel mais dominantes na CU atual. O tamanho e os elementos da palheta são os primeiros transmitidos. Depois da transmissão, os pixels na CU são codificados de acordo com uma certa ordem de digitalização. Para cada localização, um elemento de sintaxe, tal como um sinalizador, sinalizador de palheta, é primeiro transmitido para indicar se o valor de pixel está na palheta ("modo de execução") ou não ("modo de pixel").

[0055] No "modo de execução", o índice de palheta pode ser sinalizado, seguido pela "execução". A execução é um elemento de sintaxe que indica o número de pixels consecutivos em uma ordem de digitalização que tenham o mesmo valor do índice de palheta que o do pixel que está atualmente sendo codificado. Se os múltiplos pixels em sucessão imediata na ordem de digitalização têm o mesmo valor do índice de palheta, então o "modo de execução" pode ser indicado pelo elemento de sintaxe, como o sinalizador de palheta. Um valor de contagem pode ser determinado, o qual iguala o número de pixels que sucedem o pixel atual que tem o mesmo valor do índice de palheta que o pixel atual e a execução é igual ao valor de contagem. Nem o sinalizador de palheta ou o índice de palheta precisam ser transmitidos para as posições seguintes que são cobertas pela "execução" à medida que todas elas têm o mesmo valor de pixel. No lado do decodificador, apenas o primeiro valor

do índice de palheta para o pixel atual seria decodificado e o resultado seria duplicado para cada pixel na "execução" de pixels indicados no elemento de sintaxe da "execução".

[0056] No "modo de pixel", o valor de amostragem de pixel é transmitido para esta posição. Se o elemento de sintaxe, tal como um sinalizador de palheta, indica o "modo de pixel", então o valor do índice de palheta é apenas determinado para um pixel que está sendo decodificado.

[0057] As técnicas convencionais podem sofrer de vários problemas. Por exemplo, uma transformada no espaço de cores pode ser executada a qual não leva em conta as características da sequência e a diversidade local. Deste modo, o desempenho da codificação pode ser inadequado. Em outro exemplo, a derivação de uma matriz de transformada pode ser necessária no decodificador, o qual aumenta a complexidade do decodificador significativamente. Além disso, a matriz de transformada pode ser derivada usando tanto os pixels reconstruídos espaciais quanto o preditor das PUs intercodificadas. No entanto, a eficiência da matriz de transformada pode ser reduzida quando o tamanho do PU é relativamente pequeno, a predição não é muito precisa ou os pixels próximos estão indisponíveis. As técnicas desta revelação podem superar um ou mais destes problemas.

[0058] Um codificador de vídeo, tal como um codificador de vídeo 20 ou decodificador de vídeo 30, podem executar as técnicas descritas nesta revelação. Em geral, esta revelação descreve técnicas para a codificação dos blocos de vídeo usando um processo de conversão no espaço de cores. Um codificador de vídeo, tal como codificador de vídeo 20 ou decodificador de vídeo 30, pode determinar se usar a conversão no espaço de cores para codificar os dados de vídeo. Em resposta a determinar se usa a conversão no

espaço de cores para codificar os dados de vídeo, o codificador de vídeo pode quantizar os dados de um primeiro componente de cor dos dados de vídeo usando um primeiro desvio de um primeiro parâmetro de quantização (QP) e quantizar os dados de um segundo componente de cor dos dados de vídeo usando um segundo desvio de um segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio do segundo QP é diferente do primeiro desvio do primeiro QP. O codificador de vídeo também pode determinar se os dados de vídeo foram codificados usando a conversão no espaço de cores. Em resposta a determinar se os dados de vídeo foram codificados usando a conversão no espaço de cores, o codificador de vídeo pode quantizar de forma inversa os dados do primeiro componente de cor dos dados de vídeo usando o primeiro desvio do primeiro QP e quantizar de forma inversa os dados do segundo componente de cor dos dados de vídeo usando o segundo desvio do segundo QP. Em resposta a determinar que os dados de vídeo não foram codificados usando a conversão no espaço de cores, o codificador de vídeo pode quantizar de forma inversa os dados de vídeo usando o primeiro QP e o segundo QP sem o primeiro e o segundo desvio.

[0059] O primeiro QP e o segundo QP podem ser os QPs derivados que correspondem ao primeiro e ao segundo componente de cor que segue o mesmo método de derivação dos blocos que são codificados sem a transformada no espaço de cores. Por exemplo, os valores de QP para o primeiro QP e o segundo QP poderiam ser derivados a partir de elementos de sintaxe sinalizados, tais como `cu_qp_delta_abs`. Um primeiro desvio e um segundo desvio, respectivamente, podem ser então adicionados aos QPs derivados para se obter o primeiro QP e o segundo QP quando os dados de vídeo foram

codificados usando a conversão no espaço de cores. Por exemplo, para um QP (por exemplo, QP_i), um respectivo desvio (por exemplo, $desvio_i$) pode ser adicionado ao QP, tornando o valor de QP resultante igual ao QP original QP mais o respectivo desvio (por exemplo, $QP_{finali} = QP_i + desvio_i$).

[0060] Esta descrição descreve técnicas que podem melhorar o desempenho da codificação de transformada no espaço de cores em sequência e podem reduzir a complexidade do decodificador. A FIG. 1 é um diagrama de blocos que ilustra um sistema de codificação de vídeo e decodificação de vídeo exemplificador 10 que pode usar as técnicas para a codificação do conteúdo do monitor, em que um formato de amostragem de crominância é usado. Conforme mostrado na FIG. 1, o sistema 10 inclui um dispositivo de fonte 12 que provê dados de vídeo codificados para serem decodificados mais tarde pelo dispositivo de destino 14. Em particular, o dispositivo de fonte 12 provê os dados de vídeo ao dispositivo de destino 14 através de uma mídia legível por computador 16. O dispositivo de fonte 12 e o dispositivo de destino 14 podem compreender qualquer um de uma ampla faixa de dispositivos, incluindo computadores, computadores do tipo notebook (isto é, laptop), tablets, caixas set-top, telefones portáteis tais como os assim chamados "smart" phones, assim chamados "smart" pads, televisões, câmeras, dispositivos monitores, reprodutores de mídia digital, consoles de videogame, dispositivo de distribuição de vídeo ou similares. Em alguns casos, o dispositivo de fonte 12 e o dispositivo de destino 14 podem ser equipados para a comunicação sem fio.

[0061] O dispositivo de destino 14 pode receber os dados de vídeo codificados para serem decodificados através de mídia legível por computador 16. A mídia legível

por computador 16 pode compreender qualquer tipo de meio ou dispositivo capaz de mover os dados de vídeo codificados do dispositivo de fonte 12 para o dispositivo de destino 14. Em um exemplo, a mídia legível por computador 16 pode compreender um meio de comunicação para permitir que o dispositivo de fonte 12 transmita dados de vídeo codificados diretamente ao dispositivo de destino 14 em tempo real. Os dados de vídeo codificados podem ser modulados de acordo com um padrão de comunicação, tal como um protocolo de comunicação sem fio e transmitido ao dispositivo de destino 14. O meio de comunicação pode compreender qualquer meio de comunicação sem fio ou com fio, tal como espectro de radiofrequência (RF) ou uma ou mais linhas de transmissão físicas. O meio de comunicação pode fazer parte de uma rede baseada em pacotes, tal como uma rede de área local, uma rede de área ampla ou uma rede global tal como a Internet. O meio de comunicação pode incluir roteadores, comutadores, estações de base ou qualquer outro equipamento que possa ser útil para facilitar a comunicação a partir do dispositivo de fonte 12 para o dispositivo de destino 14.

[0062] Em alguns exemplos, os dados codificados podem ser a saída da interface de saída 22 para um dispositivo de armazenamento. Similarmente, os dados codificados podem ser acessados a partir do dispositivo de armazenamento pela interface de entrada. O dispositivo de armazenamento pode incluir qualquer variedade de meio de armazenamento de dados distribuído ou localmente acessado tal como um disco rígido, discos Blu-ray, DVDs, CD-ROMs, memória flash, memória volátil ou não volátil ou qualquer outro meio de armazenamento digital adequado para armazenar os dados de vídeo codificados. Em um exemplo adicional, o dispositivo de armazenamento pode corresponder a um

servidor de arquivo ou outro dispositivo de armazenamento intermediário que pode armazenar o vídeo codificado gerado pelo dispositivo de fonte 12. O dispositivo de destino 14 pode acessar os dados de vídeo armazenados a partir do dispositivo de armazenamento através de streaming ou download. O servidor de arquivos pode ser qualquer tipo de servidor capaz de armazenar dados de vídeo codificados e transmitir os dados de vídeo codificados ao dispositivo de destino 14. Os servidores de arquivos incluem um servidor da web (por exemplo, para um website), um servidor FTP, dispositivos de armazenamento de rede interligados (NAS) ou um drive de disco local. O dispositivo de destino 14 pode acessar os dados de vídeo codificados através de qualquer conexão de dados padrão, incluindo uma conexão de Internet. Isto pode incluir um canal sem fio (por exemplo, uma conexão Wi-Fi), uma conexão com fio (por exemplo, DSL, modem da TV a cabo, etc.), ou uma combinação de ambos que é adequado para acessar os dados de vídeo codificados armazenados em um servidor de arquivos. A transmissão de dados de vídeo codificados do dispositivo de armazenamento pode ser uma transmissão streaming, uma transmissão por download ou uma combinação das mesmas.

[0063] As técnicas desta descrição não são necessariamente limitadas a aplicações ou ajustes sem fio. As técnicas podem ser aplicadas à codificação de vídeo em suporte a qualquer uma das aplicações de multimídia, tal como transmissões de televisão regulares, transmissões de televisão a cabo, transmissões de televisão por satélite, transmissões de streaming de vídeo pela Internet, tal como streaming dinâmico adaptativo sobre HTTP (DASH), vídeo digital que é codificado em um meio de armazenamento de dados, decodificação de 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 uma via ou duas vias para suportar as aplicações tais como streaming de vídeo, playback de vídeo, transmissão de vídeo e/ou videotelefonia.

[0064] No exemplo da FIG. 1, o dispositivo de fonte 12 inclui fonte de vídeo 18, codificador de vídeo 20 e interface de saída 22. O dispositivo de destino 14 inclui interface de entrada 28, decodificador de vídeo 30 e dispositivo de monitor 32. De acordo com esta revelação, o codificador de vídeo 20 do dispositivo de fonte 12 pode ser configurado para aplicar as técnicas para codificar os blocos de vídeo usando um processo de conversão no espaço de cores. Em outros exemplos, o dispositivo de fonte e o dispositivo de destino podem incluir outros componentes ou disposições. Por exemplo, o dispositivo de fonte 12 pode receber dados de vídeo de uma fonte de vídeo externa 18, tal como uma câmera externa. Da mesma forma, o dispositivo de destino 14 pode ter interface com um dispositivo de monitor externo, em vez de incluir um dispositivo de monitor integrado.

[0065] O sistema ilustrado 10 da FIG. 1 é meramente um exemplo. As técnicas para codificar os blocos de vídeo usando o processo de conversão no espaço de cores podem ser realizadas por qualquer dispositivo de codificação e/ou decodificação de vídeo digital. Embora em geral as técnicas desta revelação sejam realizadas por um dispositivo de codificação de vídeo, as técnicas também podem ser realizadas por um codificador/decodificador de vídeo, tipicamente referido como um "CODEC". Além disso, as técnicas desta revelação também podem ser realizadas por um pré-processador de vídeo. O dispositivo de fonte 12 e o dispositivo de destino 14 são meramente exemplos de tais dispositivos de codificação nos quais o dispositivo de

fonte 12 gera dados de vídeo codificados para a transmissão ao dispositivo de destino 14. Em alguns exemplos, os dispositivos 12, 14 podem operar de uma maneira substancialmente simétrica de modo que cada um dos dispositivos 12, 14 inclua componentes de codificação e decodificação de vídeo. Por conseguinte, o sistema 10 pode suportar transmissão de vídeo de uma via ou duas vias entre os dispositivos de vídeo 12, 14, por exemplo, para o streaming de vídeo, playback de vídeo, transmissão de vídeo ou videotelefonia.

[0066] A fonte de vídeo 18 do dispositivo de fonte 12 pode incluir um dispositivo de captura de vídeo, tal como uma videocâmera, um arquivo de vídeo contendo vídeo capturado anteriormente e/ou uma interface de alimentação de vídeo para receber vídeo a partir de um provedor de conteúdo de vídeo. Como uma alternativa adicional, a fonte de vídeo 18 pode gerar dados com base em gráficos de computador como o vídeo, ou uma combinação de vídeos ao vivo, vídeos de arquivo e vídeo gerado por computador. Em alguns casos, se a fonte de vídeo 18 for uma videocâmera, o dispositivo de fonte 12 e o dispositivo de destino 14 podem formar os assim chamados telefones com câmera ou telefones com videocâmera. Conforme mencionado acima, no entanto, as técnicas descritas nesta revelação podem ser aplicáveis à codificação de vídeo em geral e podem ser aplicadas a aplicações sem fio e/ou com fio. Em cada caso, os vídeos capturados, pré-capturados ou vídeos gerados por computador podem ser codificados pelo codificador de vídeo 20. A informação de vídeo codificada pode então ser retirada pela interface de saída 22 em uma mídia legível por computador 16.

[0067] A mídia legível por computador 16 pode incluir meio transitório, tal como uma transmissão sem fio

ou transmissão por rede com fio ou meio de armazenamento (isto é, meio de armazenamento não transitório), tal como disco rígido, unidade de disco, disco compacto, disco de vídeo digital, disco de Blu-ray ou outra mídia legível por computador. Em alguns exemplos, um servidor de rede (não mostrado) pode receber dados de vídeo codificados a partir do dispositivo de fonte 12 e prover os dados de vídeo codificados ao dispositivo de destino 14, por exemplo, através de uma transmissão em rede. Similarmente, um dispositivo de computação de uma unidade de produção de mídia, tal como uma unidade de gravação de disco, pode receber os dados de vídeo codificados a partir do dispositivo de fonte 12 e produzir um disco contendo os dados de vídeo codificados. Portanto, entende-se que a mídia legível por computador 16 inclui uma ou mais mídias legíveis por computador de várias formas, em vários exemplos.

[0068] A interface de entrada 28 do dispositivo de destino 14 recebe informação a partir da mídia legível por computador 16. A informação da mídia legível por computador 16 pode incluir a informação de sintaxe definida pelo codificador de vídeo 20, o qual também é usado pelo decodificador de vídeo 30, que inclui os elementos de sintaxe que descrevem características e/ou processamento de blocos e outras unidades codificadas, por exemplo, GOPs. O dispositivo de monitor 32 exibe os dados do vídeo decodificado a um usuário e pode compreender qualquer um de uma variedade de dispositivos de monitor tal como um tubo de raios catodo (CRT), um monitor de cristal líquido (LCD), um monitor de plasma, um monitor de diodo de emissão de luz orgânica (OLED) ou outro tipo de dispositivo de monitor.

[0069] O codificador de vídeo 20 e o decodificador de vídeo 30 podem operar de acordo com um

padrão de codificação de vídeo, tal como um padrão de codificação de vídeo de alta eficiência (HEVC) atualmente em desenvolvimento e podem se conformar ao modelo de teste HEVC (HM). Alternadamente, o codificador de vídeo 20 e o decodificador de vídeo 30 podem operar de acordo com outro padrão de proprietário ou indústria, tal como padrão ITU-T H.264, alternadamente referido como MPEG-4, Parte 10, Codificação de vídeo Avançada (AVC), ou extensões de tais padrões. As técnicas desta revelação, no entanto, não se limitam a qualquer padrão de codificação em particular. Outros exemplos de padrões de codificação de vídeo incluem MPEG-2 e ITU-T H.263. Embora não mostrado na FIG. 1, em alguns aspectos, o codificador de vídeo 20 e o decodificador de vídeo 30 podem cada um deles se integrar com um codificador e decodificador de áudio e podem incluir unidades MUX-DEMUX adequadas ou outro hardware e software, para lidar com a codificação de ambos áudio e vídeo em um fluxo de dados comum ou fluxos de dados separados. Caso aplicável, as unidades MUX-DEMUX podem se conformar ao protocolo multiplexador ITU H.223 ou outros protocolos tal como o protocolo datagrama de usuário (UDP).

[0070] O padrão ITU-T H.264/MPEG-4 (AVC) foi formulado pelo ITU-T Grupo de Especialistas em Codificação de vídeo (VCEG) junto com o ISO/IEC Grupo de Especialistas em Imagem e Movimento (MPEG) como o produto de uma parceria coletiva conhecida como a Equipe de Vídeo Conjunta (JVT). Em alguns aspectos, as técnicas descritas nesta revelação podem ser aplicadas aos dispositivos que se conformam em geral ao padrão H.264. O padrão H.264 é descrito na ITU-T H.264, Codificação de Vídeo Avançada para serviços audiovisuais genéricos, pelo Grupo de Estudo ITU-T e com data de março de 2005, o que pode ser referido no presente documento como o padrão H.264 ou a especificação H.264 ou o

padrão ou a especificação H.264/ AVC. A Equipe de Vídeo conjunta (JVT) continua a trabalhar em extensões para H.264/MPEG-4 AVC.

[0071] O codificador de vídeo 20 e o decodificador de vídeo 30 cada um pode ser implementado como qualquer variedade de sistema de circuitos de codificador adequado, tal como um ou mais microprocessadores, processadores de sinal digital (DSPs), circuitos integrados de aplicação específica (ASICs), matriz de portas programável em campo (FPGAs), lógica discreta, software, hardware, firmware ou qualquer combinação dos mesmos. Quando as técnicas são implementadas parcialmente no software, um dispositivo pode armazenar as instruções para o software em uma mídia legível por computador não transitória, adequada e executar as instruções em hardware usando um ou mais processadores para realizar as técnicas desta revelação. Cada codificador de vídeo 20 e decodificador de vídeo 30 pode ser incluído em um ou mais codificadores ou decodificadores, cada um dos quais pode estar integrado como parte de um codificador/decodificador combinado (CODEC) em um respectivo dispositivo.

[0072] A JCT-VC está trabalhando no desenvolvimento do padrão HEVC. Os esforços de padronização HEVC são baseados em um modelo em evolução de um codificador de vídeo referido como o Modelo de Teste HEVC (HM). O HM assume várias capacidades adicionais dos codificadores de vídeo em reação aos dispositivos existentes de acordo com, por exemplo, ITU-T H.264/ AVC. Por exemplo, enquanto o H.264 provê nove modos de codificação intrapredição, o HM pode prover tantas quanto trinta e três modos de codificação intrapredição.

[0073] Em geral, o modelo de trabalho do HM descreve que um quadro de vídeo ou imagem pode ser dividido em uma sequência de treeblocks ou as maiores unidades de codificação (LCU) que incluem ambas as amostras de crominância e luminância. Os dados de sintaxe dentro de um fluxo de dados podem definir um tamanho para o LCU, o qual é a maior unidade de codificação em termos do número de pixels. Uma fatia inclui um número de treeblocks consecutivos na ordem de codificação. Um quadro de vídeo ou imagem pode ser dividido em uma ou mais fatias. Cada treeblock pode ser dividido em unidades de codificação (CUs) de acordo com um quadtree. Em geral, uma estrutura de dados quadtree inclui um nó por CU. Se uma CU é dividida em quatro sub-CUs, o nó que corresponde à CU inclui quatro nós "folhas", cada um dos quais corresponde a uma das sub-CUs.

[0074] Cada nó da estrutura de dados 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 que corresponde ao nó se divide em sub-CUs. Os elementos de sintaxe para uma CU podem ser definidos de maneira recorrente e podem depender se a CU é dividida nas sub-CUs. Se uma CU não se divide mais, ela é referida como folha-CU. Nesta revelação, quatro sub-CUs de uma folha-CU também serão referidos como folhas-CUs mesmo se não houver divisão explícita da folha-CU original. Por exemplo, se uma CU de tamanho 16x16 não é mais dividida, as quatro sub-CUs de 8x8 também serão referidas como folhas-CUs embora a CU de 16x16 não foi nunca dividida.

[0075] Uma CU tem um objetivo similar como um macrobloco do padrão H.264, exceto pelo fato de que uma CU não tem a distinção de tamanho. Por exemplo, uma treeblock pode se dividir em quatro nós filhos (também referidos como sub-CUs) e cada nó filho pode por sua vez ser um nó mãe e

se dividir em outros quatro nós filhos. Um nó filho final, não dividido, referido como um nó folha do quadtree, compreende um nó de codificação, também referido como uma folha-CU. Os dados de sintaxe associados com um fluxo de dados codificado podem definir um número máximo de vezes um treeblock pode se dividir, referido como uma profundidade máxima de CU e também pode definir um tamanho mínimo de nós de codificação. Consequentemente, um fluxo de dados também pode definir uma menor unidade de codificação (SCU). Esta revelação usa o termo "bloco" para se referir a qualquer um de CU, PU ou TU, no contexto de HEVC ou estruturas de dados similares no contexto de outros padrões (por exemplo, macroblocos e sub-blocos dos mesmos em H.264/ AVC).

[0076] Uma CU inclui um nó de codificação e unidades de predição (PUs) e unidades de transformada (TUs) associadas com o nó de codificação. Um tamanho da CU corresponde a um tamanho do nó de codificação e deve ser na forma de um quadrado. O tamanho da CU pode variar de 8x8 pixels até o tamanho do treeblock com um máximo de 64x64 pixels ou mais. Cada CU pode conter uma ou mais PUs e uma ou mais TUs. Os dados de sintaxe associados com uma CU podem descrever, por exemplo, a divisão da CU em uma ou mais PUs. Os modos de divisão podem diferir entre CU, modo codificado de intrapredição ou modo codificado de interpredição. As PUs podem ser divididas em forma não quadrada. Os dados de sintaxe associados com uma CU também podem descrever, por exemplo, a divisão da CU em uma ou mais TUs de acordo com uma quadtree. Uma TU pode ser de forma quadrada ou não quadrada (por exemplo, retangular).

[0077] O padrão HEVC permite as transformações de acordo com as TUs, as quais podem ser diferentes para as diferentes CUs. As TUs são tipicamente dimensionadas com base no tamanho de PUs dentro de uma dada CU definida para

uma LCU dividida, embora isto possa nem sempre ser o caso. As TUs são tipicamente do mesmo tamanho ou menores do que as PUs. Em alguns exemplos, as amostras residuais que correspondem a uma CU podem ser subdivididas em unidades menores usando uma estrutura quadtree conhecida como "quadtree residual" (RQT). Os nós de folha da RQT podem ser referidos como unidades de transformada (TUs). Os valores de diferença de pixel associados com as TUs podem ser transformados para produzir coeficientes de transformada, os quais podem ser quantizados.

[0078] Uma folha-CU pode incluir uma ou mais unidades de predição (PUs). Em geral, uma PU representa uma área espacial que corresponde a todas ou uma parte da CU correspondente e pode incluir dados para recuperar uma amostra de referência para a PU. Além disso, uma PU inclui dados relacionados à predição. Por exemplo, quando o PU é codificado intramodo, os dados para a PU podem ser incluídos em uma quadtree residual (RQT), a qual pode incluir dados que descrevem um modo de intrapredição para uma TU que corresponde à PU. Como outro exemplo, quando o PU é codificado intermodo, a PU pode incluir dados que definem um ou mais vetores de movimento para a PU. Os dados que definem o vetor de movimento para uma PU 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, precisão de quarto de pixel ou precisão de oitavo de pixel), uma imagem de referência à qual o vetor de movimento aponta, e/ou uma lista de imagem de referência (por exemplo, Lista 0, Lista 1, ou Lista C) para o vetor de movimento.

[0079] uma folha-CU que tem uma ou mais PUs também pode incluir uma ou mais unidades de transformada

(TUs). As unidades de transformada podem ser especificadas usando uma RQT (também referida como uma estrutura quadtree de TU), conforme discutida acima. Por exemplo, um sinalizador de divisão pode indicar se uma folha-CU está dividida em quatro unidades de transformada. Depois, cada unidade de transformada pode ser dividida adicionalmente em mais sub-TUs. Quando uma TU não é adicionalmente dividida, ela pode ser referida como uma folha-TU. Geralmente, para codificação intra, todas as folhas-TUs que pertencem a uma folha-CU compartilham o mesmo modo de intrapredição. Isto é, o mesmo modo de intrapredição é geralmente aplicado para calcular os valores preditos para todas as TUs de uma folha-CU. Para a codificação intra, um codificador de vídeo pode calcular um valor residual para cada folha-TU usando o modo de intrapredição, como uma diferença entre a parte da CU que corresponde à TU e o bloco original. Uma TU não está necessariamente limitada ao tamanho de uma PU. Sendo assim, as TUs podem ser maiores ou menores do que uma PU. Para a codificação intra, uma PU pode ser colocada com uma folha-TU correspondente para a mesma CU. Em alguns exemplos, o tamanho máximo de uma folha-TU pode corresponder ao tamanho da folha-CU correspondente.

[0080] Além disso, as TUs de folhas-CUs também podem ser associadas com as respectivas estruturas de dados quadtree, referidas como quadrees residuais (RQTs). Isto é, uma folha-CU pode incluir uma quadtree indicando como a folha-CU é dividida em TUs. O modo de origem de uma quadtree de TU geralmente corresponde a uma folha-CU, enquanto que o nó de origem de uma quadtree de CU geralmente corresponde a um treeblock (or LCU). As TUs da RQT que não são divididas são referidas como folha-TUs. Em geral, esta revelação usa os termos CU e TU para referir a

folha-CU e folha-TU, respectivamente, a menos que observado de outra forma.

[0081] Uma sequência de vídeo tipicamente inclui uma série de quadro de vídeos ou imagens. Um grupo de imagens (GOP) geralmente compreende uma série de uma ou mais videoimagens. Um GOP pode incluir dados de sintaxe em um cabeçalho do GOP, um cabeçalho de uma ou mais das imagens, ou em outro local, que descreve um número de imagens incluídas no GOP. Cada fatia de uma imagem pode incluir dados de sintaxe da fatia que descreve um modo de codificação para a respectiva fatia. O codificador de vídeo 20 tipicamente opera nos blocos de vídeo dentro de fatias de vídeo individuais de modo a 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 fixos ou variáveis e podem diferir em tamanho de acordo com um padrão de codificação especificado.

[0082] Como um exemplo, o HM suporta a predição em vários tamanhos de PU. Assumindo que o tamanho de uma particular CU é $2N \times 2N$, o HM suporta a intrapredição nos tamanhos de PU de $2N \times 2N$ ou $N \times N$ e a interpredição em tamanhos de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$ ou $N \times N$. O HM também suporta a divisão assimétrica para a interpredição em tamanhos de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ e $nR \times 2N$. Na divisão assimétrica, uma direção de uma CU não é dividida, enquanto que a outra direção é dividida em 25% e 75%. A parte da CU que corresponde à divisão de 25% é indicada por um "n" seguido por uma indicação de "Cima", "Baixo" "Esquerda" ou "Direita". Sendo assim, " $2N \times nU$ " se refere a uma CU de $2N \times 2N$ que é dividida horizontalmente com uma PU de $2N \times 0.5N$ na parte inferior e uma PU de $2N \times 1.5N$ na parte posterior.

[0083] Nesta revelação, "NxN" e "N por N" podem ser usados de forma intercambiável para se referirem às dimensões de pixel de um bloco de vídeo em termos de dimensões verticais e horizontais, por exemplo, 16x16 pixels ou 16 por 16 pixels. Em geral, um bloco 16x16 terá 16 pixels em uma direção vertical ($y = 16$) e 16 pixels em uma direção horizontal ($x = 16$). Da mesma forma, um bloco NxN tem geralmente N pixels em uma direção vertical e N pixels em uma direção horizontal, onde N representa um valor inteiro não negativo. Os pixels em um bloco podem ser dispostos em fileiras ou colunas. Além disso, os blocos não precisam necessariamente ter o mesmo número de pixels na direção horizontal como na direção vertical. Por exemplo, os blocos podem compreender NxM pixels, onde M não é necessariamente igual a N.

[0084] Em seguida à codificação intrapreditiva ou interpredictiva usando as PUs de uma CU, o codificador de vídeo 20 pode calcular os dados residuais para as TUs da CU. As PUs podem compreender dados de sintaxe que descrevem um método ou modo de geração de dados de pixel preditivo no domínio espacial (também referido como domínio de pixel) e as TUs podem compreender coeficientes no domínio da transformada seguindo a aplicação de uma transformada, por exemplo, uma transformada discreta de cosseno (DCT), uma transformada de inteiro ou uma transformada conceitualmente similar aos dados de vídeo residuais. Os dados residuais podem corresponder às diferenças de pixel entre os pixels da imagem não codificada e os valores de predição que correspondem às PUs. O codificador de vídeo 20 pode formar as TUs incluindo os dados residuais para a CU e depois transformar as TUs para produzir coeficientes de transformada para a CU.

[0085] Ao seguir qualquer transformada para produzir coeficientes de transformada, o codificador de vídeo 20 pode realizar a quantização dos coeficientes de transformada. A quantização geralmente se refere a um processo no qual os coeficientes de transformada são quantizados para possivelmente reduzir a quantidade de dados usados para representar os coeficientes, provendo compressão adicional. O processo de quantização pode reduzir a profundidade dos bits associada com alguns ou todos os coeficientes. Por exemplo, um valor de n-bit pode ser arredondado para baixo até um valor de m-bit durante a quantização, onde n é maior do que m.

[0086] Após a quantização, o codificador de vídeo pode digitalizar os coeficientes de transformada, produzindo um vetor unidimensional a partir da matriz bidimensional incluindo os coeficientes de transformada quantizada. A digitalização pode ser designada para estabelecer coeficientes de maior energia (e, portanto, menor frequência) na parte frontal da matriz e para estabelecer coeficientes de menor energia (e, portanto, maior frequência) na parte posterior da matriz. Em alguns exemplos, o codificador de vídeo 20 pode usar uma ordem de digitalização predefinida para digitalizar os coeficientes de transformada quantizados para produzir um vetor em série que pode ser codificado em entropia. Em outros exemplos, o codificador de vídeo 20 pode realizar uma digitalização adaptativa. Depois de digitalizar os coeficientes de transformada quantizados para formar um vetor unidimensional, o codificador de vídeo 20 pode codificar em entropia o vetor unidimensional, por exemplo, de acordo com codificação de comprimento variável que se adapta ao contexto (CAVLC), codificação aritmética binária que se adapta ao contexto (CABAC), codificação aritmética binária

que se adapta ao contexto com base na sintaxe (SBAC), Codificação por entropia de divisão de intervalo de probabilidade (PIPE) ou outra metodologia de codificação por entropia. O codificador de vídeo 20 também pode codificar por entropia os elementos de sintaxe associados com os dados de vídeo codificados para uso pelo decodificador de vídeo 30 ao decodificar os dados de vídeo.

[0087] Para realizar CABAC, o codificador de vídeo 20 pode determinar um contexto dentro de um modelo de contexto para um símbolo a ser transmitido. O contexto pode se referir a, por exemplo, se os valores próximos do símbolo são zero ou não. Para realizar CAVLC, o codificador de vídeo 20 pode selecionar um código de comprimento variável para um símbolo a ser transmitido. Os criptogramas em VLC podem ser construídos de modo que códigos relativamente menores correspondam a símbolos mais prováveis, enquanto códigos mais longos correspondam a símbolos menos prováveis. Neste caso, o uso de VLC pode alcançar uma economia de bit, por exemplo, usando criptogramas de comprimento igual para cada símbolo a ser transmitido. A determinação da probabilidade pode estar baseada em um contexto determinado para o símbolo.

[0088] De acordo com as técnicas desta revelação, um codificador de vídeo, tal como codificador de vídeo 20 ou decodificador de vídeo 30, pode determinar se usa a conversão no espaço de cores para codificar os dados de vídeo. Em resposta a determinar se usa a conversão no espaço de cores para codificar os dados de vídeo, o codificador de vídeo pode quantizar os dados de um primeiro componente de cor dos dados de vídeo usando um primeiro desvio de um primeiro parâmetro de quantização (QP) e quantizar os dados de um segundo componente de cor dos dados de vídeo usando um segundo desvio de um segundo QP,

em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio do segundo QP é diferente do primeiro desvio do primeiro QP. O codificador de vídeo também pode determinar se os dados de vídeo foram codificados usando a conversão no espaço de cores. Em resposta a determinar se os dados de vídeo foram codificados usando a conversão no espaço de cores, the codificador de vídeo pode quantizar de forma inversa os dados do primeiro componente de cor dos dados de vídeo usando o primeiro desvio do primeiro QP e quantizar de forma inversa os dados do segundo componente de cor dos dados de vídeo usando o segundo desvio do segundo QP. Em resposta a determinar que os dados de vídeo não foram codificados usando a conversão no espaço de cores, the codificador de vídeo pode quantizar de forma inversa os dados de vídeo usando o primeiro QP e o segundo QP sem o primeiro e o segundo desvio.

[0089] O primeiro QP e o segundo QP podem ser os QPs derivados que correspondem ao primeiro e ao segundo componente de cor que segue o mesmo método de derivação que os blocos os quais são codificados sem a transformada no espaço de cores. Por exemplo, os valores de QP para o primeiro QP e o segundo QP poderiam ser derivados dos elementos de sintaxe sinalizados, tais como $cu_qp_delta_abs$. Um primeiro desvio e um segundo desvio, respectivamente, podem ser então adicionados aos QPs derivados para obter o primeiro QP e o segundo QP quando os dados de vídeo foram codificados usando a conversão no espaço de cores. Por exemplo, para um QP (por exemplo, QP_i), um respectivo desvio (por exemplo, $desvio_i$) pode ser adicionado ao QP, tornando o valor de QP resultante igual ao QP original mais o respectivo desvio (por exemplo, $QP_{finali} = QP_i + desvio_i$).

[0090] Um codificador de vídeo, tal como codificador de vídeo 20 ou decodificador de vídeo 30, podem realizar qualquer uma das técnicas conforme descritas com relação às FIGS. 1-9. Ao usar diferentes parâmetros de quantização para quantizar e quantizar de forma inversa diferentes componentes de cor, um decodificador de vídeo pode ser capaz de decodificar dados de vídeo de forma mais eficiente quando a transformação no espaços de cores é usada, desse modo aumentando a eficiência de codificação total do sistema e reduzindo o consumo de energia.

[0091] Uma variedade de elementos de sintaxe pode ser usada em alinhamento com as técnicas da revelação atual. Estes elementos de sintaxe podem incluir:

TABELA 1

	Descriptor
seq_parameter_set_rbsp() {	
sps_video_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
sps_temporal_id_nesting_flag	u(1)
profile_tier_level(sps_max_sub_layers_minus1)	
...	
vui_parameters_present_flag	u(1)
if(vui_parameters_present_flag)	
vui_parameters()	
sps_extension_present_flag	u(1)
if(sps_extension_present_flag) {	
for(i = 0; i < 1; i++)	
sps_extension_flag[i]	u(1)
sps_extension_7bits	u(7)
if(sps_extension_flag[0]) {	
transform_skip_rotation_enabled_flag	u(1)
transform_skip_context_enabled_flag	u(1)
intra_block_copy_enabled_flag	u(1)
implicit_rdpem_enabled_flag	u(1)
explicit_rdpem_enabled_flag	u(1)
extended_precision_processing_flag	u(1)
intra_smoothing_disabled_flag	u(1)
high_precision_offsets_enabled_flag	u(1)
fast_rice_adaptation_enabled_flag	u(1)
cabac_bypass_alignment_enabled_flag	u(1)
color_transform_enabled_flag (new)	u(1)
}	
if(sps_extension_7bits)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
}	
rbsp_trailing_bits()	
}	

coding_unit(x0, y0, log2CbSize) {	Descriptor
if(transquant_bypass_enabled_flag)	
cu_transquant_bypass_flag	ae(v)
if(slice_type != I)	
cu_skip_flag[x0][y0]	ae(v)
nCbS = (1 << log2CbSize)	
if(cu_skip_flag[x0][y0])	
prediction_unit(x0, y0, nCbS, nCbS)	
else {	
if(intra_block_copy_enabled_flag)	
intra_bc_flag[x0][y0]	ae(v)
if(slice_type != I && !intra_bc_flag[x0][y0])	
pred_mode_flag	ae(v)
if(CuPredMode[x0][y0] != MODE_INTRA intra_bc_flag[x0][y0] log2CbSize == MinCbLog2SizeY)	
part_mode	ae(v)
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(PartMode == PART_2Nx2N && pcm_enabled_flag && !intra_bc_flag[x0][y0] && log2CbSize >= Log2MinIpcmCbSizeY && log2CbSize <= Log2MaxIpcmCbSizeY)	
pcm_flag[x0][y0]	ae(v)
if(pcm_flag[x0][y0]) {	
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
pcm_sample(x0, y0, log2CbSize)	
} else if(intra_bc_flag[x0][y0]) {	
mvd_coding(x0, y0, 2)	
if(PartMode == PART_2NxN)	
mvd_coding(x0, y0 + (nCbS / 2), 2)	
else if(PartMode == PART_Nx2N)	
mvd_coding(x0 + (nCbS / 2), y0, 2)	
else if(PartMode == PART_NxN) {	
mvd_coding(x0 + (nCbS / 2), y0, 2)	
mvd_coding(x0, y0 + (nCbS / 2), 2)	
mvd_coding(x0 + (nCbS / 2), y0 + (nCbS / 2), 2)	
}	
} else {	
pbOffset = (PartMode == PART_NxN) ? (nCbS / 2) : nCbS	
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
prev_intra_luma_pred_flag[x0 + i][y0 + j]	ae(v)
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
if(prev_intra_luma_pred_flag[x0 + i][y0 + j])	
mpm_idx[x0 + i][y0 + j]	ae(v)
else	

rem_intra_luma_pred_mode [x0 + i][y0 + j]	ac(v)
if(ChromaArrayType == 3)	
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
intra_chroma_pred_mode [x0 + i][y0 + j]	ac(v)
else if(ChromaArrayType != 0)	
intra_chroma_pred_mode [x0][y0]	ac(v)
}	
} else {	
...	
}	
if(!pcm_flag[x0][y0]) {	
if(CuPredMode[x0][y0] != MODE_INTRA && !(PartMode == PART_2Nx2N && merge_flag[x0][y0]) (CuPredMode[x0][y0] == MODE_INTRA && intra_bc_flag[x0][y0]))	
rqt_root_cbf	ac(v)
if(rqt_root_cbf) {	
if(color_transform_enabled_flag) (new)	
color_transform_flag [x0][y0] (new)	ac(v)
MaxTrafoDepth = (CuPredMode[x0][y0] == MODE_INTRA ? (max_transform_hierarchy_depth_intra + IntraSplitFlag) : max_transform_hierarchy_depth_inter)	
transform_tree(x0, y0, x0, y0, log2CbSize, 0, 0)	
}	
}	
}	
}	

[0092] Quando o elemento de sintaxe `color_transform_enabled_flag` é igual a 1, o processo de transformada no espaço de cores em loop pode ser invocado no processo de decodificação. Quando o elemento de sintaxe `color_transform_enabled_flag` é igual a 0, a transformada no espaço de cores em loop não pode ser aplicada. Quando o elemento de sintaxe não está presente, o valor do elemento de sintaxe `color_transform_enabled_flag` pode ser deduzido como igual a 0.

[0093] Quando o elemento de sintaxe `color_transform_flag[x0][y0]` é igual a 1, a unidade de codificação em curso pode ser codificada com transformada no espaço de cores. Quando o elemento de sintaxe `color_transform_flag[x0][y0]` é igual a 0, a unidade de codificação em curso é codificada sem transformada no

espaço de cores. Quando o elemento de sintaxe não está presente, o valor do elemento de sintaxe `color_transform_flag` pode ser deduzido como sendo igual a 0. Os índices de matriz `x0` e `y0` pode especificar a localização (`x0`, `y0`) de uma amostra de luminância superior esquerda do bloco de codificação considerado em relação à amostra de luminância do canto superior esquerdo da imagem.

[0094] Em alguns exemplos, o codificador de vídeo, tal como o codificador de vídeo 20 ou decodificador de vídeo 30, pode determinar se deve usar a conversão de espaço de cores para uma unidade de codificação. O codificador de vídeo pode definir um valor de um elemento de sintaxe da unidade de codificação para indicar o uso da conversão de espaço de cores. O codificador de vídeo pode aplicar uma matriz de transformada no espaço de cores a um bloco residual da unidade de codificação. O codificador de vídeo pode decodificar o elemento de sintaxe da unidade de codificação, em que o elemento de sintaxe indica se a unidade de codificação foi codificada usando conversão de espaço de cores. O codificador de vídeo pode determinar se um valor do elemento de sintaxe indica que a unidade de codificação foi codificada usando a conversão de espaço de cores. O codificador de vídeo pode aplicar uma matriz de transformada reversa no espaço de cores a um bloco residual da unidade de codificação em resposta à determinação de que o elemento de sintaxe indica que a unidade de codificação foi codificada usando a conversão de espaço de cores.

[0095] Em um exemplo, para CUs intracodificadas, uma bandeira pode ser diretamente sinalizada. Alternativamente, para CUs intracodificadas, uma bandeira pode ser sinalizada apenas quando a CU não é o modo I-PCM. Em outro exemplo, para CUs intercodificada e / ou CUs intra BC-codificadas, uma bandeira pode ser sinalizada apenas

quando há coeficientes diferentes de zero na CU em curso, isto é, `rqt_root_cbf` é igual a 1. Em outro exemplo, a bandeira não é sinalizada quando a CU é codificada com o modo de paleta. Em todos os exemplos acima, quando a bandeira de transformada da cor decodificada é igual a 0, ou a bandeira não está presente no fluxo de bits para uma CU, o processo de transformada no espaço de cores pode ser ignorado.

[0096] Além disso, uma bandeira em qualquer conjunto de parâmetros de sequência (SPS) imagem / conjunto de parâmetros de imagem (PPS) / cabeçalho de fatia pode ser sinalizada para controlar o uso da transformada no espaço de cores. Quando a bandeira em SPS / PPS / cabeçalho de fatia é igual a 0, a sinalização de bandeira em nível CU pode ser ignorada em todas as CUs na sequência / imagem / fatia correspondente, respectivamente. Alternativamente, a transformada no espaço de cores pode ser realizada em nível de PU ou em nível de TU, em que cada PU / TU tem uma bandeira para indicar o uso da transformada no espaço de cores.

[0097] Em um exemplo, uma matriz de transformada YCoCg modificada pode ser aplicada no processo de codificação e decodificação. A matriz de transformada YCoCg modificada pode ser definida como segue:

Avançada:

$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & -2 \\ -1 & 2 & -1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix},$$

Inversa:

`temp = Y-Cg;`

$G = (Y + C_g + \text{desvio}) \gg 2;$
 $B = (\text{temperatura} - C_o + \text{desvio}) \gg 2;$
 $R = (\text{temp} + C_o + \text{desvio}) \gg 2,$ em que o desvio é igual a 2.

[0098] Em alternativa, a matriz de transformada YCoCg modificada pode ser definida como segue:

Avançada:

$$\begin{bmatrix} Y \\ C_o \\ C_g \end{bmatrix} = \begin{bmatrix} 1/2 & 1 & 1/2 \\ 1 & 0 & -1 \\ -1/2 & 1 & -1/2 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Inversa:

$\text{temp} = Y - C_g;$
 $G = (Y + C_g + \text{desvio}) \gg 1;$
 $B = (\text{temp} - C_o + \text{desvio}) \gg 1;$
 $R = (\text{temp} + C_o + \text{desvio}) \gg 1;$ em que o desvio é igual a 1 ou 0.

[0099] O parâmetro de quantificação correspondente usado por CUs ou blocos que são codificados com transformada de cor pode ser considerado como sendo igual a $(dQP + 2)$, enquanto o usado por CUs ou blocos sem transformada de cor pode ser deduzido como sendo igual a dQP . Simultaneamente, a profundidade de bits pode ser aumentada em 1 em ambos os processos de quantização e transformada.

[0100] Em alternativa, duas etapas de transformada avançada / inversa são aplicadas com a transformada de YCoCg original inalterada além de um processo de normalização. Para transformada avançada, a transformada avançada de YCoCg original é primeiramente

aplicada. Então, para cada componente i , ou seja, Y , Co e Cg , o componente é redefinido para $(i * forward_factor + desvio) \gg BIT_PRECISION$, em que $BIT_PRECISION$ é um número inteiro sem sinal e $forward_factor$ é dependente de $BIT_PRECISION$. Em um exemplo, $forward_factor$ é igual a $1/\sqrt{6} * (1 \ll BITS_TRANS) + 0,5$; $BIT_PRECISION$ é igual a 15, o desvio é igual a $(1 \ll (BIT_PRECISION - 1))$.

[0101] Para a transformada inversa, a transformada inversa de $YCoCg$ original é primeiramente aplicada. Em seguida, para cada componente i , isto é, Y , Co e Cg , o componente é redefinido para $(i * backward_factor + offset) \gg BIT_PRECISION$, em que $BIT_PRECISION$ é o mesmo que o usado na transformada avançada, e o $backward_factor$ é dependente de $BIT_PRECISION$. Em um exemplo, o $backward_factor$ é igual a $\sqrt{6}/4 * (1 \ll BITS_TRANS) + 0,5$; desvio é igual a $(1 \ll (BIT_PRECISION - 1))$.

[0102] Em alternativa / Adicionalmente, a bandeira em nível de CU pode ser sinalizada somente quando pelo menos uma das bandeiras de blocos codificados em três componentes de cor (isto é, cbf_luma , cbf_cb e cbf_cr) é igual a 1. Alternativamente, o resíduo modificado após a transformada no espaço de cores pode ainda ser modificado para assegurar que a gama de resíduos em CUs ou blocos com transformada no espaço de cores é a mesma que a dos resíduos em CUs / blocos sem transformada no espaço de cores. Em um exemplo, é aplicada uma operação de grampo.

[0103] A matriz de transformada no espaço de cores pode ser independente dos pixels reconstruídos. Em vez disso, pode ser dependente do modo de codificação com ou sem perdas. Em um exemplo, quando a CU é codificada com o modo com perdas, $YCoCg$ pode ser aplicada, ao passo que $YCoCg-R$ é aplicada quando a CU é codificada com o modo sem perdas. Além disso, quando $YCoCg-R$ é aplicada, a

profundidade dos componentes Co e Cg pode ser aumentada em 1. Em outro exemplo, a matriz de transformada pode ser dependente dos modos intra, inter / intra BC. Neste caso, uma matriz predefinida para cada modo pode ser especificada tanto no codificador quanto no decodificador e a mesma matriz pode ser usada tanto no codificador quanto no decodificador de acordo com o modo de codificação quando a bandeira em nível CU é 1. Além disso, a bandeira em nível de CU pode ser sinalizada somente quando pelo menos uma das bandeiras de blocos codificados em três componentes de cor (isto é, cbf_luma, cbf_cb e cbf_cr) é igual a 1.

[0104] As técnicas da presente invenção podem também incluir um método de sinalização e derivação da matriz de transformada, tal como em nível de quadro. Em um exemplo, para cada quadro, uma matriz de transformada pode ser derivada com base nas estatísticas através dos componentes de cor na imagem original. A matriz de transformada pode ser codificada em um fluxo de bits e diretamente transmitida ao decodificador. Alternativamente, a predição da matriz de transformada entre quadros consecutivamente codificados pode ser aplicada. Por exemplo, uma bandeira pode ser transmitida para indicar se a matriz é a mesma que a usada no quadro anterior. Alternativamente, uma matriz de transformada pode ser derivada com base em um conjunto de imagens originais e sinalizada no fluxo de bits.

[0105] Em outro exemplo, um conjunto de matrizes de transformada é derivado e sinalizado na sintaxe de alto nível, como PPS. Em um exemplo, a imagem é classificada em várias regiões com base nas características e, para cada região, uma matriz de transformada é derivada. As matrizes resultantes podem ser codificadas e sinalizadas na sintaxe de alto nível. O índice da matriz de transformada

selecionado é ainda sinalizado em nível de CU / TU / PU. Alternativamente, o índice da matriz de transformada selecionada é sinalizado no cabeçalho de fatia ou para cada *tile region*. Alternativamente, a matriz de transformada pode ser derivada em cabeçalho de fatia ou para cada *tile region*.

[0106] Quando a profundidade de bits de componentes de luminância e croma é diferente, a profundidade de bits de todos os componentes de cor pode em primeiro lugar ser modificada para ser a mesma, então a transformada de cor pode ser aplicada em seguida. Em alternativa, o fluxo de bits pode estar em conformidade com uma restrição de que, quando a profundidade de bits de componentes de luminância e croma é diferente no fluxo de bits codificado, a transformada de cor deve ser desativada (ou seja, `color_transform_enabled_flag` igual a 0).

[0107] Diferentes QPs podem ser aplicados a CUs / blocos que são codificados com ou sem transformada de cor, e / ou a profundidade de bits usada em processo de quantização e transformada pode ser modificada por causa dessa transformada de cor não normalizada. Em um exemplo, quando a transformada de YCoCg acima modificada é usada, as seguintes etapas (isto é, Escalonamento, processo de transformação e construção da matriz antes do processo de filtro de desbloqueio) são ainda modificadas com base em `color_transform_flag`.

[0108] Em um exemplo, quando a transformada de YCoCg modificada, tal como descrito acima, é usada, o processo de derivação do parâmetro de quantificação correspondente e processo de transformada inversa são modificados em conformidade. As entradas para este processo podem incluir uma localização de luminância (`xCb`, `yCb`)

especificando a amostra superior esquerda do bloco de codificação de luminância em curso em relação à amostra de luminância superior esquerda da imagem em curso. Neste processo, a variável QpY , o parâmetro de quantização de luminância $Qp'Y$, e os parâmetros de quantização de croma $Qp'Cb$ e $Qp'Cr$ podem ser derivados. A localização de luminância (xQg , yQg) pode especificar a amostra de luminância superior esquerda do grupo de quantização em curso em relação à amostra de luminância superior esquerda da imagem em curso. As posições horizontal e vertical xQg e yQg podem ser iguais a $xCb - (xCb \& ((1 \ll \text{Log2MinCuQpDeltaSize}) - 1))$ e $yCb - (yCb \& ((1 \ll \text{Log2MinCuQpDeltaSize}) - 1))$, respectivamente. O tamanho da luminância de um grupo de quantização, $\text{Log2MinCuQpDeltaSize}$, pode determinar o tamanho da luminância da menor área dentro de um treeblock de codificação que compartilha o mesmo qPY_PRED .

[0109] A variável QpY pode ser derivada da seguinte forma:

$$QpY = ((qPY_PRED + CuQpDeltaVal + 52 + 2 * QpBdOffsetY) \% (52 + QpBdOffsetY)) - QpBdOffsetY$$

[0110] O parâmetro da quantização de luminância $Qp'Y$ pode ser derivado da seguinte forma:

$$Qp'Y = QpY + QpBdOffsetY$$

[0111] Quando o elemento de sintaxe `cromaArrayType` não é igual a 0, o seguinte pode ser aplicado:

As variáveis $qPiCb$ e $qPiCr$ podem ser derivadas da seguinte forma:

$$qPiCb = \text{Clip3}(-QpBdOffsetC, 57, QpY + pps_cb_qp_offset + slice_cb_qp_offset + CuQpAdjValCb)$$

$$qPiCr = \text{Clip3}(-QpBdOffsetC, 57, QpY + pps_cr_qp_offset + slice_cr_qp_offset + CuQpAdjValCr)$$

[0112] Se o elemento de sintaxe `crominanciaArrayType` é igual a 1, as variáveis `qPCb` e `qPCr` podem ser definidas iguais ao valor de `QpC` conforme especificado na tabela abaixo com base no índice `qPi` igual a `qPiCb` e `qPiCr`, respectivamente. Caso contrário, as variáveis `qPCb` e `qPCr` são definidos iguais a $\text{Min}(qPi, 51)$, com base no índice `qPi` igual a `qPiCb` e `qPiCr`, respectivamente.

TABELA 2

qPi	< 30	30	31	32	33	34	35	36	37	38	39	40	41	42	43	> 43
Qp _c	= qP _i	29	30	31	32	33	33	34	34	35	35	36	36	37	37	= qPi - 6

[0113] Os parâmetros de quantização de crominância para os componentes Cb e Cr, $Qp'Cb$ e $Qp'Cr$, podem ser derivados da seguinte forma:

$$Qp'Cb = qPCb + QpBdOffsetC$$

$$Qp'Cr = qPCr + QpBdOffsetC$$

[0114] Quando `color_transform_flag` é igual a 1, o seguinte pode ser aplicado:

$$Qp'Y = (Qp'Y + 8)$$

When ChromaArrayType $\neq 0$, $Qp'Cb = (Qp'cb + 8)$ and $Qp'Cr = (Qp'cr + 8)$

$$BitDepthY = (BitDepthY + 2)$$

$$BitDepthC = (BitDepthC + 2)$$

[0115] As entradas para este processo podem incluir uma localização de luminância (xTbY, yTbY) especificando a amostra superior esquerda do bloco de transformada de luminância em curso em relação à amostra de luminância superior esquerda da imagem atual, um nTbs variável especificando o tamanho do bloco de transformada em curso, um cIdx variável especificando o componente de cor do bloco em curso, e um qP variável especificando o parâmetro de quantização. A emissão deste processo pode incluir a matriz d (nTbS) x (nTbS) de coeficientes de transformada escalonados com elementos d[x][y].

[0116] As variáveis log2TransformRange, bdShift, coeffMin e coeffMax podem ser derivadas da seguinte forma:

Quando color_transform_flag = 1, é aplicável o seguinte:

$$CoeffCTMinY = -(1 << (extended_precision_processing_flag ?$$

$$Max(15, BitDepthY + 6) : 15))$$

$$CoeffCTMinC = -(1 << (extended_precision_processing_flag ?$$

$$Max(15, BitDepthC + 6) : 15))$$

$$CoeffCTMaxY = (1 << (extended_precision_processing_flag ?$$

$$Max(15, BitDepthY + 6) : 15)) - 1$$

$$CoeffCTMaxC = (1 << (extended_precision_processing_flag ?$$

$$Max(15, BitDepthC + 6) : 15)) - 1$$

Se cldx = 0,

$\text{log2TransformRange} = \text{extended_precision_processing_flag} ?$
 $\text{Max}(15, \text{BitDepthY} + 6) : 15$
 $\text{bdShift} = \text{BitDepthY} + \text{Log2}(\text{nTbS}) + 10 - \text{log2TransformRange}$
 $\text{coeffMin} = (\text{color_transform_flag} ? \text{CoeffCTMinY} : \text{CoeffMinY})$
 $\text{coeffMax} = (\text{color_transform_flag} ? \text{CoeffCTMaxY} : \text{CoeffMaxY})$

Caso contrário,

$\text{log2TransformRange} = \text{extended_precision_processing_flag} ?$
 $\text{Max}(15, \text{BitDepthC} + 6) : 15$
 $\text{bdShift} = \text{BitDepthC} + \text{Log2}(\text{nTbS}) + 10 - \text{log2TransformRange}$
 $\text{coeffMin} = (\text{color_transform_flag} ? \text{CoeffCTMinC} : \text{CoeffMinC})$
 $\text{coeffMax} = (\text{color_transform_flag} ? \text{CoeffCTMaxC} : \text{CoeffMaxC})$

[0117] A lista `levelScale[]` pode ser especificada como `levelScale[k] = {40, 45, 51, 57, 64, 72}` com $k = 0..5$.

[0118] Para a derivação dos coeficientes de transformada escalonados $d[x][y]$ com $x = 0..nTbS - 1$, $y = 0..nTbS - 1$, pode ser aplicado o seguinte:

O fator de escalonamento $m[x][y]$ pode ser derivado da seguinte forma:

Se uma ou mais das seguintes condições forem verdadeiras, $m[x][y]$ é definido igual a 16:

$\text{scaling_list_enabled_flag} = 0.$
 $\text{transform_skip_flag}[xTbY][yTbY] = 1 \text{ and } nTbS > 4.$

Caso contrário, é aplicável o seguinte:

$m[x][y] = \text{ScalingFactor}[\text{sizeId}][\text{matrixId}][x][y]$ (8-283)

[0119] Quando o elemento de sintaxe `sizeId` é especificado para o tamanho da matriz de quantização igual a $(nTnS) \times (nTbS)$ e `matrixId` é especificado na Tabela 7-4 para `sizeId`, `CuPredMode[xTbY][yTbY]`, e `cIdx`, respectivamente, então o coeficiente de transformada escalonado $d[x][y]$ pode ser obtido como a seguir:

$$d[x][y] = \text{Clip3}(\text{coeffMin}, \text{coeffMax}, ((\text{TransCoeffLevel}[xTbY][yTbY][cIdx][x][y] * m[x][y] * \text{levelScale}[qP \% 6] << (qP / 6)) + (1 << (bdShift - 1))) >> bdShift)$$

[0120] Quando `color_transform_flag` é igual a 1, o seguinte podem ser aplicado:

$$\text{BitDepthY} = \text{BitDepthY} - 2$$

$$\text{BitDepthC} = \text{BitDepthC} - 2$$

[0121] Em um exemplo, quando a transformada no espaço de cores é aplicada aos modos intra, o resíduo é derivado no domínio do espaço de cores convertido. Isto é, antes de invocar o processo de predição de amostra intra, os pixels vizinhos da CU em curso podem ser primeiro convertidos em outro espaço de cores sub-ótimo com uma transformada de YCoCg ou YCoCg-R avançada modificada. Os pixels vizinhos modificados podem então ser usados para derivar o preditor da CU em curso. O resíduo (isto é, erro de predição) é derivado da CU em curso e os pixels vizinhos da CU em curso no domínio sub-ótimo. O resíduo é codificado da mesma forma que o fluxo de codificação convencional, tal como predição residual entre componentes, transformada, quantização e codificação de entropia. Após a codificação de entropia, os coeficientes de transformada podem ser ainda modificados com a transformada de YCoCg ou YCoCg-R inversa. Os coeficientes de transformada modificados podem

então ser adicionados ao preditor para derivar os pixels reconstruídos da CU em curso. Depois que o processo de reconstrução é invocado, a transformada de YCoCg ou YCoCg-R inversa pode ser aplicada aos pixels vizinhos modificados. Portanto, ao nível do descodificador, antes de invocar o processo de reconstrução, um processo adicional pode ser aplicado, em que a transformada no espaço de cores inversa é aplicada aos coeficientes de transformada derivados.

[0122] Quando a transformada no espaço de cores é aplicada a modos intra, em vez de gerar o resíduo subtraindo o preditor do bloco em curso no domínio da cor convertida, o resíduo pode ser gerado usando os pixels antes da transformada no espaço de cores, seguido pela transformada no espaço de cores. Neste caso, os modos intra de luminância e croma podem ser ajustados ao modo idêntico. Em alternativa, a sinalização do modo de croma pode ser ignorada. Alternativamente, a bandeira da transformada de cor pode ser sinalizada apenas quando os modos intra de luminância e croma são os mesmos.

[0123] Tal como descrito acima, em alguns exemplos, os métodos de codificação sem perdas e com perdas podem compartilhar a mesma matriz, por exemplo, YCoCg-R. Adicionalmente ou em alternativa, quando a transformada de YCoCg-R é aplicada à codificação com perdas, a profundidade de bits de componentes de croma de blocos com transformada de cor pode ser aumentada em 1 em comparação com blocos sem transformada de cor. O componente de luminância QP de blocos com transformada de cor pode ser igual ao de blocos sem transformada de cor menos 4. Em um exemplo, os dois componentes QP de croma de blocos com transformada de cor podem ser iguais ao componente QP de croma / luminância de blocos sem transformada de cor mais 2. Em um exemplo, os componentes de croma Cg

de blocos com transformada de cor podem ser iguais ao componente QP de luminância / croma de blocos sem transformada de cor mais 2. O componente QP de croma Co de blocos com transformada de cor podem ser iguais ao componente QP de luminância / croma de blocos sem transformada de cor mais 3.

[0124] Em tais exemplos, quando YCoCg-R é usado para ambos os modos com perdas e sem perdas, se o bloco em curso é codificado com perdas, a profundidade de bits de luminância e croma pode ser a mesma para aqueles blocos codificados sem habilitar a transformada de cor adaptativa. Neste caso, quando a codificação com perdas é aplicada, após a transformada avançada de YCoCg-R, os componentes Co e Cg podem ser escalonados pelo deslocamento à direita de N bits para reduzir a profundidade de bits aumentada devido à transformada de YCoCg-R avançada e tornar a precisão da profundidade de bits a mesma que a precisão da profundidade de bits do componente Y. Além disso, antes de executar a transformada de YCoCg-R reversa, os componentes Co e Cg podem ser modificados com deslocamento à esquerda de N bits. Em um exemplo, N é igual a um.

[0125] Em um exemplo, pode ser assumido que a profundidade de bits de entrada de componentes de luminância e croma é a mesma. Para o modo de codificação com perdas, operações de deslocamento podem ser aplicadas aos dois componentes de croma (isto é, Cg, Co) após a transformada avançada, e antes da transformada inversa. Em um exemplo, o seguinte processo é aplicado em ordem:

1. YCoCg-R avançada é inalterada:

$$\begin{aligned}
 Co &= R - B \\
 t &= B + \lfloor Co/2 \rfloor \\
 Cg &= G - t \quad , \\
 Y &= t + \lfloor Cg/2 \rfloor
 \end{aligned}$$

2. Se o bloco em curso é codificado em modo com perdas, o seguinte pode ainda ser aplicado:

$$\begin{aligned}
 Cg &= Cg \gg 1; \\
 Co &= Co \gg 1;
 \end{aligned}$$

[0126] Em outro exemplo, um desvio pode ser considerado no processo de deslocamento à direita. Por exemplo, as equações acima destacadas podem ser substituídas por:

$$\begin{aligned}
 Cg &= (Cg + 1) \gg 1; \\
 Co &= (Co + 1) \gg 1;
 \end{aligned}$$

[0127] Antes de invocar a transformada de YCoCg-R reversa, pode ser aplicado o seguinte:

Se o bloco em curso é codificado em modo com perdas, pode ser aplicado o seguinte:

$$\begin{aligned}
 Cg &= Cg \ll 1; \\
 Co &= Co \ll 1;
 \end{aligned}$$

E o YCoCg-R inverso pode permanecer inalterado:

$$\begin{aligned}
 t &= Y - \lfloor Cg/2 \rfloor \\
 G &= Cg + t \\
 B &= t - \lfloor Co/2 \rfloor \\
 R &= B + Co
 \end{aligned}$$

[0128] Em tais exemplos, um processo de modificação residual para blocos de transformada usando a transformada de cor adaptativa pode ser invocado quando ChromaArrayType é igual a 3. As entradas para este processo podem incluir um blkSize variável especificando o tamanho do bloco, uma matriz (blkSize)x(blkSize) de amostras residuais de luminância rY com elementos rY[x][y], uma matriz (blkSize)x(blkSize) de amostras residuais de cromaância rCb com elementos rCb[x][y], e uma matriz (blkSize)x(blkSize) de amostras residuais de cromaância rCr com elementos rCr[x][y]. Saídas para este processo podem incluir uma matriz (blkSize)x(blkSize) modificada rY de amostras residuais de luminância, uma matriz (blkSize)x(blkSize) modificada rCb de amostras residuais de cromaância, e uma matriz (blkSize)x(blkSize) modificada de amostras residuais de cromaância.

[0129] As matrizes (blkSize)x(blkSize) de amostras residuais rY, rCb e rCr com x = 0..blkSize - 1, y = 0..blkSize - 1 podem ser alteradas como a seguir:

Se cu_transquant_bypass_flag = 0, rCb[x][y] = rCb[x][y] << 1 e rCr[x][y] = rCr[x][y] << 1

$$\begin{aligned}
 \text{tmp} &= rY[x][y] - (rCb[x][y] >> 1) \\
 rY[x][y] &= \text{tmp} + rCb[x][y] \\
 rCb[x][y] &= \text{tmp} - (rCr[x][y] >> 1) \\
 rCr[x][y] &= rCb[x][y] + rCr[x][y]
 \end{aligned}$$

[0130] Em alternativa, 'se $\text{cu_transquant_bypass_flag}$ é igual a 0, $\text{r}_{\text{cb}}[\text{x}][\text{y}] = \text{r}_{\text{cb}}[\text{x}][\text{y}] \ll 1$ e $\text{r}_{\text{cr}}[\text{x}][\text{y}] = \text{r}_{\text{cr}}[\text{x}][\text{y}] \ll 1$ ' pode ser substituído por ' $\text{r}_{\text{cb}}[\text{x}][\text{y}] = \text{r}_{\text{cb}}[\text{x}][\text{y}] \ll (1 - \text{cu_transquant_bypass_flag})$ e $\text{r}_{\text{cr}}[\text{x}][\text{y}] = \text{r}_{\text{cr}}[\text{x}][\text{y}] \ll (1 - \text{cu_transquant_bypass_flag})$ '.

[0131] Em outro exemplo de um método de codificação com perdas e um método de codificação sem perdas que partilha de uma matriz de transformada de cor, a profundidade de bits de entrada de componentes de luminância e crominância pode ser diferente. Em tal exemplo, quando as profundidades de bits de entrada de componentes de luminância e crominância são diferentes e YCoCg-R é usado para ambos os modos de codificação com perdas e sem perdas, antes da transformada inversa, pelo menos um dos valores de componente Y, Co, Cg pode ser primeiro deslocado em N bits para fazer todos os três componentes terem a mesma precisão de profundidade de bits. Além disso, mais um bit pode ser deslocado para os dois componentes de crominância (isto é, Co e Cg). Pelo menos um dos valores de predição de todos os três componentes podem ser modificados para ter a mesma precisão de profundidade de bits antes de invocar a transformada de YCoCg-R avançada. E após a transformada avançada, os dois componentes de crominância podem ser deslocados à direita com 1 bit. Para os modos de codificação sem perdas, a profundidade de bits diferente não pode ser considerada e, portanto, a transformada no espaço de cores adaptativa pode ser desabilitada. Em um exemplo, o fluxo de bits pode incluir uma restrição de que, quando a profundidade de bits de componentes de luminância e crominância é diferente no fluxo de bits codificado, a transformada de cor deve ser desabilitada (ou seja, $\text{color_transform_enabled_flag}$ igual a 0).

[0132] Neste exemplo, um processo de modificação residual para blocos de transformada usando a transformada de cor adaptativa pode ser invocado quando `crominânciaArrayType` é igual a 3. As entradas para este processo podem incluir um `blkSize` variável especificando o tamanho do bloco, uma matriz $(\text{blkSize}) \times (\text{blkSize})$ de amostras residuais de luminância r_Y com elementos $r_Y[x][y]$, uma matriz $(\text{blkSize}) \times (\text{blkSize})$ de amostras residuais de crominância r_{Cb} com elementos $r_{Cb}[x][y]$, e uma matriz $(\text{blkSize}) \times (\text{blkSize})$ de amostras residuais de crominância r_{Cr} com elementos $r_{Cr}[x][y]$. Saídas para este processo podem incluir uma matriz $(\text{blkSize}) \times (\text{blkSize})$ modificada r_Y de amostras residuais de luminância, uma matriz $(\text{blkSize}) \times (\text{blkSize})$ modificada de amostras residuais de crominância r_{Cb} , e uma matriz $(\text{blkSize}) \times (\text{blkSize})$ modificada r_{Cr} de amostras residuais de crominância.

[0133] As variáveis `deltaBDy` `deltaBDc` podem ser derivadas da seguinte forma:

$$\begin{aligned} \text{BitDepthMax} &= \text{Max}(\text{BitDepthY}, \text{BitDepthC}) \\ \text{deltaBDY} &= \text{cu_transquant_bypass_flag} ? 0 : \text{BitDepthMax} - \text{BitDepthY} \\ \text{deltaBDC} &= \text{cu_transquant_bypass_flag} ? 0 : \text{BitDepthMax} - \text{BitDepthC} \\ \text{OY} &= \text{cu_transquant_bypass_flag} || (\text{BitDepthMax} == \text{BitDepthY}) ? 0 : 1 \ll (\text{deltaBDY} - 1) \\ \text{OC} &= \text{cu_transquant_bypass_flag} || (\text{BitDepthMax} == \text{BitDepthC}) ? 0 : 1 \ll (\text{deltaBDc} - 1) \end{aligned}$$

[0134] As matrizes $(\text{blkSize}) \times (\text{blkSize})$ de amostras residuais r_Y , r_{Cb} e r_{Cr} com $x = 0..\text{blkSize} - 1$, $y = 0..\text{blkSize} - 1$ podem ser alteradas como segue: amostras residuais $r_Y[x][y]$, $r_{Cb}[x][y]$ e $r_{Cr}[x][y]$ podem ser modificados da seguinte forma:

```

rY[ x ][ y ] = rY[ x ][ y ] << deltaBDY
rCb[ x ][ y ] = rCb[ x ][ y ] << (deltaBDC + 1 - cu_transquant_bypass_flag)
rCr[ x ][ y ] = rCr [ x ][ y ] << (deltaBDC + 1 - cu_transquant_bypass_flag)
tmp = rY[ x ][ y ] - ( rCb[ x ][ y ] >> 1 )
rY[ x ][ y ] = tmp + rCb[ x ][ y ]
rCb[ x ][ y ] = tmp - ( rCr[ x ][ y ] >> 1 )
rCr[ x ][ y ] = rCb[ x ][ y ] + rCr[ x ][ y ]
rY[ x ][ y ] = (rY[ x ][ y ] + OY) >> deltaBDY
rCb[ x ][ y ] = (rCb[ x ][ y ] + Oc) >> deltaBDC
rCr[ x ][ y ] = (rCr [ x ][ y ] + Oc) >> deltaBDC

```

[0135] Nos exemplos acima, a função Max pode ser usada para derivar o valor maior entre duas variáveis. Em alternativa, pelo menos um dos desvios de arredondamento O_y e O_c pode ser definido igual a 0.

[0136] Em alternativa, mesmo para codificação sem perdas, quando a profundidade de bits de componentes de luminância e crominância é diferente, os componentes podem ser deslocados para a mesma precisão de profundidade de bits, se necessário. Em nível do codificador, depois da transformada avançada, pode não haver necessidade de deslocar de volta à precisão de profundidade de bits original. Em outras palavras, o resíduo codificado dos três componentes pode ser da mesma profundidade de bits para ter certeza de que é a codificação sem perdas. Diferente da codificação com perdas, as entradas não podem ser modificadas antes de invocar a transformada de YCoCg-R para trás para codificação sem perdas. O deslocamento à direita pode ainda ser necessário para garantir que as saídas estejam na mesma profundidade de bits que as entradas originais.

[0137] Neste exemplo, um processo de modificação residual para blocos de transformada usando a transformada de cor adaptativa pode ser invocado quando ChromaArrayType é igual a 3. As entradas para este processo podem incluir um blkSize variável especificando o tamanho do bloco, uma matriz (blkSize)x(blkSize) de amostras residuais de luminância r_Y com elementos $r_Y[x][y]$, uma matriz (blkSize)x(blkSize) de amostras residuais de croma r_{Cb} com elementos $r_{Cb}[x][y]$, e uma matriz (blkSize)x(blkSize) de amostras residuais de croma r_{Cr} com elementos $r_{Cr}[x][y]$. Saídas para este processo podem incluir uma matriz (blkSize)x(blkSize) modificada r_Y de amostras residuais de luminância, uma matriz (blkSize)x(blkSize) modificada r_{Cb} de amostras residuais de croma e uma matriz (blkSize)x(blkSize) modificada r_{Cr} de amostras residuais de croma.

[0138] As variáveis ΔBD_Y e ΔBD_C podem ser derivadas da seguinte forma:

$$\text{BitDepthMax} = \text{Max}(\text{BitDepthY}, \text{BitDepthC})$$

$$\Delta BD_Y = \text{BitDepthMax} - \text{BitDepthY}$$

$$\Delta BD_C = \text{BitDepthMax} - \text{BitDepthC}$$

$$OY = (\text{BitDepthMax} == \text{BitDepthY}) ? 0 : 1 \ll (\Delta BD_Y - 1)$$

$$OC = (\text{BitDepthMax} == \text{BitDepthC}) ? 0 : 1 \ll (\Delta BD_C - 1)$$

[0139] As matrizes (blkSize)x(blkSize) de amostras residuais r_Y , r_{Cb} e r_{Cr} com $x = 0 \dots \text{blkSize} - 1$, $y = 0 \dots \text{blkSize} - 1$ podem ser alteradas como segue:

Quando `cu_transquant_bypass_flag` é igual a 0, amostras residuais `rY[x][y]`, `rcb[x][y]` e `rCr[x][y]` podem ser modificadas da seguinte forma:

$$\begin{aligned} rY[x][y] &= rY[x][y] \ll \text{deltaBDY} \\ rCb[x][y] &= rCb[x][y] \ll (\text{deltaBDC} + 1 - \text{cu_transquant_bypass_flag}) \\ rCr[x][y] &= rCr[x][y] \ll (\text{deltaBDC} + 1 - \text{cu_transquant_bypass_flag}) \\ tmp &= rY[x][y] - (rCb[x][y] \gg 1) \\ rY[x][y] &= tmp + rCb[x][y] \\ rCb[x][y] &= tmp - (rCr[x][y] \gg 1) \\ rCr[x][y] &= rCb[x][y] + rCr[x][y] \\ rY[x][y] &= (rY[x][y] + OY) \gg \text{deltaBDY} \\ rCb[x][y] &= (rCb[x][y] + Oc) \gg \text{deltaBDC} \\ rCr[x][y] &= (rCr[x][y] + Oc) \gg \text{deltaBDC} \end{aligned}$$

[0140] Alternativamente, um de O_y e O_c pode ser igual a 0.

[0141] O codificador de vídeo 20 pode ainda enviar dados de sintaxe, como dados de sintaxe baseados em blocos, dados de sintaxe baseados em quadros e dados de sintaxe baseados em GOP, ao decodificador de vídeo 30, por exemplo, em um cabeçalho de quadro, um cabeçalho de bloco, um cabeçalho de fatia ou um cabeçalho GOP. Os dados de sintaxe GOP podem descrever uma série de quadros no respectivo GOP, e os dados de sintaxe de quadro podem indicar um modo de codificação / predição usado para codificar o quadro correspondente.

[0142] O codificador de vídeo 20 e o decodificador de vídeo 30 podem, cada um, ser implementados como qualquer um de uma variedade de circuitos de codificador ou decodificador adequados, conforme o caso, tal como um ou mais microprocessadores, processadores de

sinais digitais (DSPs), circuitos integrados de aplicação específica (ASIC), arranjo de portas programável em campo (FPGAs), circuitos lógicos discretos, software, hardware, firmware ou quaisquer suas combinações. Cada um dentro o codificador de vídeo 20 e o decodificador de vídeo 30 pode ser incluído em um ou mais codificadores ou decodificadores, cada um dos quais pode ser integrado como parte de um codificador / decodificador de vídeo combinado (CODEC). Um dispositivo incluindo o codificador de vídeo 20 e / ou decodificador de vídeo 30 pode compreender um circuito integrado, um microprocessador e / ou um dispositivo de comunicação sem fios, tal como um telefone celular.

[0143] A FIG. 2 é um diagrama em blocos que ilustra um exemplo do codificador de vídeo 20 que pode implementar técnicas para codificação de blocos de vídeo, usando um processo de conversão de espaço de cores. O codificador de vídeo 20 pode realizar intra- e intercodificação de blocos de vídeo dentro de fatias de vídeo. Intracodificação baseia-se em predição espacial para reduzir ou eliminar a redundância espacial em vídeo dentro de um determinado quadro de vídeo ou imagem. Intercodificação baseia-se em predição temporal para reduzir ou eliminar a redundância temporal em vídeo dentro de quadros ou imagens adjacentes de uma sequência de vídeo. Intra-Modo (modo I) pode referir-se a qualquer um dos vários modos de codificação de base espacial. Inter-modos, tal como predição unidirecional (modo P) ou bipredição (modo B), pode se referir a qualquer um dos vários modos de codificação de base temporal.

[0144] Como mostrado na FIG. 2, o codificador de vídeo 20 recebe um bloco de vídeo em curso dentro de um quadro de vídeo a ser codificado. No exemplo da FIG. 2, o

codificador de vídeo 20 inclui a unidade de seleção de modo 40, memória de imagem de referência 64, somador 50, unidade de processamento de transformada 52, unidade de quantização 54 e unidade de codificação de entropia 56. A unidade de seleção de modo 40, por sua vez, inclui a unidade de compensação de movimento 44, unidade de estimativa de movimento 42, unidade de intrapredição 46 e a unidade de partição 48. Para reconstrução do bloco de vídeo, um codificador de vídeo 20 também inclui a unidade de quantização inversa 58, unidade de transformada inversa 60, e somador 62. Um filtro de desbloqueio (não mostrado na FIG. 2) pode também ser incluído para filtrar limites de bloco para remover artefatos de blocagem a partir de vídeo reconstruído. Se desejado, o filtro de desbloqueio tipicamente filtraria a saída do somador 62. Filtros adicionais (em sequência ou pós-sequência) podem também ser usados em adição ao filtro de desbloqueio. Tais filtros não são mostrados por questões de brevidade, mas, se desejado, podem filtrar a saída do somador 50 (como um filtro em sequência).

[0145] Durante o processo de codificação, o codificador de vídeo 20 recebe um quadro ou fatia de vídeo a ser codificada. O quadro ou fatia pode ser dividido em vários blocos de vídeo. A unidade de estimativa de movimento 42 e a unidade de compensação de movimento 44 executam codificação interpreditiva do bloco de vídeo recebido relativo a um ou mais blocos em um ou mais quadros de referência para fornecer predição temporal. A unidade de intrapredição 46 pode alternativamente executar codificação intrapreditiva do bloco de vídeo recebido em relação a um ou mais blocos vizinhos no mesmo quadro ou fatia que o bloco a ser codificado para fornecer uma predição espacial. O codificador de vídeo 20 pode executar várias passagens de

codificação, por exemplo, para selecionar um modo de codificação apropriado para cada bloco de dados de vídeo.

[0146] Além disso, a unidade de partição 48 pode particionar blocos de dados de vídeo em sub-blocos, com base na avaliação dos esquemas de partição anteriores em passes de codificação anteriores. Por exemplo, a unidade de partição 48 pode inicialmente particionar um quadro ou fatia em LCUs, e particionar cada uma das LCUs em sub-CUs com base na análise da taxa de distorção (por exemplo, otimização de taxa de distorção). A unidade de seleção de modo 40 pode ainda produzir uma estrutura de dados quadtree indicativa de partição de uma LCU em sub-CUs. CUs de nó folha da quadtree podem incluir um ou mais PUs e uma ou mais TUs.

[0147] A unidade de seleção de modo 40 pode selecionar um dos modos de codificação, intra ou inter, por exemplo, com base em resultados de erro, e fornece o bloco intra ou intercodificado resultante para o somador 50 para gerar dados de bloco residuais e para o somador 62 para reconstruir o bloco codificado para uso como um quadro de referência. A unidade de seleção de modo 40 também fornece elementos de sintaxe, como vetores de movimento, indicadores intramodo, informações de partição e outras informações de sintaxe, à unidade de codificação de entropia 56.

[0148] A unidade de estimativa de movimento 42 e a unidade de compensação de movimento 44 podem ser altamente integradas, mas são ilustradas separadamente para fins conceituais. A estimativa de movimento, realizado pela unidade de estimativa de movimento 42, é o processo de geração de vetores de movimento, que estimam o movimento para blocos de vídeo. Um vetor de movimento, por exemplo, pode indicar que o deslocamento de uma PU de um bloco de

vídeo dentro de um quadro de vídeo em curso ou da imagem em relação a um bloco preditivo dentro de um quadro de referência (ou outra unidade codificada) em relação ao bloco atual sendo codificado dentro do quadro em curso (ou outra unidade de codificação). Um bloco preditivo é um bloco considerado intimamente correspondente ao bloco a ser codificado, em termos de diferença de pixels, o que pode ser determinado pela soma da diferença absoluta (SAD), soma da diferença quadrática (SSD), ou outros indicadores de diferença. Em alguns exemplos, o codificador de vídeo 20 pode calcular os valores para posições de pixel de subinteiros de imagens de referência armazenadas na memória de imagem de referência 64. Por exemplo, o codificador de vídeo 20 pode interpolar valores de posições de pixel de um quarto, posições de pixel de um oitavo, ou outras posições de pixel fracionárias da imagem de referência. Por conseguinte, a unidade de estimativa de movimento 42 pode executar uma busca de movimento em relação às posições de pixel completas e posições de pixel fracionárias e emitir um vector de movimento com uma precisão de pixel fracionária.

[0149] A unidade de estimativa de movimento 42 calcula um vetor de movimento para uma PU de um bloco de vídeo em uma fatia intercodificada comparando a posição da PU com a posição de um bloco preditivo de uma imagem de referência. A imagem de referência pode ser selecionada a partir de uma primeira lista de imagem de referência (Lista 0) ou uma segunda lista de imagem de referência (Lista 1), cada uma delas identifica uma ou mais imagens de referência armazenadas na memória de imagem de referência 64. A unidade de estimativa de movimento 42 envia o vetor de movimento calculado à unidade de codificação de entropia 56 e à unidade de compensação de movimento 44.

[0150] A compensação de movimento, realizada pela unidade de compensação de movimento 44, pode envolver buscar ou gerar o bloco preditivo com base no vetor de movimento determinado pela unidade de estimativa de movimento 42. Mais uma vez, a unidade de estimativa de movimento 42 e a unidade de compensação de movimento 44 podem ser funcionalmente integradas, em alguns exemplos. Ao receber o vetor de movimento para a PU do bloco de vídeo em curso, a unidade de compensação de movimento 44 pode localizar o bloco preditivo para o qual o vetor de movimento aponta em uma das listas de imagem de referência. O somador 50 forma um bloco de vídeo residual subtraindo os valores de pixel do bloco de predição a partir dos valores de pixel do bloco de vídeo em curso a ser codificado, formando valores de diferença de pixels, como discutido abaixo. Em geral, a unidade de estimativa de movimento 42 executa estimativa de movimento em relação aos componentes de luminância, e a unidade de compensação de movimento 44 usa os vetores de movimento calculados com base nos componentes de luminância para ambos os componentes de crominância e de luminância. A unidade de seleção de modo 40 pode também gerar elementos de sintaxe associados com os blocos de vídeo e a fatia de vídeo para uso pelo decodificador de vídeo 30 na decodificação dos blocos de vídeo da fatia de vídeo.

[0151] A unidade de intrapredição 46 pode intrapredizer um bloco em curso, como uma alternativa para a interpredição realizada pela unidade de estimativa de movimento 42 e a unidade de compensação de movimento 44, conforme descrito acima. Em particular, a unidade de intrapredição 46 pode determinar um modo de intrapredição a usar para codificar um bloco em curso. Em alguns exemplos, a unidade de intrapredição 46 pode codificar um bloco em

curso usando diversos modos de intrapredição, por exemplo, durante passagens de codificação separadas, e a unidade de intrapredição 46 (ou a unidade de seleção de modo 40, em alguns exemplos) pode selecionar um modo de intrapredição adequado a usar a partir dos modos testados.

[0152] Por exemplo, a unidade de intrapredição 46 pode calcular os valores de taxa de distorção usando uma análise da taxa de distorção para os vários modos de intrapredição testados, e selecionar o modo de intrapredição com as melhores características de taxa de distorção entre os modos testados. A análise da taxa de distorção determina geralmente uma quantidade de distorção (ou erro) entre um bloco codificado e um bloco original, não codificado que foi codificado para produzir o bloco codificado, bem como uma taxa de bits (isto é, um número de bits) usada para produzir o bloco codificado. A unidade de intrapredição 46 pode calcular as razões das distorções e taxas para os vários blocos codificados para determinar qual modo de intrapredição apresenta o melhor valor da taxa de distorção para o bloco.

[0153] Após selecionar um modo de intrapredição para um bloco, a unidade de intrapredição 46 pode proporcionar informação indicativa do modo de intrapredição selecionado para o bloco à unidade de codificação de entropia 56. A unidade de codificação de entropia 56 pode codificar a informação indicando o modo de intrapredição selecionado. O codificador de vídeo 20 pode incluir nos dados de configuração de fluxo de bits transmitidos, o que pode incluir uma pluralidade de tabelas de índice do modo de intrapredição e uma pluralidade de tabelas de índice do modo de intrapredição modificado (também referidas como tabelas de mapeamento com criptogramas), definições de contextos de codificação para vários blocos, e indicações

de um modo de intrapredição mais provável, uma tabela de índice de modo de intrapredição, e uma tabela de índice de modo de intrapredição modificado para usar para cada um dos contextos.

[0154] O codificador de vídeo 20 forma um bloco de vídeo residual subtraindo os dados de predição da unidade de seleção de modo 40 a partir do bloco de vídeo original que sendo codificado. O somador 50 representa o componente ou componentes que executam esta operação de subtração. A unidade de processamento de transformada 52 aplica uma transformada, tal como uma transformada de cosseno discreta (DCT) ou uma transformada conceitualmente semelhante, ao bloco residual, produzindo um bloco de vídeo compreendendo valores de coeficientes de transformada residuais. A unidade de processamento de transformada 52 pode realizar outras transformadas que são conceitualmente semelhantes à DCT. Transformadas Wavelet, transformadas de inteiros, transformadas de sub-bandas ou outros tipos de transformadas também poderiam ser usados. Em qualquer caso, a unidade de processamento de transformada 52 aplica a transformada ao bloco residual, produzindo um bloco de coeficientes de transformada residuais. A transformada pode converter a informação residual a partir de um domínio de valores de pixel em um domínio da transformada, tal como um domínio de frequência. A unidade de processamento de transformada 52 pode enviar os coeficientes de transformada resultantes para a unidade de quantização 54. A unidade de quantização 54 quantifica os coeficientes de transformada para reduzir ainda mais a taxa de bits. O processo de quantização pode reduzir a profundidade de bits associada com alguns ou todos os coeficientes. O grau de quantização pode ser modificado por ajuste de um parâmetro de quantização. Em alguns exemplos, a unidade de quantização

54 pode, em seguida, executar uma digitalização da matriz incluindo os coeficientes de transformada quantizados. Alternativamente, a unidade de codificação de entropia 56 pode realizar a digitalização.

[0155] Após a quantização, a unidade de codificação de entropia 56 codifica por entropia os coeficientes de transformada quantizados. Por exemplo, a unidade de codificação de entropia 56 pode realizar codificação de comprimento variável adaptativa ao contexto (CAVLC), codificação aritmética binária adaptativa ao contexto (CABAC), codificação aritmética binária adaptativa ao contexto baseada em sintaxe (SBAC), codificação de entropia por partição de intervalo de probabilidade (PIPE) codificação ou outra técnica de codificação por entropia. No caso de codificação por entropia com base em contexto, o contexto pode ser baseado em blocos vizinhos. Após a codificação por entropia pela unidade de codificação por entropia 56, o fluxo de bits codificado pode ser transmitido para outro dispositivo (por exemplo, o decodificador de vídeo 30) ou arquivado para posterior transmissão ou recuperação.

[0156] De acordo com as técnicas desta invenção, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode executar uma ou mais técnicas da presente invenção. Por exemplo, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode determinar se deve usar a conversão de espaço de cores para codificar os dados de vídeo. A unidade de codificação de entropia 56 pode ainda, em resposta à determinação de usar conversão de espaço de cores para codificar os dados de vídeo, quantizar dados de um primeiro componente de cor dos dados de vídeo usando um primeiro desvio de um primeiro parâmetro de quantização (QP) e quantizar dados de um segundo componente de cor dos

dados de vídeo usando um segundo desvio de um segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio do segundo QP é diferente do primeiro desvio do primeiro QP.

[0157] A unidade de quantização inversa 58 e a unidade de transformada inversa 60 aplicam quantização inversa e transformação inversa, respectivamente, para reconstruir o bloco residual no domínio do pixel, por exemplo, para posterior uso como um bloco de referência. A unidade de compensação de movimento 44 pode calcular um bloco de referência, adicionando o bloco residual a um bloco preditivo de um dos quadros da memória de imagem de referência 64. A unidade de compensação de movimento 44 pode também aplicar um ou mais filtros de interpolação ao bloco residual reconstruído para calcular valores de pixel subinteiros para uso na estimativa de movimento. O somador 62 acrescenta o bloco residual reconstruído ao bloco de predição compensado em movimento produzido pela unidade de compensação de movimento 44 para produzir um bloco de vídeo reconstruído para armazenamento na memória de imagem de referência 64. O bloco de vídeo reconstruído pode ser usado pela unidade de estimativa de movimento 42 e a unidade de compensação de movimento 44 como um bloco de referência para intercodificar um bloco em um quadro de vídeo subsequente.

[0158] Deste modo, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode executar uma ou mais técnicas da presente invenção. Por exemplo, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode determinar se deve usar a conversão de espaço de cores para codificar os dados de vídeo. A unidade de codificação de entropia 56 pode ainda, em resposta à determinação de usar a conversão de espaço de cores para codificar os dados

de vídeo, quantizar dados de um primeiro componente de cor dos dados de vídeo usando um primeiro desvio de um primeiro parâmetro de quantização (QP) e quantizar dados de um segundo componente de cor dos dados de vídeo usando um segundo desvio de um segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio do segundo QP é diferente do primeiro desvio do primeiro QP.

[0159] O primeiro QP e o segundo QP podem ser os QPs derivados correspondentes aos primeiro e segundo componentes de cor seguindo o mesmo método de derivação dos blocos que são codificados sem transformada no espaço de cores. Por exemplo, os valores QP para o primeiro QP e o segundo QP podem ser derivados a partir de elementos de sintaxe sinalizados, tais como `cu_qp_delta_abs`. Um primeiro desvio e um segundo desvio, respectivamente, podem então ser adicionados aos QPs derivados para se obter o primeiro QP e o segundo QP quando os dados de vídeo foram codificados usando a conversão no espaço de cores. Por exemplo, para um QP (por exemplo, QP_i), um respectivo desvio (por exemplo, $desvio_i$) pode ser adicionado ao QP, tornando o valor QP resultante igual a QP original mais o respectivo desvio (por exemplo, $QP_{finali} = QP_i + desvio_i$).

[0160] Em alguns exemplos, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode ainda determinar um primeiro valor para o primeiro QP e um segundo valor para o segundo QP. Em resposta à determinação de não usar a conversão de espaço de cores para codificar os dados de vídeo, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode quantizar os dados do primeiro componente de cor usando o primeiro valor do primeiro QP e quantizar os dados do segundo componente de cor usando o segundo valor do segundo QP. Alternativamente,

em resposta à determinação de usar a conversão no espaço de cores para codificar os dados de vídeo, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode quantizar os dados do primeiro componente de cor usando o primeiro desvio adicionado ao primeiro valor do primeiro QP e quantizar os dados do segundo componente de cor usando o segundo desvio adicionado ao segundo valor do segundo QP, em que o primeiro desvio é diferente do segundo desvio.

[0161] Em alguns exemplos, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode ainda, em resposta à determinação de usar a conversão no espaço de cores para codificar os dados de vídeo, aumentar a profundidade de bits usada para processos de quantização e transformada para os dados de vídeo em dois, em que o primeiro desvio é igual a 8. Em outros exemplos, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode ainda, em resposta à determinação de usar a conversão no espaço de cores para codificar os dados de vídeo, aumentar a profundidade de bits usada para processos de quantização e transformada para os dados de vídeo em 1, em que o primeiro desvio é igual a 2.

[0162] Em alguns exemplos, em resposta à determinação de usar conversão no espaço de cores para codificar os dados de vídeo, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode aplicar um processo de transformada no espaço de cores à unidade de codificação. Em alguns exemplos, o processo de transformada no espaço de cores compreende uma matriz de transformada de YCoCg-R original. Em outros exemplos, o processo de transformada no espaço de cores compreende aplicar uma matriz de transformada de YCoCg modificada. Em tais exemplos, um coeficiente da matriz de transformada de YCoCg modificada pode ser definido como igual a um coeficiente de

uma matriz de transformada de YCoCg original após um desvio à direita de 2 bits com um desvio de arredondamento. A matriz de transformada de YCoCg modificada pode ter um desvio igual a 0, 1 ou 2. Em outros exemplos, o processo de transformada no espaço de cores compreende aplicar uma matriz de YCoCg original. Nesses exemplos, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode ainda aplicar um processo de normalização aos dados de vídeo.

[0163] Em alguns exemplos, o primeiro componente de cor é um componente de luminância (Y). Em alguns exemplos, o primeiro componente de cor é um componente de croma (Cb). Além disso, em alguns exemplos, o segundo componente de cor é um componente de croma (Cr). Em alguns exemplos, pelo menos um dos primeiro desvio e o segundo desvio não é igual a 0.

[0164] Em alguns exemplos, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode determinar usar a conversão no espaço de cores para codificar os dados de vídeo. Ao fazê-lo, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode definir um valor de um elemento de sintaxe dos dados de vídeo para indicar o uso de conversão no espaço de cores em resposta à determinação de usar a conversão no espaço de cores à unidade de codificação. Em alguns exemplos, o elemento de sintaxe compreende uma bandeira de um bit. Em alguns exemplos, o elemento de sintaxe é sinalizado quando a unidade de codificação é codificada usando um modo diferente do modo de modulação por código intrapulso (IPCM). Em outros exemplos, o elemento de sintaxe é sinalizado apenas quando há coeficientes não nulos em uma unidade de transformada da unidade de codificação. Em alguns exemplos, um valor de 1 para o elemento de sintaxe

indica que a unidade de codificação foi codificada usando a conversão no espaço de cores.

[0165] Em alguns exemplos, o elemento de sintaxe pode não ser sinalizado. Por exemplo, o elemento de sintaxe pode não ser sinalizado quando a unidade de codificação é intracodificada e quando um modo de predição de luminância e um modo de predição de crominância de uma unidade de predição dentro da unidade de codificação são diferentes. Em outro exemplo, o elemento de sintaxe pode não ser sinalizado quando a unidade de codificação é codificada com um modo de paleta.

[0166] A FIG. 3 é um diagrama em blocos que ilustra um exemplo do decodificador de vídeo 30 que pode aplicar técnicas para decodificação de blocos de vídeo, alguns dos quais foram codificados usando um processo de conversão no espaço de cores. No exemplo da FIG. 3, o decodificador de vídeo 30 inclui uma unidade de decodificação de entropia 70, unidade de compensação de movimento 72, unidade de intrapredição 74, unidade de quantização inversa 76, unidade de transformada inversa 78, memória de imagem de referência 82 e o somador 80. O decodificador de vídeo 30 pode, em alguns exemplos, executar uma passagem de decodificação geralmente recíproca à passagem de codificação descrita em relação ao codificador de vídeo 20 (FIG. 2). A unidade de compensação de movimento 72 pode gerar dados de predição baseados em vetores de movimento recebidos da unidade de decodificação de entropia 70, enquanto a unidade de intrapredição 74 pode gerar dados de predição com base em indicadores do modo de intrapredição recebidos da unidade de decodificação de entropia 70.

[0167] Durante o processo de decodificação, o decodificador de vídeo 30 recebe um fluxo de bits de vídeo

codificado que representa blocos de vídeo de uma fatia de vídeo codificada e elementos de sintaxe associados a partir do codificador de vídeo 20. A unidade de decodificação de entropia 70 do decodificador de vídeo 30 decodifica por entropia o fluxo de bits para gerar coeficientes quantizados, vetores de movimento ou indicadores do modo de intrapredição e outros elementos de sintaxe. A unidade de decodificação de entropia 70 avança os vetores de movimento e outros elementos de sintaxe para a unidade de compensação de movimento 72. O decodificador de vídeo 30 pode receber os elementos de sintaxe em nível da fatia de vídeo e / ou em nível do bloco de vídeo.

[0168] De acordo com as técnicas da presente descrição, a unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode executar uma ou mais técnicas da presente invenção. Por exemplo, a unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode determinar se os dados de vídeo foram codificados usando a conversão no espaço de cores. A unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode ainda, em resposta à determinação de que os dados de vídeo foram codificados usando a conversão no espaço de cores, quantizar de forma inversa dados de um primeiro componente de cor dos dados de vídeo, usando um primeiro desvio de um primeiro parâmetro de quantização (QP) e dados de quantização inversa de um segundo componente de cor dos dados de vídeo usando um segundo desvio de um segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio do segundo QP é diferente do primeiro desvio do primeiro QP. Em resposta à determinação de que os dados de vídeo não foram codificados usando a conversão no espaço de cores, o codificador de vídeo pode quantizar de forma inversa os

dados de vídeo usando o primeiro QP para o primeiro componente de cor e o segundo QP para o segundo componente de cor, sem o primeiro ou o segundo desvio.

[0169] Quando a fatia de vídeo é codificada como um fatia intracodificada (I), a unidade de intrapredição 74 pode gerar dados de predição para um bloco de vídeo da fatia de vídeo em curso com base em um modo de intrapredição sinalizado e dados de blocos anteriormente decodificados do quadro ou imagem em curso. Quando o quadro de vídeo é codificado como uma fatia intercodificada (ou seja, B, P ou GPB), a unidade de compensação de movimento 72 produz blocos preditivos para um bloco de vídeo da fatia de vídeo em curso com base nos vetores de movimento e outros elementos de sintaxe recebidos da unidade de decodificação de entropia 70. Os blocos preditivos podem ser produzidos a partir de uma das imagens de referência dentro de uma das listas de imagem de referência. O decodificador de vídeo 30 pode construir as listas de quadro de referência, lista 0 e lista 1, usando técnicas de construção padrão com base em imagens de referência armazenadas na memória de imagem de referência 82. A unidade de compensação de movimento 72 determina informações de predição para um bloco de vídeo para a fatia de vídeo em curso ao analisar os vetores de movimento e outros elementos de sintaxe, e usa as informações de predição para produzir os blocos preditivos para o bloco de vídeo em curso sendo decodificado. Por exemplo, a unidade de compensação de movimento 72 usa algum dos elementos de sintaxe recebidos para determinar um modo de predição (por exemplo, intra ou interpredição) usado para codificar os blocos de vídeo da fatia de vídeo, um tipo de fatia de interpredição (por exemplo, fatia B, fatia P ou fatia GPB), informações de construção para uma ou mais das listas de

imagem de referência para a fatia, vetores de movimento para cada bloco de vídeo intercodificado da fatia, o estado de interpriedição para cada bloco de vídeo intercodificado da fatia, e outra informação para decodificar os blocos de vídeo na fatia de vídeo em curso.

[0170] A unidade de compensação de movimento 72 pode também executar a interpolação com base em filtros de interpolação. A unidade de compensação de movimento 72 pode usar filtros de interpolação como usados pelo codificador de vídeo 20 durante a codificação dos blocos de vídeo para calcular os valores interpolados para pixels subinteiros de blocos de referência. Neste caso, a unidade de compensação de movimento 72 pode determinar os filtros de interpolação usados pelo codificador de vídeo 20 a partir dos elementos de sintaxe recebidos e usar os filtros de interpolação para produzir blocos preditivos.

[0171] A unidade de quantização inversa 76 quantifica de forma inversa, isto é, desquantifica, os coeficientes de transformada quantizados fornecidos no fluxo de bits e decodificados pela unidade de decodificação de entropia 70. O processo de quantização inversa pode incluir o uso de um parâmetro de quantização QP_Y calculado pelo decodificador de vídeo 30 para cada bloco de vídeo na fatia de vídeo para determinar um grau de quantização e, do mesmo modo, um grau de quantização inversa que deve ser aplicado.

[0172] A unidade de transformada inversa 78 aplica uma transformada inversa, por exemplo, uma DCT inversa, uma transformada de inteiros inversa ou um processo de transformada inversa conceitualmente semelhante, aos coeficientes de transformada a fim de produzir blocos residuais no domínio do pixel.

[0173] Depois que a unidade de compensação de movimento 72 gera o bloco preditivo para o bloco de vídeo em curso com base nos vetores de movimento e outros elementos de sintaxe, o decodificador de vídeo 30 forma um bloco de vídeo decodificado pela soma dos blocos residuais da unidade de transformada inversa 78 com os blocos preditivos correspondentes gerados pela unidade de compensação de movimento 72. O somador 80 representa o componente ou componentes que executam esta operação somatória. Se desejado, um filtro de desbloqueio pode também ser aplicado para filtrar os blocos decodificados, a fim de remover artefatos de blocagem. Outros filtros em sequência (ou na sequência de codificação ou após a sequência de codificação) também podem ser usados para suavizar as transições de pixel, ou de outra forma melhorar a qualidade do vídeo. Os blocos de vídeo decodificados em um dado quadro ou imagem são então armazenados na memória de imagem de referência 82, que armazena imagens de referência usadas para compensação de movimento subsequente. A memória de imagem de referência 82 também armazena vídeo decodificado para apresentação posterior em um dispositivo de exibição, como o dispositivo de exibição 32 da FIG. 1.

[0174] Deste modo, a unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode executar uma ou mais técnicas da presente invenção. Por exemplo, a unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode determinar se os dados de vídeo foram codificados usando a conversão no espaço de cores. A unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode ainda, em resposta à determinação de que os dados de vídeo foram codificados usando a conversão no espaço de cores, quantizar de forma inversa dados de um

primeiro componente de cor dos dados de vídeo, usando um primeiro desvio de um primeiro parâmetro de quantização (QP) e dados de quantização inversa de um segundo componente de cor dos dados de vídeo usando um segundo desvio de um segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio do segundo QP é diferente do primeiro desvio do primeiro QP. Em resposta à determinação de que os dados de vídeo não foram codificados usando a conversão no espaço de cores, o codificador de vídeo pode quantizar de forma inversa os dados de vídeo usando o primeiro QP e o segundo QP sem os primeiro e o segundo desvios.

[0175] O primeiro QP e o segundo QP podem ser os QPs derivados correspondentes aos primeiro e segundo componentes de cor segundo o mesmo método de derivação que os blocos que são codificados sem transformada no espaço de cores. Por exemplo, os valores de QP para o primeiro QP e o segundo QP podem ser derivados a partir de elementos de sintaxe sinalizados, tal como `cu_qp_delta_abs`. Um primeiro desvio e um segundo desvio, respectivamente, podem então ser adicionados aos QPs derivados para se obter o primeiro QP e o segundo QP quando os dados de vídeo foram codificados usando a conversão no espaço de cores. Por exemplo, para um QP (por exemplo, QP_i), um respectivo desvio (por exemplo, $desvio_i$) pode ser adicionado ao QP, tornando o valor de QP resultante igual ao QP original mais o respectivo desvio (por exemplo, $QP_{finali} = QP_i + desvio_i$).

[0176] Em alguns exemplos, a unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode ainda determinar um primeiro valor para o primeiro QP e um segundo valor para o segundo QP. Em resposta à determinação de não usar a conversão no espaço de cores para codificar os dados de vídeo, a unidade de

decodificação de entropia 70 do decodificador de vídeo 30 pode quantizar de forma inversa os dados do primeiro componente de cor usando o primeiro valor do primeiro QP e quantizar de forma inversa os dados do segundo componente de cor usando o segundo valor do segundo QP. Alternativamente, em resposta à determinação de usar a conversão no espaço de cores para codificar os dados de vídeo, a unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode quantizar de forma inversa os dados do primeiro componente de cor usando o primeiro desvio adicionado ao primeiro valor do primeiro QP e quantizar de forma inversa os dados do segundo componente de cor usando o segundo desvio adicionado ao segundo valor do segundo QP, em que o primeiro desvio é diferente do desvio.

[0177] Em alguns exemplos, uma diferença entre o primeiro QP e o segundo QP é dependente de uma norma de um processo de transformada no espaço de cores usado para codificar os dados de vídeo. Em alguns exemplos, a unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode, quando os dados de vídeo foram codificados usando a conversão no espaço de cores, aumentar a profundidade de bits usada para processos de quantização e transformada para os dados de vídeo em 2, em que o primeiro desvio é igual a 8. Em outros exemplos, a unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode, quando os dados de vídeo foram codificados usando a conversão no espaço de cores, aumentar a profundidade de bits usada para processos de quantização e transformada para os dados de vídeo em 1, em que o primeiro desvio é igual a 2.

[0178] Em alguns exemplos, em resposta à determinação de que os dados de vídeo foram codificados usando a conversão no espaço de cores, a unidade de

decodificação de entropia 70 do decodificador de vídeo 30 pode aplicar um processo de transformada inversa no espaço de cores à unidade de codificação. Em alguns exemplos, a processo de transformada inversa no espaço de cores compreende uma matriz de transformada reversa de YCoCg-R original. Em outros exemplos, o processo de transformada inversa no espaço de cores pode compreender a aplicação de uma matriz de transformada reversa de YCoCg modificada. Em tais exemplos, um coeficiente da matriz de transformada reversa de YCoCg modificada pode ser definido como igual a um coeficiente de uma matriz de transformada reversa de YCoCg original após um deslocamento à direita de 2 bits com um desvio de arredondamento. A matriz de transformada de YCoCg modificada pode ter um desvio igual a 0, 1 ou 2. Em outros exemplos, o processo de transformada inversa no espaço de cores pode compreender aplicar uma matriz de YCoCg original. Em tais exemplos, a unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode ainda aplicar um processo de normalização aos dados de vídeo.

[0179] Em alguns exemplos, o primeiro componente de cor é um componente de luminância (Y). Em alguns exemplos, o primeiro componente de cor é um componente de croma (Cb). Além disso, em alguns exemplos, o segundo componente de cor é um componente de croma (Cr). Em alguns exemplos, pelo menos um dos primeiro desvio e o segundo desvio não é igual a 0.

[0180] Em alguns exemplos, a unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode decodificar um elemento de sintaxe de uma unidade de codificação de dados de vídeo. O elemento de sintaxe pode indicar se a unidade de codificação foi codificada usando a conversão de espaço de cores. Em alguns exemplos, o

elemento de sintaxe pode compreender uma bandeira de um bit. Em alguns exemplos, a unidade de codificação é codificada em um modo diferente do modo de modulação por código intrapulso (IPCM) e o elemento de sintaxe é sinalizado somente para unidades de codificação usando um modo diferente do modo IPCM. Em alguns exemplos, o elemento de sintaxe indica que a unidade de codificação foi codificada usando a conversão no espaço de cores quando não há coeficientes não nulos em uma unidade de transformada da unidade de codificação. Em alguns exemplos, um valor de 1 para o elemento de sintaxe indica que a unidade de codificação foi codificada usando a conversão de espaço de cores. A unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode ainda determinar se um valor do elemento de sintaxe indica que a unidade de codificação foi codificada usando a conversão de espaço de cores. Em resposta à determinação de que o elemento de sintaxe indica que a unidade de codificação foi codificada usando a conversão de espaço de cores, a unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode aplicar o processo de transformada inversa no espaço de cores.

[0181] Em alguns exemplos, o elemento de sintaxe pode indicar que a conversão no espaço de cores não foi usada para codificar a unidade de codificação. Por exemplo, o elemento de sintaxe pode indicar que a conversão no espaço de cores não foi usada para codificar a unidade de codificação, quando a unidade de codificação é intracodificada e quando um modo de predição de luminância e um modo de predição crominância de uma unidade de predição dentro da unidade de codificação são diferentes. Em outro exemplo, o elemento de sintaxe pode indicar que a conversão no espaço de cores não foi usada para codificar a unidade de codificação quando a unidade de codificação é

codificada com um modo de paleta. Nestes exemplos, o elemento de sintaxe pode não estar presente em um fluxo de bits recebido que compreende os dados de vídeo, e decodificar o elemento de sintaxe pode compreender inferir o valor do elemento de sintaxe.

[0182] A FIG. 4 é um diagrama conceitual que ilustra os modos de predição 35 HEVC de acordo com uma ou mais técnicas da presente invenção. No HEVC atual, para o componente de luminância de cada Unidade de Predição (PU), um método de intrapredição é usado com 33 modos de predição angulares (indexados de 2 a 34), o modo DC (indexados com 1) e o modo Planar (indexado com 0), tal como descrito em relação à FIG. 4.

[0183] Além dos 35 modos intra acima, mais um modo, chamado 'I-PCM', é também usado por HEVC. No modo I-PCM, predição, transformada, quantização e codificação de entropia são ignorados, enquanto as amostras de predição são codificadas por um número predefinido de bits. O principal objetivo do modo I-PCM é lidar com a situação quando o sinal não pode ser eficientemente codificado por outros modos.

[0184] A FIG. 5 é um diagrama conceitual que ilustra candidatos ao vetor de movimento vizinho espacial para os modos de predição de vetor de movimento avançado (AMVP) de acordo com uma ou mais das técnicas da presente invenção. Conforme descrito em relação à FIG. 5, os candidatos MV espaciais são derivados dos blocos vizinhos mostrados na FIG. 5 para uma PU específica (PU0), embora os métodos de geração dos candidatos dos blocos sejam diferentes para os modos de mesclagem e AMVP.

[0185] No modo de mesclagem, até quatro candidatos MV espaciais podem ser derivados com as ordens mostradas na FIG. 5(a) com números, e a ordem é a seguinte:

esquerda (0), acima (1), acima à direita (2), embaixo à esquerda (3) e acima à esquerda (4), como mostrado na FIG. 5(a).

[0186] No modo de AMVP, os blocos vizinhos são divididos em dois grupos: o grupo à esquerda que consiste no bloco 0 e 1, e o grupo acima que consiste nos blocos 2, 3 e 4, como mostrado na FIG. 5(b). Para cada grupo, o potencial candidato em um bloco vizinho referindo-se à mesma imagem de referência que a indicada pelo índice de referência sinalizado tem a prioridade mais alta a ser escolhida para formar um candidato final do grupo. É possível que todos os blocos vizinhos não contenham um vetor de movimento que aponta para a mesma imagem de referência. Portanto, se esse candidato não pode ser encontrado, o primeiro candidato disponível será dimensionado para formar o candidato final, assim as diferenças de distância temporais podem ser compensadas.

[0187] A FIG. 6 é um diagrama conceitual que ilustra um exemplo de cópia intrabloco (BC) de acordo com uma ou mais técnicas da invenção em curso. Conforme descrito em relação à FIG. 6, a cópia intrabloco (BC) exemplificativa foi incluída em RExt. Um exemplo de intra BC é mostrado como na FIG. 6, em que a CU em curso é predita a partir de um bloco já decodificado da imagem / fatia em curso. O tamanho do bloco Intra BC em curso pode ser tão grande quanto o tamanho da CU, que varia de 8x8 a 64x64, apesar de algumas aplicações, outras restrições podem ser aplicadas adicionalmente.

[0188] A FIG. 7 é um diagrama conceitual que ilustra um exemplo de um bloco alvo e amostra de referência para um bloco 8x8 intra, de acordo com uma ou mais técnicas da presente invenção. Tal como descrito abaixo em relação à FIG. 7, uma matriz de transformada é derivada a partir dos

valores de amostra de referência. Diferentes amostras de referência são usadas para o caso intra e o caso inter. Para o caso de intrabloco, bloco alvo e amostras de referência são apresentados na FIG. 7. Nesta figura, o bloco alvo consiste em amostras hachuradas transversalmente 8x8 e as referências são amostras listradas e pontilhadas.

[0189] Para o caso de interblobo, amostras de referência para a derivação de matriz são a mesma que para a compensação de movimento. A fim de realizar a operação de deslocamento, as amostras de referência no bloco AMP é subamostrada de tal modo que o número de amostras se torna a potência-de-dois. Por exemplo, o número de amostras de referência de um bloco 12x16 é reduzida a 2/3.

[0190] De acordo com algumas técnicas da presente invenção, pode ser aplicado o processo de transformada no espaço de cores. Em tais exemplos, não há necessidade de sinalizar se o processo de transformada é invocado ou não. Além disso, ambos os lados do codificador e decodificador podem derivar a matriz de transformada com o mesmo método, para evitar a sobrecarga para sinalização da matriz de transformada.

[0191] A FIG. 8 é um fluxograma que ilustra um exemplo de método para codificação de um bloco em curso. O bloco em curso pode compreender uma CU em curso ou uma parte da CU em curso. Embora descrito em relação ao codificador de vídeo 20 (FIGS. 1 e 2), deve ser compreendido que outros dispositivos podem ser configurados para realizar um método semelhante ao da FIG. 8.

[0192] Neste exemplo, o codificador de vídeo 20 inicialmente prediz o bloco em curso (150). Por exemplo, o codificador de vídeo 20 pode calcular uma ou mais unidades de predição (PUs) para o bloco em curso. O codificador de vídeo 20 pode, então, calcular um bloco residual para o

bloco em curso, por exemplo, para produzir uma unidade de transformada (TU) (152). Para calcular o bloco residual, o codificador de vídeo 20 pode calcular uma diferença entre o bloco original, não codificado e o bloco predito para o bloco em curso. A unidade de codificação de entropia 56 do codificador de vídeo 20 pode determinar se deve usar a conversão no espaço de cores para codificar os dados de vídeo e, quando a conversão de espaço de cores estiver habilitada, é aplicada uma transformada no espaço de cores (154). O codificador de vídeo 20 pode, em seguida, transformar os coeficientes do bloco residual (156). A unidade de codificação de entropia 56 pode ainda, em resposta à determinação de usar conversão no espaço de cores para codificar os dados de vídeo, quantizar os dados de um primeiro componente de cor dos dados de vídeo, usando um primeiro desvio de um primeiro parâmetro de quantização (QP) (158) e quantizar os dados de um segundo componente de cor dos dados de vídeo usando um segundo desvio de um segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio do segundo QP é diferente do primeiro desvio do primeiro QP (160). Em seguida, o codificador de vídeo 20 pode digitalizar coeficientes de transformada quantizados do bloco residual (162). Durante a digitalização, ou após a digitalização, o codificador de vídeo 20 pode codificar por entropia os coeficientes (164). Por exemplo, o codificador de vídeo 20 pode codificar os coeficientes usando CAVLC ou CABAC. O codificador de vídeo 20 pode, em seguida, emitir os dados codificados por entropia do bloco (166).

[0193] Em alguns exemplos, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode ainda determinar um primeiro valor para o primeiro QP e um segundo valor para o segundo QP. Em resposta à determinação

de não usar a conversão no espaço de cores para codificar os dados de vídeo, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode quantizar os dados do primeiro componente de cor usando o primeiro valor do primeiro QP e quantizar os dados do segundo componente de cor usando o segundo valor do segundo QP. Alternativamente, em resposta à determinação de usar a conversão no espaço de cores para codificar os dados de vídeo, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode quantizar os dados do primeiro componente de cor usando o primeiro desvio adicionado ao primeiro valor do primeiro QP e quantizar os dados do segundo componente de cor usando o segundo desvio adicionado ao segundo valor do segundo QP, em que o primeiro desvio é diferente do segundo desvio.

[0194] Em alguns exemplos, a diferença entre o primeiro QP e o segundo QP é dependente de uma norma do processo de transformada no espaço de cores usado para codificar os dados de vídeo. Em alguns exemplos, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode ainda, em resposta à determinação de usar conversão no espaço de cores para codificar os dados de vídeo, aumentar a profundidade de bits usada para processos de quantização e transformada para os dados de vídeo em 2, em que o primeiro desvio é igual a 8. Em outros exemplos, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode ainda, em resposta à determinação de usar a conversão no espaço de cores para codificar os dados de vídeo, aumentar a profundidade de bits usada para processos de quantização e transformada para os dados de vídeo em 1, em que o primeiro desvio é igual a 2.

[0195] Em alguns exemplos, em resposta à determinação de usar a conversão no espaço de cores para codificar os dados de vídeo, a unidade de codificação de

entropia 56 do codificador de vídeo 20 pode aplicar um processo de transformada no espaço de cores à unidade de codificação. Em alguns exemplos, o processo de transformada no espaço de cores compreende uma matriz de transformada de YCoCg-R original. Em outros exemplos, o processo de transformada no espaço de cores compreende aplicar uma matriz de transformada de YCoCg modificada. Em tais exemplos, um coeficiente da matriz de transformada de YCoCg modificada pode ser definido como igual um coeficiente de uma matriz de transformada de YCoCg original após um deslocamento à direita de 2 bits com um desvio de arredondamento. A matriz de transformada de YCoCg modificada pode ter um desvio igual a 0, 1 ou 2. Em outros exemplos, o processo de transformada no espaço de cores compreende aplicar uma matriz de YCoCg original. Em tais exemplos, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode ainda aplicar um processo de normalização aos dados de vídeo.

[0196] Em alguns exemplos, o primeiro componente de cor é um componente de luminância (Y). Em alguns exemplos, o primeiro componente de cor é um componente de croma (Cb). Além disso, em alguns exemplos, o segundo componente de cor é um componente de croma (Cr). Em alguns exemplos, pelo menos um dos primeiro desvio e o segundo desvio não é igual a 0.

[0197] Em alguns exemplos, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode determinar usar a conversão no espaço de cores para codificar os dados de vídeo. Ao fazê-lo, a unidade de codificação de entropia 56 do codificador de vídeo 20 pode definir um valor de um elemento de sintaxe dos dados de vídeo para indicar o uso de conversão no espaço de cores em resposta à determinação de usar a conversão no espaço de

cores à unidade de codificação. Em alguns exemplos, o elemento de sintaxe compreende uma bandeira de um bit. Em alguns exemplos, o elemento de sintaxe é sinalizado quando a unidade de codificação é codificada usando um modo diferente do modo de modulação por código intrapulso (IPCM). Em outros exemplos, o elemento de sintaxe é sinalizado apenas quando há coeficientes não nulos em uma unidade de transformada da unidade de codificação. Em alguns exemplos, um valor de 1 para o elemento de sintaxe indica que a unidade de codificação foi codificada usando a conversão no espaço de cores.

[0198] Em alguns exemplos, o elemento de sintaxe pode não ser sinalizado. Por exemplo, o elemento de sintaxe pode não ser sinalizado quando a unidade de codificação é intracodificada e quando um modo de predição de luminância e um modo de predição de crominância de uma unidade de predição dentro da unidade de codificação são diferentes. Em outro exemplo, o elemento de sintaxe pode não ser sinalizado quando a unidade de codificação é codificada com um modo de paleta. Nestes exemplos, o elemento de sintaxe pode não estar presente em um fluxo de bits recebido que compreende os dados de vídeo, e decodificar o elemento de sintaxe pode compreender inferir o valor do elemento de sintaxe.

[0199] A FIG. 9 é um fluxograma que ilustra um exemplo de método para decodificação de um bloco em curso de dados de vídeo. O bloco em curso pode compreender uma CU em curso ou uma parte da CU em curso. Embora descrito em relação ao decodificador de vídeo 30 (FIGS. 1 e 3), deve ser compreendido que outros dispositivos podem ser configurados para realizar um método semelhante ao da FIG. 9.

[0200] O decodificador de vídeo 30 pode prever o bloco em curso (200), por exemplo, usando um modo de intra ou interpredição para calcular um bloco predito para o bloco em curso. O decodificador de vídeo 30 também pode receber dados codificados por entropia para o bloco em curso, como dados codificados por entropia para coeficientes de um bloco residual correspondente ao bloco em curso (202). O decodificador de vídeo 30 pode decodificar por entropia os dados codificados por entropia para reproduzir coeficientes do bloco residual (204). O decodificador de vídeo 30 pode então digitalizar de forma inversa os coeficientes reproduzidos (206), para criar um bloco de coeficientes de transformada quantizados. A unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode determinar se os dados de vídeo foram codificados usando a conversão no espaço de cores (208). A unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode ainda, em resposta à determinação de que os dados de vídeo foram codificados usando a conversão no espaço de cores, dados de quantização inversa de um primeiro componente de cor dos dados de vídeo, usando um primeiro desvio de um primeiro parâmetro de quantização (QP) (210) e os dados de quantização inversa de um segundo componente de cor dos dados de vídeo usando um segundo desvio de um segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio do segundo QP é diferente do primeiro desvio do primeiro QP (212). O decodificador de vídeo 30 pode, então, transformar de forma inversa os coeficientes (214) e, quando a conversão no espaço de cores estiver habilitada, uma transformada inversa no espaço de cores pode ser ainda aplicada. O decodificador de vídeo 30 pode, por fim, decodificar o bloco em curso, combinando o bloco

predito e o bloco residual (216). Em resposta à determinação de que os dados de vídeo não foram codificados usando a conversão no espaço de cores, o codificador de vídeo pode quantizar de forma inversa os dados de vídeo do primeiro QP para o primeiro componente de cor e o segundo QP para o segundo componente de cor, sem o primeiro ou segundo desvio.

[0201] Em alguns exemplos, a unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode ainda determinar um primeiro valor para o primeiro QP e um segundo valor para o segundo QP. Em resposta à determinação de não usar a conversão no espaço de cores para codificar os dados de vídeo, a unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode quantizar de forma inversa os dados do primeiro componente de cor usando o primeiro valor do primeiro QP e quantizar de forma inversa os dados do segundo componente de cor usando o segundo valor do segundo QP. Alternativamente, em resposta à determinação de usar a conversão no espaço de cores para codificar os dados de vídeo, a unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode quantizar de forma inversa os dados do primeiro componente de cor usando o primeiro desvio adicionado ao primeiro valor do primeiro QP e quantizar de forma inversa os dados do segundo componente de cor usando o segundo desvio adicionado ao segundo valor do segundo QP, em que o primeiro desvio é diferente do segundo desvio.

[0202] Em alguns exemplos, a diferença entre o primeiro QP e o segundo QP é dependente de uma norma de um processo de transformada no espaço de cores usado para codificar os dados de vídeo. Em alguns exemplos, a unidade de decodificação de entropia 70 do decodificador de vídeo

30 pode, quando os dados de vídeo foram codificados usando a conversão no espaço de cores, aumentar a profundidade de bits usada para processos de quantização e transformada para os dados de vídeo em 2, em que o primeiro desvio é igual 8. Em outros exemplos, a unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode, quando os dados de vídeo foram codificados usando a conversão no espaço de cores, aumentar a profundidade de bits usada para processos de quantificação e transformada para os dados de vídeo em 1, em que o primeiro desvio é igual a 2.

[0203] Em alguns exemplos, em resposta à determinação de que os dados de vídeo foram codificados usando a conversão no espaço de cores, a unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode aplicar um processo de transformada inversa no espaço de cores à unidade de codificação. Em alguns exemplos, o processo de transformada inversa no espaço de cores compreende uma matriz de transformada reversa de YCoCg-R original. Em outros exemplos, o processo de transformada inverso no espaço de cores pode compreender aplicar uma matriz de transformada reversa de YCoCg modificada. Em tais exemplos, um coeficiente da matriz de transformada reversa de YCoCg modificada pode ser definido como igual um coeficiente de uma matriz de transformada reversa de YCoCg original após um deslocamento à direita de 2 bits com um desvio de arredondamento. A matriz de transformada de YCoCg modificada pode ter um desvio igual a 0, 1 ou 2. Em outros exemplos, o processo de transformada inversa no espaço de cores pode compreender aplicar uma matriz de YCoCg original. Em tais exemplos, a unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode aplicar ainda um processo de normalização aos dados de vídeo.

[0204] Em alguns exemplos, o primeiro componente de cor é um componente de luminância (Y). Em alguns exemplos, o primeiro componente de cor é um componente de crominância (Cb). Além disso, em alguns exemplos, o segundo componente de cor é um componente de crominância (Cr). Em alguns exemplos, pelo menos um dos primeiro desvio e o segundo desvio não é igual a 0.

[0205] Em alguns exemplos, a unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode decodificar um elemento de sintaxe de uma unidade de codificação de dados de vídeo. O elemento de sintaxe pode indicar se a unidade de codificação foi codificada usando a conversão no espaço de cores. Em alguns exemplos, o elemento de sintaxe pode compreender uma bandeira de um bit. Em alguns exemplos, a unidade de codificação é codificada em um modo diferente do modo de modulação por código intrapulso (IPCM) e o elemento de sintaxe é sinalizado somente para unidades de codificação que usam um modo diferente do modo de IPCM. Em alguns exemplos, o elemento de sintaxe indica que a unidade de codificação foi codificada usando a conversão no espaço de cores quando há coeficientes não nulos em uma unidade de transformada da unidade de codificação. Em alguns exemplos, um valor de 1 para o elemento de sintaxe indica que a unidade de codificação foi codificada usando a conversão no espaço de cores. A unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode ainda determinar se um valor do elemento de sintaxe indica que a unidade de codificação foi codificada usando a conversão no espaço de cores. Em resposta à determinação de que o elemento de sintaxe indica que a unidade de codificação foi codificada usando a conversão no espaço de cores, a unidade de decodificação de

entropia 70 do decodificador de vídeo 30 pode aplicar o processo de transformada inversa no espaço de cores.

[0206] Em alguns exemplos, o elemento de sintaxe pode indicar que a conversão no espaço de cores não foi usada para codificar a unidade de codificação. Por exemplo, o elemento de sintaxe pode indicar que a conversão no espaço de cores não foi usada para codificar a unidade de codificação quando a unidade de codificação é intracodificada e quando um modo de predição de luminância e um modo de predição de crominânciaa de uma unidade de predição dentro da unidade de codificação são diferentes. Em outro exemplo, o elemento de sintaxe pode indicar que a conversão no espaço de cores não foi usada para codificar a unidade de codificação quando a unidade de codificação é codificada com um modo de paleta. Nestes exemplos, o fluxo de bits recebido pode não incluir o elemento de sintaxe.

[0207] Deve ser reconhecido que, dependendo do exemplo, certos atos ou eventos de qualquer das técnicas aqui descritas podem ser realizados em uma sequência diferente, podem ser adicionados, mesclados ou deixados de fora (por exemplo, nem todos os atos ou eventos descritos são necessários para a prática das técnicas). Além disso, em certos exemplos, atos ou eventos podem ser realizados simultaneamente, por exemplo, através do processamento *multi-threaded*, processamento de interrupção ou vários processadores, em vez de sequencialmente.

[0208] Em um ou mais exemplos, as funções descritas podem ser implementadas em hardware, software, firmware, ou qualquer combinação dos mesmos. Se implementadas em software, as funções podem ser armazenadas ou transmitidas como uma ou mais instruções ou código em um meio de leitura por computador e executadas por uma unidade de processamento com base em hardware. Meios de leitura por

computador podem incluir meios de armazenamento legíveis por computador, o que corresponde a um meio tangível, como meios de armazenamento de dados ou meios de comunicação, incluindo qualquer meio que facilite a transferência de um programa de computador de um lugar para outro, por exemplo, de acordo com um protocolo de comunicação. Deste modo, meios legíveis por computador podem geralmente corresponder a (1) um meio de armazenamento legível por computador tangível que é não transitório ou (2) um meio de comunicação, tal como um sinal ou onda portadora. Meios de armazenamento de dados podem ser quaisquer meios disponíveis que possam 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 nesta invenção. Um produto de programa de computador pode incluir um meio legível por computador.

[0209] A título de exemplo, e não como limitação, tais meios de armazenamento legíveis por computador podem compreender RAM, ROM, EEPROM, CD-ROM ou outro armazenamento em disco óptico, armazenamento em disco magnético, ou outros dispositivos de armazenamento magnéticos, memória flash, ou qualquer outro meio que possa ser usado para armazenar o código de programa desejado sob a forma de instruções ou estruturas de dados, e que possa ser acessado por um computador. Além disso, qualquer ligação é adequadamente denominada um meio legível por computador. Por exemplo, se instruções são transmitidas a partir de um site, servidor ou outra fonte remota usando um cabo coaxial, cabo de fibra óptica, par trançado, linha de assinante digital (DSL) ou tecnologias sem fios, tal como infravermelho, rádio e micro-ondas, então o cabo coaxial, cabo de fibra óptica, par torcido, DSL, ou tecnologias sem

fios, tais como infravermelho, rádio e micro-ondas estão incluídos na definição de meio. Deve ser entendido, no entanto, que os meios de armazenamento legíveis por computador e mídias de armazenamento de dados não incluem conexões, ondas portadoras, sinais ou outros meios transitórios, mas são direcionados para meios de armazenamento tangível, não transitório. Disco (disk) e disco (disc), como aqui usados, incluem disco compacto (CD), disco laser, disco óptico, disco versátil digital (DVD), disquete e disco Blu-ray, em que discos (disks) geralmente reproduzem dados magneticamente, ao passo que discos (discs) reproduzem dados opticamente com lasers. Combinações dos anteriores também devem ser incluídas dentro do âmbito dos meios de leitura por computador.

[0210] As 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 de portas programável em campo (FPGAs), ou outros circuitos lógicos discretos ou integrados equivalentes. Por conseguinte, o termo "processador" tal como aqui usado pode referir-se a qualquer das estruturas anteriores ou qualquer outra estrutura adequada para implementação das técnicas aqui descritas. Além disso, em alguns aspectos, a funcionalidade aqui descrita pode ser fornecida dentro de módulos de hardware e / ou software dedicado configurados para codificação e decodificação, ou incorporada em um codec combinado. Além disso, as técnicas podem ser totalmente implementadas em um ou mais circuitos ou elementos lógicos.

[0211] As técnicas da presente invenção podem ser implementadas em uma vasta variedade de dispositivos ou aparelhos, incluindo um monofone sem fios, um circuito

integrado (IC) ou um conjunto de ICs (por exemplo, um conjunto de chips). Vários componentes, módulos ou unidades são descritos na presente invenção para enfatizar os aspectos funcionais de dispositivos configurados para executar as técnicas divulgadas, mas não precisam necessariamente de realização por diferentes unidades de hardware. Em vez disso, tal como descrito acima, várias unidades podem ser combinadas em uma unidade de hardware codec ou fornecidas por um conjunto de unidades de hardware interoperativas, incluindo um ou mais processadores, conforme descrito acima, em conjunto com software e / ou firmware adequado.

[0212] Vários exemplos da invenção foram descritos. Qualquer combinação dos sistemas, operações ou funções descritas é contemplada. Estes e outros exemplos estão dentro do âmbito das reivindicações a seguir.

REIVINDICAÇÕES

1. Método de decodificação de dados de vídeo, **caracterizado** pelo fato de compreender:

determinar se os dados de vídeo foram codificados usando conversão no espaço de cores;

determinar um primeiro valor para um parâmetro de quantização (QP);

determinar um segundo valor para um segundo QP;

em resposta à determinação de que os dados de vídeo foram codificados usando a conversão no espaço de cores:

quantizar de forma inversa dados de um primeiro componente de cor dos dados de vídeo usando um primeiro desvio, em que o primeiro desvio está associado ao primeiro QP; e

quantizar de forma inversa dados de um segundo componente de cor dos dados de vídeo usando um segundo desvio, em que o segundo desvio está associado ao segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio associado ao segundo QP é diferente do primeiro desvio associado ao primeiro QP, em que o primeiro desvio e o segundo desvio são dependentes de um processo de transformada em espaço de cores para codificar os dados de vídeo.

2. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de compreender adicionalmente:

determinar um primeiro valor associado ao primeiro QP;

determinar um segundo valor associado ao segundo QP;

em resposta à determinação de que conversão no espaço de cores não foi usada na codificação dos dados de vídeo:

quantizar de forma inversa os dados do primeiro componente de cor usando o primeiro valor; e

quantizar de forma inversa os dados do segundo componente de cor usando o segundo valor; e

em resposta à determinação de que os dados de vídeo foram codificados usando conversão no espaço de cores:

quantizar de forma inversa os dados do primeiro componente de cor usando o primeiro desvio adicionado ao primeiro valor; e

quantizar de forma inversa os dados do segundo componente de cor usando o segundo desvio adicionado ao segundo valor, em que o primeiro desvio é diferente do que o segundo desvio.

3. Método, de acordo com a reivindicação 2, **caracterizado** pelo fato de que uma diferença entre o primeiro QP e o segundo QP é dependente de uma norma do processo de transformada no espaço de cores usado para codificar os dados de vídeo.

4. Método, de acordo com a reivindicação 2, **caracterizado** pelo fato de compreender ainda:

quando os dados de vídeo foram codificados usando conversão no espaço de cores:

aumentar a profundidade de bits usada para processos de quantização e transformada para os dados de vídeo em 2, em que o primeiro desvio é igual a 8; ou

aumentar a profundidade de bits usada para processos de quantização e transformada para os dados de vídeo em 1, em que o primeiro desvio é igual a 2;

em que o primeiro componente de cor é preferencialmente um componente luma, e em que o segundo componente de cor é preferencialmente um componente croma.

5. Método de acordo com a reivindicação 1, **caracterizado** pelo fato de que determinar se os dados de vídeo foram codificados usando conversão no espaço de cores compreende:

decodificar um elemento de sintaxe de uma unidade de codificação dos dados de vídeo, em que o elemento de sintaxe indica se a unidade de codificação foi codificada usando conversão no espaço de cores;

determinar se um valor do elemento de sintaxe indica que os dados de vídeo foram codificados usando conversão no espaço de cores; e

em resposta à determinação de que o elemento de sintaxe indica que a unidade de codificação foi codificada usando conversão no espaço de cores, aplicar um processo de transformada inversa no espaço de cores,

em que o elemento de sintaxe compreende uma bandeira de um bit, e em que um valor de 1 para o elemento de sintaxe preferencialmente indica que a unidade de codificação foi codificada usando conversão de espaço de cor.

6. Método de acordo com a reivindicação 5, **caracterizado** pelo fato de que o elemento de sintaxe indica que a unidade de codificação foi codificada usando conversão no espaço de cores quando a unidade de codificação é codificada usando um modo diferente do modo de modulação por código intrapulso (IPCM); e/ou

em que o elemento de sintaxe indica que a unidade de codificação foi codificada usando conversão no espaço de cores quando há coeficientes não nulos em uma unidade de transformada da unidade de codificação; e/ou

em que o elemento de sintaxe indica que a unidade de codificação não foi codificada usando conversão no espaço de cores quando a unidade de codificação é intracodificada e quando um modo de predição de luminância e um modo de predição de croma de uma unidade de predição dentro da unidade de codificação são diferentes, em que o elemento de sintaxe não está presente em um fluxo de bits recebido que compreende os dados de vídeo, e em que a decodificação do elemento de sintaxe compreende inferir o valor do elemento de sintaxe; e/ou

em que o elemento de sintaxe indica que a unidade de codificação não foi codificada usando conversão no espaço de cores quando a unidade de codificação é codificada com um modo de paleta, em que o elemento de sintaxe não está presente em um fluxo de bits recebido que compreende os dados de vídeo, e em que a decodificação do elemento de sintaxe compreende inferir o valor do elemento de sintaxe.

7. Método de acordo com a reivindicação 1, **caracterizado** pelo fato de compreender ainda:

em resposta à determinação de que os dados de vídeo foram codificados usando conversão no espaço de cores, aplicar um processo de transformada inversa no espaço de cores à unidade de codificação, em que o processo de transformada inversa no espaço de cores pode compreender uma matriz de transformada reversa de YCoCg-R original, ou pode compreender uma matriz de transformada reversa de YCoCg modificada;

em que, no caso onde o processo de transformada inversa de espaço de cor compreende uma matriz de transformada reversa de YCoCg modificada, um coeficiente da matriz de transformada reversa de YCoCg modificada é ajustado igual a um coeficiente de uma matriz transformada

inversa de YCoCg original após um deslocamento à direita de 2-bits com um desvio de arredondamento; ou

em que, no caso onde o processo de transformada inversa no espaço de cores compreende uma matriz de transformada de YCoCg original, o método ainda compreende aplicar um processo de normalização aos dados de vídeo.

8. Dispositivo de decodificação de vídeo, **caracterizado** pelo fato de que compreende:

uma memória configurada para armazenar dados de vídeo; e

um ou mais processadores configurados para:

determinar se os dados vídeo foram codificados usando conversão no espaço de cores; e

em resposta à determinação de que os dados de vídeo foram codificados usando conversão no espaço cores:

quantizar de forma inversa dados de um primeiro componente de cor dos dados de vídeo usando um primeiro desvio, em que o primeiro desvio está associado a um primeiro parâmetro de quantização (QP); e

quantizar de forma inversa dados de um segundo componente de cor dos dados de vídeo usando um segundo desvio, em que o segundo desvio está associado a um segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio associado ao segundo QP é diferente do primeiro desvio associado ao primeiro QP, em que o primeiro desvio e o segundo desvio são dependentes de um processo de transformada no espaço de cor usado para codificar os dados de vídeo.

9. Dispositivo, de acordo com a reivindicação 8, **caracterizado** pelo fato de que os um ou mais processadores são adicionalmente configurados para:

determinar um primeiro valor associado ao primeiro QP;

determinar um segundo valor associado ao segundo QP;

em resposta à determinação de que conversão no espaço de cores não foi usada na codificação dos dados de vídeo:

quantizar de forma inversa os dados do primeiro componente de cor usando o primeiro valor; e

quantizar de forma inversa os dados do segundo componente de cor usando o segundo valor; e

em resposta à determinação de que os dados de vídeo foram codificados usando conversão no espaço de cores:

quantizar de forma inversa os dados do primeiro componente de cor usando o primeiro desvio adicionado ao primeiro valor; e

quantizar de forma inversa os dados do segundo componente de cor usando o segundo desvio adicionado ao segundo valor, em que o primeiro desvio é diferente do que o segundo desvio.

10. Dispositivo, de acordo com a reivindicação 9, **caracterizado** pelo fato de que uma diferença entre o primeiro QP e o segundo QP é dependente de uma norma do processo de transformada no espaço de cores usado para codificar os dados de vídeo.

11. Dispositivo, de acordo com a reivindicação 8, **caracterizado** pelo fato de que a unidade de codificação é codificada em um modo diferente do modo de modulação de código intra-pulso (IPCM), em que o elemento de sintaxe é sinalizado somente para unidades de codificação usando um modo diferente do modo IPCM.

12. Memória legível por computador, **caracterizada** pelo fato de compreender instruções que, quando executadas, levam um ou mais processadores de um dispositivo de decodificação de vídeo a realizar o método como definido em qualquer uma das reivindicações 1 a 7 e 13 a 18.

13. Método de codificação de dados de vídeo, **caracterizado** pelo fato de compreender:

determinar quanto ao uso de conversão no espaço de cores para codificar os dados de vídeo; e

em resposta à determinação de usar conversão no espaço de cores para codificar os dados de vídeo:

quantizar os dados de um primeiro componente de cor dos dados de vídeo usando um primeiro desvio, em que o primeiro desvio está associado a um primeiro parâmetro de quantização (QP); e

quantizar os dados de um segundo componente de cor dos dados de vídeo usando um segundo desvio, em que o segundo desvio está associado a um segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio associado ao segundo QP é diferente do primeiro desvio associado ao primeiro QP, em que o primeiro desvio e o segundo desvio são dependentes de um processo de transformada no espaço de cor usado para codificar os dados de vídeo.

14. Método, de acordo com a reivindicação 13, **caracterizado** pelo fato de que uma diferença entre o primeiro QP e o segundo QP é dependente de uma norma do processo de transformada no espaço de cores usado para codificar os dados de vídeo.

15. Método de acordo com a reivindicação 13, **caracterizado** pelo fato de compreender ainda:

em resposta à determinação de usar conversão no espaço de cores para codificar os dados de vídeo,

aumentar a profundidade de bits usada para processos de quantização e transformada para os dados de vídeo em 2, em que o primeiro desvio é igual a 8; ou

aumentar a profundidade de bits usada para processos de quantização e transformada para os dados de vídeo em 1, em que o primeiro desvio é igual a 2;

em que o primeiro componente de cor é preferencialmente um componente luma, e em que o segundo componente de cor é preferencialmente um componente croma.

16. Método de acordo com a reivindicação 13, **caracterizado** pelo fato de compreender ainda, em resposta à determinação de usar conversão no espaço de cores para codificar os dados de vídeo, estabelecer um valor de um elemento de sintaxe que indica que os dados de vídeo foram codificados usando conversão no espaço de cores, em que o elemento de sintaxe compreende uma bandeira de um bit, e em que o valor de 1 para o elemento de sintaxe preferencialmente indica que a unidade de codificação foi codificada usando conversão de espaço de cor.

17. Método de acordo com a reivindicação 16, **caracterizado** pelo fato de que o elemento de sintaxe é sinalizado para indicar que a unidade de codificação foi codificada usando conversão no espaço de cores quando a unidade de codificação é codificada usando um modo diferente do modo de modulação por código intrapulso (IPCM) e/ou

em que o elemento de sintaxe indica que a unidade de codificação foi codificada usando conversão no espaço de cores quando não há coeficientes não nulos em uma unidade de transformada da unidade de codificação; e/ou

em que o elemento de sintaxe não é sinalizado quando a unidade de codificação é codificada com um modo paleta.

18. Método de acordo com a reivindicação 13, **caracterizado** pelo fato de compreender ainda:

em resposta à determinação de usar conversão no espaço de cores para codificar os dados de vídeo, aplicar um processo de transformada no espaço de cores à unidade de codificação, em que o processo de transformada no espaço de cores pode compreender uma matriz de transformada de YCoCg-R original ou pode compreender uma matriz de transformada inversa YCoCg modificada, ou pode compreender uma matriz transformada YCoCg original

em que no caso onde o processo de transformada no espaço de cores compreende uma matriz de transformada inversa de YCoCg modificada, um coeficiente da matriz de transformada de YCoCg modificada é definido como igual a um coeficiente de uma matriz de transformada de YCoCg original após um deslocamento à direita de 2 bits com um desvio de arredondamento, em que a matriz YCoCg modificada preferencialmente tem um deslocamento igual a 2; ou

em que, no caso do processo de transformada de espaço de cor compreender uma matriz transformada de YCoCg original, o método ainda compreende

aplicar um processo de normalização aos dados de vídeo.

19. Dispositivo de codificação de vídeo, **caracterizado** pelo fato de que compreende:

uma memória configurada para armazenar dados de vídeo; e

um ou mais processadores configurados para:

determinar quanto ao uso de conversão no espaço de cores para codificar os dados de vídeo; e

em resposta à determinação de usar conversão no espaço de cores para codificar os dados de vídeo:

quantizar dados de um primeiro componente de cor dos dados de vídeo usando um primeiro desvio, em que o primeiro desvio está associado a um primeiro parâmetro de quantização (QP); e

quantizar dados de um segundo componente de cor dos dados de vídeo usando um segundo desvio, em que o segundo desvio está associado a um segundo QP, em que o segundo componente de cor é diferente do primeiro componente de cor, e em que o segundo desvio associado ao segundo QP é diferente do primeiro desvio associado ao primeiro QP, em que o primeiro desvio e o segundo desvio são dependentes de um processo de transformada no espaço de cor usado para codificar os dados de vídeo.

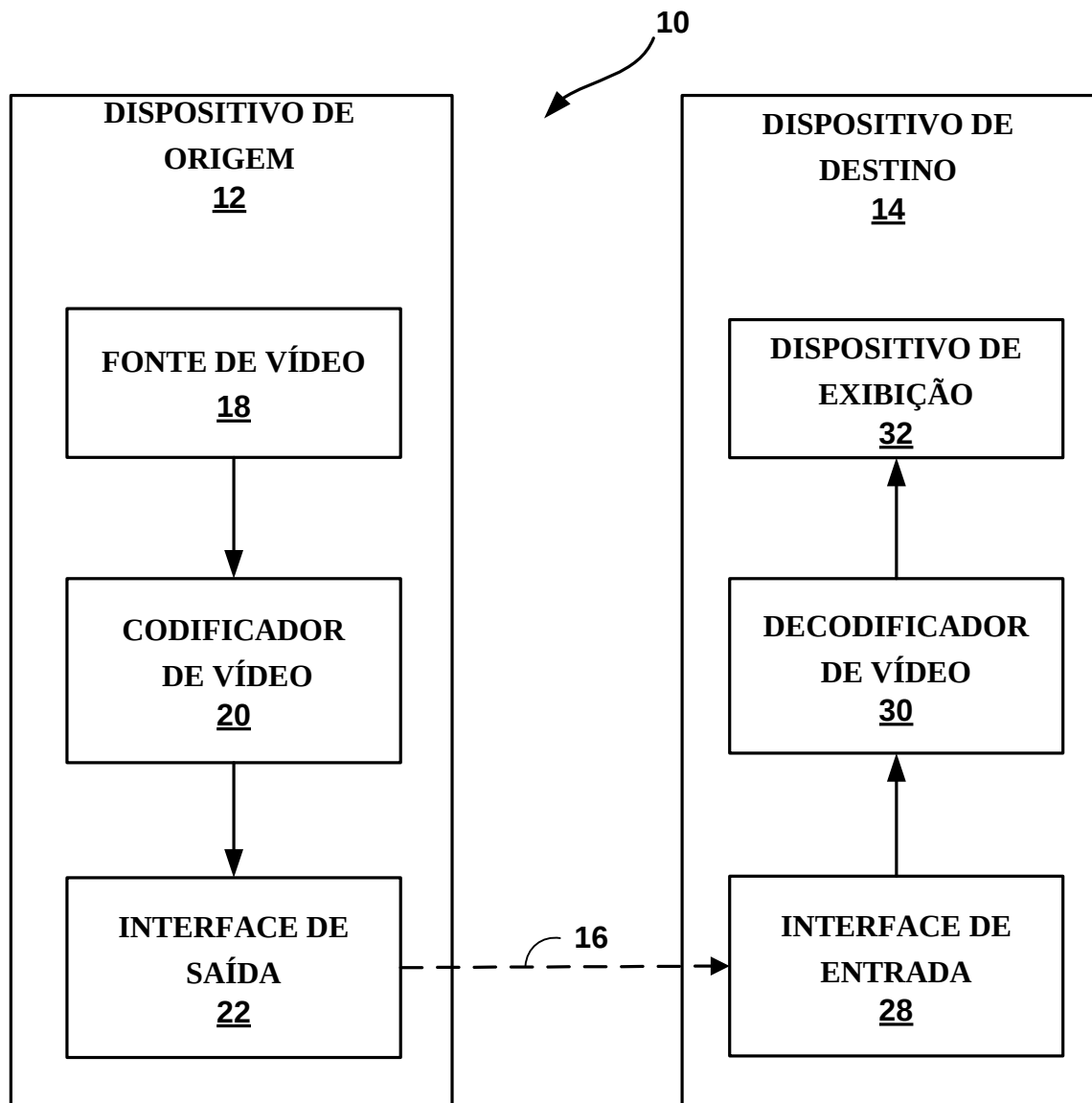


FIG. 1



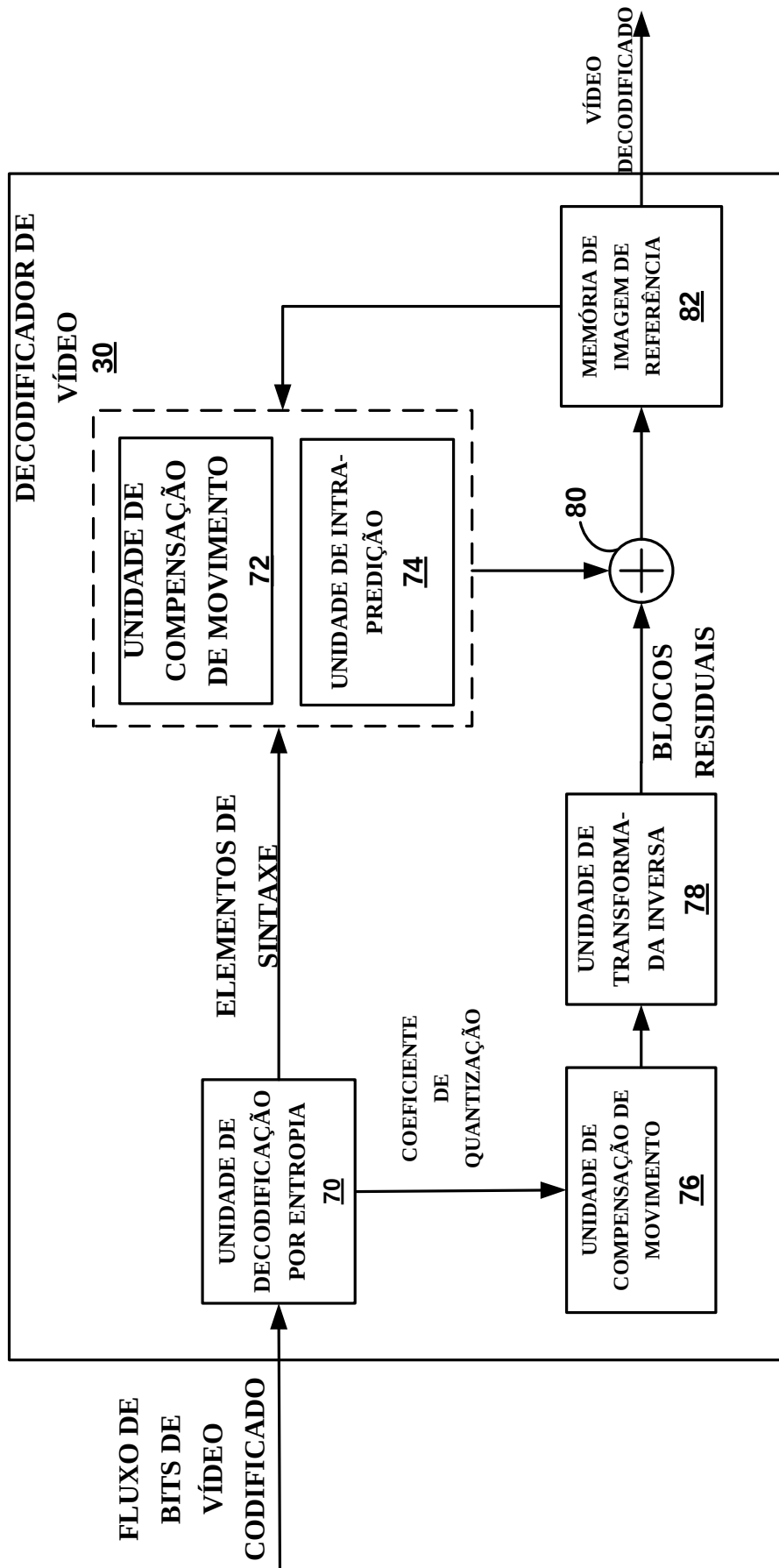
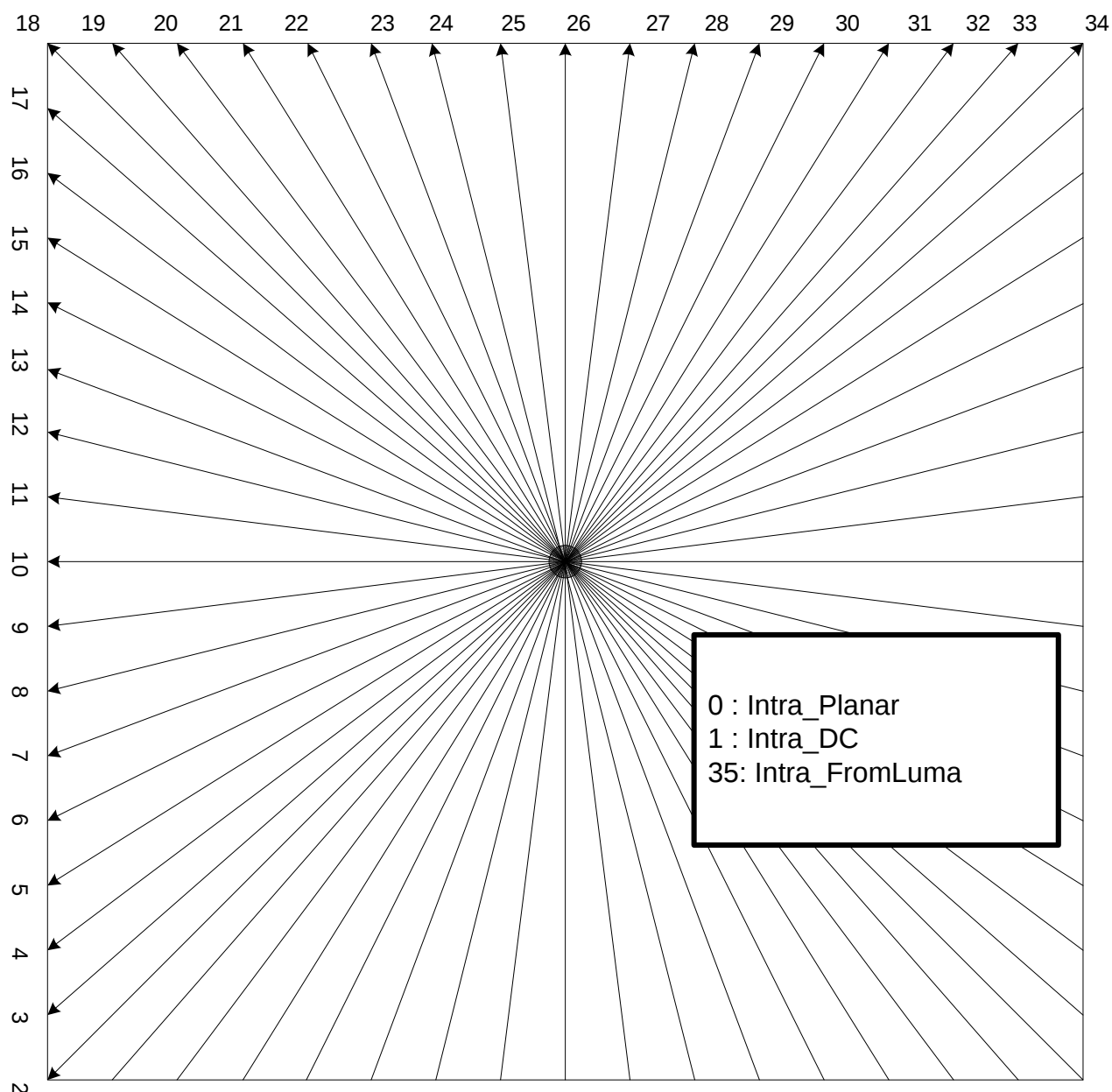


FIG. 3

**FIG. 4**

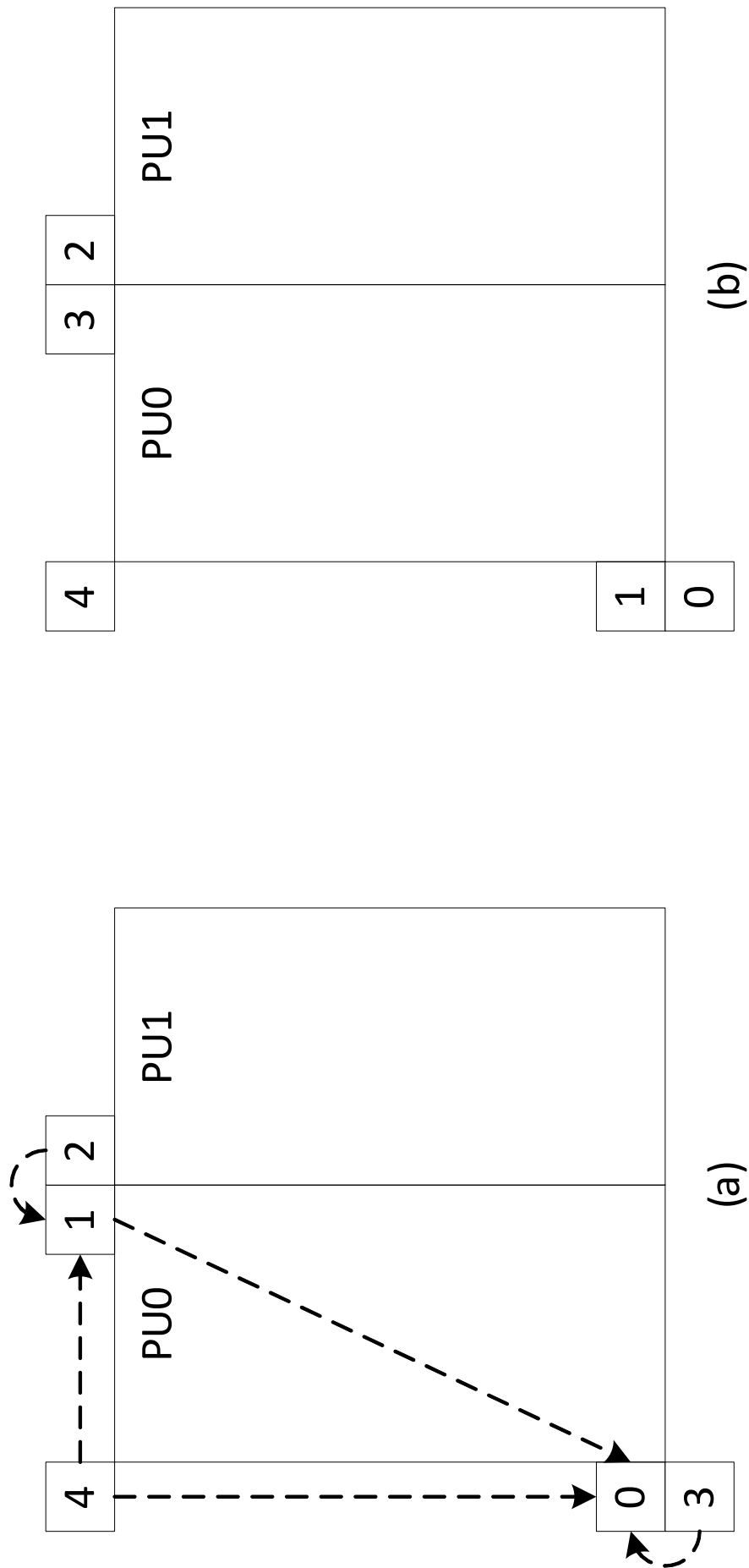


FIG. 5

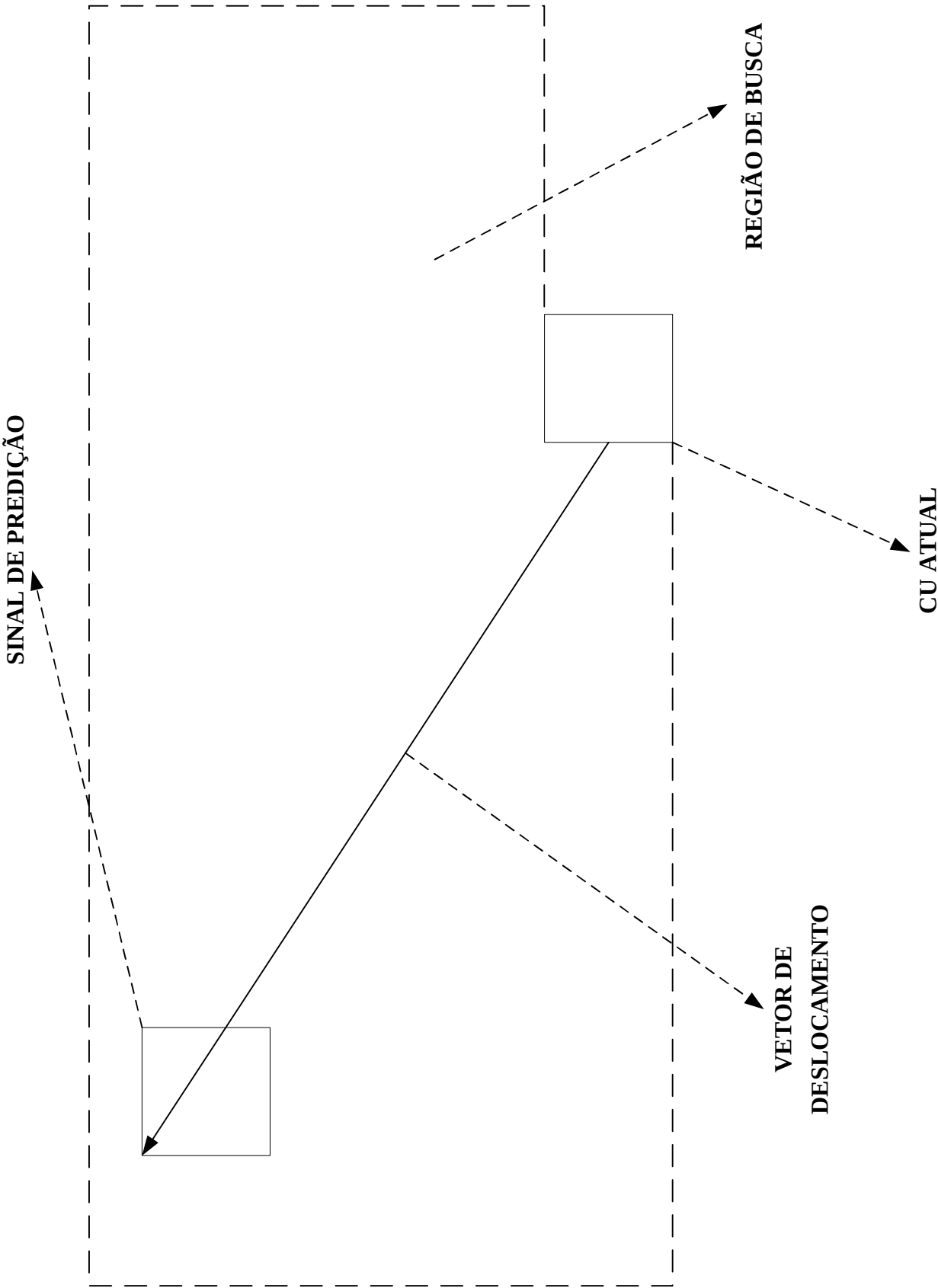
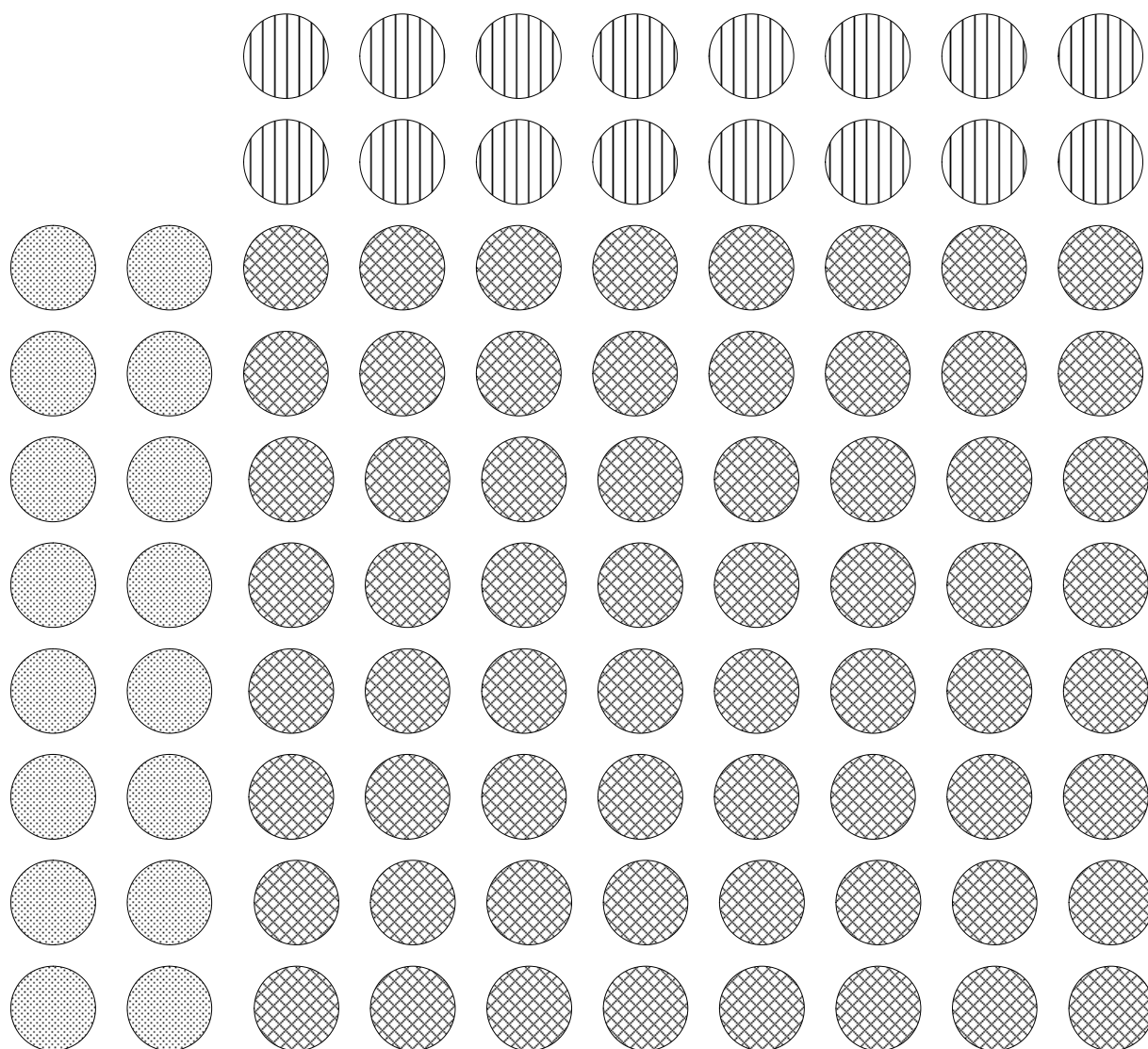
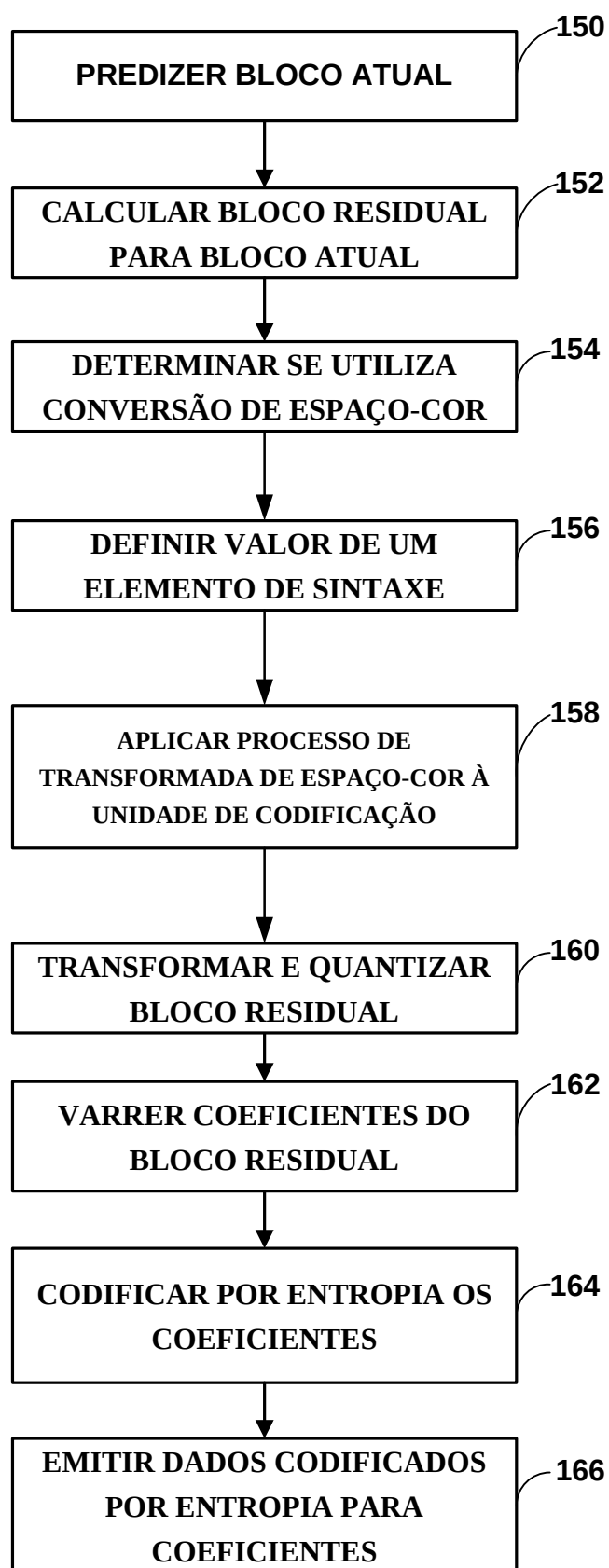
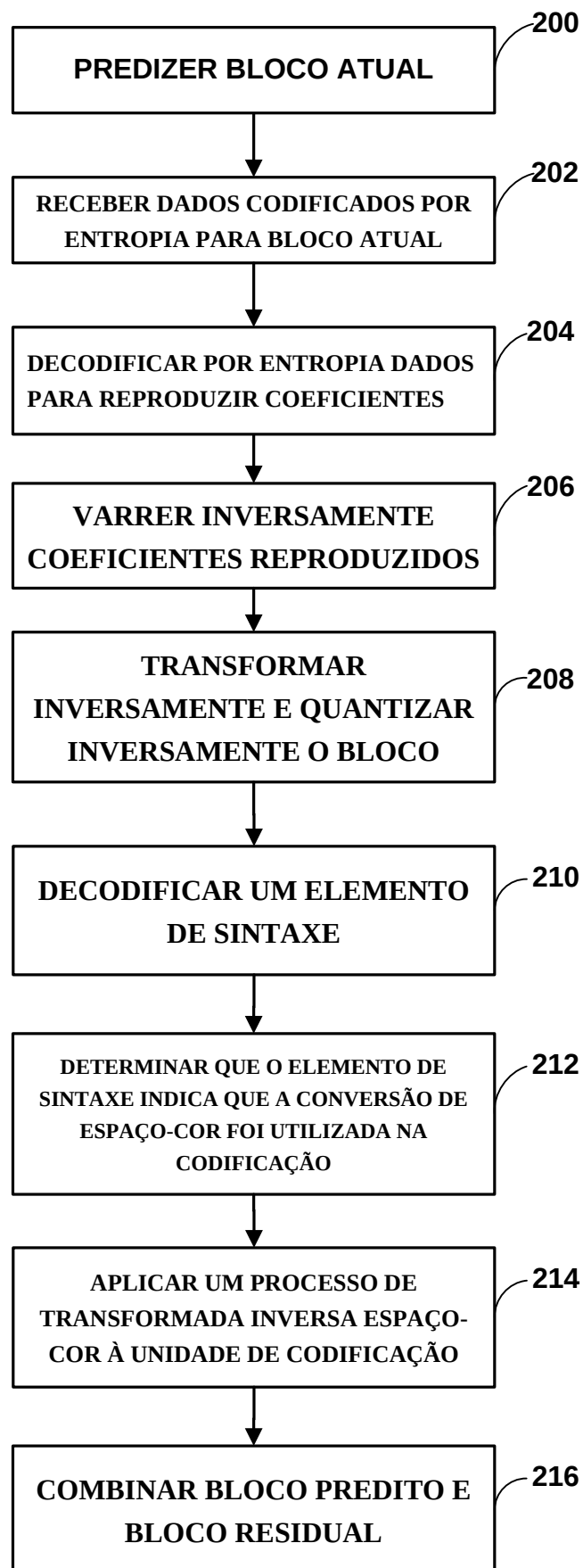


FIG. 6

**FIG. 7**

**FIG. 8**

**FIG. 9**