República Federativa do Brasil

Ministério do Desenvolvimento, Indústria, Comércio e Serviços

Instituto Nacional da Propriedade Industrial

(11) BR 112016021151-0 B1

(22) Data do Depósito: 13/03/2015

(45) Data de Concessão: 27/02/2024

(54) Título: TRANSFORMADA INVERSA DE ESPAÇO-COR TANTO PARA VÍDEO CODIFICADO COM PERDAS COMO SEM PERDAS

(51) Int.Cl.: H04N 19/176; H04N 19/70; H04N 19/126; H04N 19/14; H04N 19/186; (...).

(30) Prioridade Unionista: 10/10/2014 US 62/062,637; 12/03/2015 US 14/656,532; 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.

(86) Pedido PCT: PCT US2015020529 de 13/03/2015

(87) Publicação PCT: WO 2015/138954 de 17/09/2015

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

(57) Resumo: TRANSFORMADA INVERSA DE ESPAÇO-COR TANTO PARA VÍDEO CODIFICADO COM PERDAS COMO SEM PERDAS Em geral, esta descrição descreve técnicas para a codificação de blocos de vídeo, usando um processo de conversão de espaço-cor. Um codificador de vídeo, tal como um codificador de vídeo ou um decodificador de vídeo, pode determinar um modo de codificação utilizado para codificar os dados de vídeo. O modo de codificação pode ser um de um modo de codificação com perdas ou um modo de codificação sem perdas. O codificador de vídeo pode determinar um processo de transformada de espaço-cor dependente do modo de codificação utilizado para codificar os dados de vídeo. O codificador de vídeo pode aplicar o processo de transformada de espaço-cor para codificar os dados de vídeo. Ao decodificar os dados de vídeo, independente do fato de o modo de codificação ser o modo de codificação com perdas ou o modo de codificação sem perdas, o codificador de vídeo pode aplicar o mesmo processo de transformada inversa de espaço-cor em um loop de decodificação do processo de codificação.

"TRANSFORMADA INVERSA DE ESPAÇO-COR TANTO PARA VÍDEO CODIFICADO COM PERDAS COMO SEM PERDAS"

[0001] Este pedido reivindica o benefício do Pedido de Patente Provisório Norte-Americano No. 61/953,573, depositado em 14 de março de 2014, do Pedido de Patente Provisório Norte-Americano No. 61/981,645, depositado em 18 de abril de 2014, e do Pedido de Patente Provisório Norte-Americano No. 62/062,637, depositado em 10 de outubro de 2014, cujos conteúdos de cada um são aqui incorporados em sua totalidade a título de referência.

CAMPO TÉCNICO

[0002] Esta descrição refere-se à codificação de vídeo e, mais especificamente, codificação de vídeo utilizando conversão de espaço-cor.

FUNDAMENTOS

[0003] As funcionalidades de vídeo digital podem ser incorporadas em uma gama variada de dispositivos, compreendendo televisores digitais, sistemas digitais de difusão direta, sistemas de difusão sem fio, assistentes digitais pessoais (PDAs), computadores laptop ou desktop, computadores tablet, leitores de e-book, câmeras digitais, dispositivos de gravação digital, reprodutores de mídia digital, dispositivos de jogos de vídeo, consoles de videogame, telefones de rádio por satélite ou celulares, os chamados "telefones inteligentes", dispositivos de vídeo de teleconferência, dispositivos de fluxo contínuo de vídeo e outros mais. Os dispositivos de vídeo digitais implementam técnicas de codificação de vídeo, como as descritas nos padrões definidos pelo MPEG-2, MPEG-4, ITU-T h.263, ITU-T h.264/MPEG-4, parte 10, Codificação de Vídeo Avançada (AVC), Codificação de Vídeo com Alta Eficiência (HEVC), padrão atualmente em desenvolvimento, e extensões desses padrões. Os dispositivos de vídeo podem transmitir,

receber, codificar, decodificar e/ou armazenar informações de vídeo digitais mais eficientemente através da implementação dessas técnicas de codificação de vídeo.

[0004] As técnicas de codificação de vídeo compreendem a predição espacial (intra-imagens) e/ou a predição temporal (inter-imagens) para reduzir ou remover a redundância inerente em sequências de vídeo. Para a codificação de vídeo baseada em blocos, uma fatia (slice) de vídeo (por exemplo, um quadro de vídeo ou uma parte de um quadro de vídeo) pode ser particionada em blocos de vídeo, que também podem ser referidos como treeblocks, unidades de codificação (CUs) e/ou nós de codificação. Os blocos de vídeo em uma fatia intra-codificada (I) de uma imagem são codificados utilizando-se a predição espacial com relação às amostras de referência em blocos vizinhos na mesma imagem. Os blocos de vídeo em uma fatia inter-codificada (P ou B) de uma imagem podem utilizar a predição espacial com relação às amostras de referência em blocos vizinhos na mesma imagem ou predição temporal com relação às amostras de referência em outras imagens de referência. As imagens podem ser referidas como quadros e as imagens de referência podem ser referidas como quadros de referência.

[0005] A predição espacial ou temporal resulta em um bloco preditivo para um bloco a ser codificado. Os dados residuais representam as diferenças de pixels entre o bloco original a ser codificado e o bloco preditivo. Um bloco inter-codificado é codificado de acordo com um vetor de movimento que aponta para um bloco de amostras de referência para formar o bloco preditivo, e os dados residuais indicam a diferença entre o bloco codificado e o bloco preditivo. Um bloco intra-codificado é codificado de acordo com um modo de intra-codificação e com os dados residuais. Para maior compressão, os dados residuais podem

ser transformados do domínio de pixel para um domínio de transformada, resultando em coeficientes de transformada residuais, que então podem ser quantizados. Os coeficientes de transformada quantizados, inicialmente dispostos em uma matriz bidimensional, podem ser varridos a fim de produzir um vetor unidimensional de coeficientes de transformada e a codificação por entropia pode ser aplicada para obter ainda mais compressão.

SUMÁRIO

[0006] De um modo geral, esta invenção descreve técnicas para codificação de blocos de vídeo utilizando um processo de conversão de espaço-cor. Um codificador de vídeo, tal como um codificador de vídeo ou um decodificador de vídeo, pode determinar um modo de codificação usado para codificar os dados de vídeo. O modo de codificação pode ser um dentre um modo de codificação com perdas ou um modo de codificação sem perdas. O codificador de vídeo pode determinar um processo de transformada de espaço-cor dependente do modo de codificação usado para codificar os dados de vídeo. Ao decodificar os dados de vídeo, independente de se o modo de codificação é o modo de codificação com perdas ou o modo de codificação sem perdas, o codificador de vídeo pode aplicar o mesmo processo de transformada inversa de espaço-cor em um loop (loop) de decodificação do processo de codificação.

[0007] Em um exemplo, um método para decodificar dados de vídeo é descrito. O método compreende receber um primeiro bloco codificado de dados de vídeo, em que o primeiro bloco de dados codificados de vídeo foi codificado usando um modo de codificação com perdas e um primeiro processo de transformada de espaço-cor. O método também inclui receber um segundo bloco codificado de dados de vídeo, em que o segundo bloco de dados codificados de vídeo

foi codificado usando um modo de codificação sem perdas e um segundo processo de transformada de espaço-cor. O método inclui adicionalmente aplicar um processo de transformada inversa de espaço-cor para o primeiro codificado e aplicar o mesmo processo de transformada inversa de espaço-cor para o segundo bloco codificado de dados de vídeo.

[0008] Em outro exemplo, um dispositivo inclui uma memória configurada para armazenar dados de vídeo e um ou mais processadores configurados para receber um primeiro bloco codificado de dados de vídeo, em que o primeiro bloco de dados codificados de vídeo foi codificado usando um modo de codificação com perdas e um primeiro processo de transformada de espaço-cor. Em adição, os um ou mais processadores também estão configurados para receber um segundo bloco codificado de dados de vídeo, em que o segundo bloco de dados codificados de vídeo foi codificado usando um modo de codificação sem perdas e um segundo processo de transformada de espaço-cor. Os um ou mais processadores estão configurados para adicionalmente aplicar um processo de transformada inversa de espaço-cor para o primeiro codificado e aplicar o mesmo processo de transformada inversa de espaço-cor para o segundo bloco codificado de dados de vídeo.

[0009] Em outro exemplo, um dispositivo inclui meios para receber um primeiro bloco codificado de dados de vídeo, em que o primeiro bloco de dados codificados de vídeo foi codificado usando um modo de codificação com perdas e um primeiro processo de transformada de espaço-cor, e meios para receber um segundo bloco codificado de dados de vídeo, em que o segundo bloco de dados codificados de vídeo foi codificado usando um modo de codificação sem perdas e um segundo processo de transformada de espaço-cor. O dispositivo inclui adicionalmente meios para aplicar um

processo de transformada inversa de espaço-cor para o primeiro bloco codificado de dados de vídeo e meios para aplicar o processo de transformada inversa de espaço-cor para o segundo bloco codificado de dados de vídeo.

[0010] Em outro exemplo, um meio de armazenamento legível por computador compreendendo instruções que, quando executadas, fazem um ou mais processadores receber um primeiro bloco codificado de dados de vídeo, em que o primeiro bloco de dados codificados de vídeo foi codificado usando um modo de codificação com perdas e um primeiro processo de transformada de espaço-cor e receber um segundo bloco codificado de dados de vídeo, em que o segundo bloco de dados codificados de vídeo foi codificado usando um modo de codificação sem perdas e um segundo processo de transformada de espaço-cor. As instruções, quando executadas, também fazem o processador aplicar um processo de transformada inversa de espaço-cor para o primeiro bloco codificado de dados de vídeo e aplicar o processo de transformada inversa de espaço-cor para o segundo bloco codificado de dados de vídeo.

[0011] Em outro exemplo, um método para a codificação de dados de vídeo é descrito. O método compreende determinar um modo de codificação utilizado para codificar os dados de vídeo, em que o modo de codificação é um dentre um modo de codificação com perdas ou um modo de codificação sem perdas, e determinar um processo de transformada de espaço-cor dependente do modo de codificação utilizado para codificar os dados de vídeo. O método adicionalmente inclui aplicar o processo de transformada de espaço-cor aos dados de vídeo. O método também inclui aplicar um processo de transformada inversa de espaço-cor em um loop de decodificação do processo de codificação, em que o processo de transformada inversa de

espaço-cor é independente de se o modo de codificação é o modo de codificação com perdas ou o modo de codificação sem perdas.

[0012] Em outro exemplo, um dispositivo inclui uma memória configurada para armazenar dados de vídeo e um ou mais processadores configurados para determinar um modo de codificação utilizado para codificar os dados de vídeo, em que o modo de codificação é um dentre um modo de codificação com perdas ou um modo de codificação sem perdas, e determinar um processo de transformada de espaço-cor dependente do modo de codificação utilizado para codificar os dados de vídeo. Os um ou mais processadores também estão configurados para aplicar o processo de transformada de espaço-cor aos dados de vídeo. Os um ou mais processadores também estão configurados para aplicar um processo de transformada inversa de espaço-cor em um loop de decodificação do processo de codificação, em que o processo de transformada inversa de espaço-cor é independente de se o modo de codificação é o modo de codificação com perdas ou o modo de codificação sem perdas.

[0013] Em outro exemplo, um dispositivo inclui meios para determinar um modo de codificação utilizado para codificar os dados de vídeo, em que o modo de codificação é um dentre um modo de codificação com perdas ou um modo de codificação sem perdas, e meios para determinar um processo de transformada de espaço-cor dependente do modo de codificação utilizado para codificar os dados de vídeo. O dispositivo inclui adicionalmente meios para aplicar o processo de transformada de espaço-cor para os dados de vídeo. O dispositivo também inclui meios para aplicar um processo de transformada inversa de espaço-cor em um loop de decodificação do processo de codificação, em que o processo de transformada inversa de espaço-cor é

independente do fato de o modo de codificação ser o modo de codificação com perdas ou o modo de codificação sem perdas.

[0014] Em outro exemplo, um meio de armazenamento legível por computador compreende instruções que, quando executadas, fazem um ou mais processadores determinar um modo de codificação utilizado para codificar os dados de vídeo, em que o modo de codificação é um dentre um modo de codificação com perdas ou um modo de codificação sem perdas, e determinar um processo de transformada de espaço-cor dependente do modo de codificação utilizado para codificar os dados de vídeo. As instruções fazem adicionalmente os um ou mais processadores aplicar o processo de transformada de espaço-cor para os dados de vídeo. As instruções também fazem os um ou mais processadores aplicar um processo de transformada inversa de espaço-cor em um loop de decodificação do processo de codificação, em que o processo de transformada inversa de espaço-cor é independente do fato de o modo de codificação ser o modo de codificação com perdas ou o modo de codificação sem perdas.

[0015] Os detalhes de um ou mais exemplos da invenção são estabelecidos nos desenhos acompanhantes e na descrição abaixo. Outras características, objetos e vantagens da presente invenção serão evidentes a partir da descrição e figuras, e a partir das reivindicações.

BREVE DESCRIÇÃO DOS DESENHOS

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

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

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

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

[0020] A FIG. 5 é um diagrama conceitual ilustrando candidatos espaciais de vectores de movimento vizinhos para os modos de fusão e de predição de vector de movimento avançado (AMVP) de acordo com uma ou mais das técnicas da presente descrição atual.

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

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

[0023] A FIG. 8 é um diagrama de fluxo ilustrando uma técnica de codificação de acordo com uma ou mais técnicas da divulgação atual.

[0024] A FIG. 9 é um diagrama de fluxo ilustrando uma técnica de decodificação de acordo com uma ou mais técnicas da divulgação atual.

DESCRIÇÃO DETALHADA

[0026] Em alguns exemplos, esta divulgação está relacionada com a codificação de conteúdo de tela, em que o formato de amostragem de alta crominância 4:4:4 é usada. Em alguns exemplos, esta divulgação é também aplicável para extensões de faixa (RCEx), incluindo o suporte de profundidade elevada de bits, possivelmente, (mais de 8 bits), formato de amostragem de alta crominância 4:4:4. Em alguns exemplos, esta divulgação é também aplicável a

outros formatos de cor, tais como formato de amostragem de crominância é 4:2:2. Mais especificamente, na presente divulgação, muitas técnicas diferentes relacionadas com a conversão de espaço-cor 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 de Vídeo Escalonável (SVC) e Codificação de Vídeo Multivista (MVC).

[0027] O projeto de um novo padrão de codificação de vídeo, ou seja, Codificação de Vídeo de Alta Eficiência (HEVC), foi finalizado pela Equipe de Colaboração Conjunta em Codificação de Vídeo (JCT-VC) do Grupo de Experts em Codificação de Vídeo do ITU-T (VCEG) e Grupo de Experts em Imagem em Movimento (MPEG) ISO/IEC.

[0028] No HEVC, a maior unidade de codificação em uma fatia é denominada bloco de árvore de codificação (CTB). O CTB contém uma quadtree, cujos os nós são unidades de codificação. O tamanho de um CTB pode variar de 16x16 a 64x64 no perfil principal do HEVC, embora os tamanhos de CTB 8x8 possam ser suportados. Uma unidade de codificação (CU) pode ser do mesmo tamanho que um CTB e tão pequeno como o tamanho 8x8. Cada CU é codificada com um modo. Quando uma CU é inter-codificada, a CU pode ser ainda dividida em duas unidades de predição (PUs) ou se tornar uma única PU quando não se aplicam mais particionamentos. Quando duas PUs estão presentes em uma CU, cada PU pode ser retângulos com metade do tamanho ou dois retângulos com um tamanho igual a 1/4 ou a 3/4 do tamanho da CU.

[0029] Quando a CU é inter-codificada, um conjunto de informações de movimento está presente para cada PU. Além disso, cada PU é codificada com um modo de

inter-predição único para derivar o conjunto de informações de movimento. No HEVC, os menores tamanhos de PU são 8x4 e 4x8.

[0030] O HEVC especifica quatro unidades de transformada (TUs) com tamanhos de 4x4, 8x8, 16x16 e 32x32 para codificar a predição residual. Um CTB pode ser recursivamente particionado em 4 ou mais TUs. As TUs utilizam as funções baseadas em números inteiros que são semelhantes às funções de transformada discreta de cosseno (DCT). Adicionalmente, os blocos de transformada de luminância 4x4 que pertencem a uma região intra codificada são transformados utilizando-se uma transformada de um número inteiro que é derivada a partir de uma função de transformada discreta de seno (DST). Crominância usa os mesmos tamanhos de TU que de luminância.

[0031] No atual padrão HEVC, para a componente de luminância de cada Unidade de Predição (PU), um método de intra-predição é utilizado com 33 modos de predição angular (indexados a partir de 2 até 34), um modo DC (indexado com 1) e um modo Planar (indexado com 0), conforme descrito abaixo com relação à Fig. 4.

[0032] Além dos 35 intra modos acima, mais um modo, denominado 'I-PCM', também é empregado pelo HEVC. No modo I-PCM a predição, transformada, quantização e codificação por 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 de quando o sinal não pode ser eficientemente codificado por outros modos.

[0033] No atual padrão HEVC, existem dois modos de inter predição disponíveis. Um é o modo de fusão (o salto (skip) é considerado como um caso especial de fusão),

e o segundo modo é o modo avançado de predição de vetor de movimento (AMVP) para uma unidade de predição (PU).

[0034] Tanto no modo AMVP quanto no modo de fusão, é mantida uma lista de candidatos de vetor de movimento (MV) para múltiplos preditores de vetor de movimento. O(s) vetor(es) de movimento, bem como os índices de referência no modo de fusão da PU atual, são gerados tomando-se um candidato da lista de candidatos MV.

[0035] A lista de candidatos MV pode conter até 5 candidatos para o modo de fusão e apenas dois candidatos para o modo AMVP. Um candidato de fusão pode conter um conjunto de informações de movimento, por exemplo, vetores de movimento correspondentes tanto às listas de imagem de referência (lista 0 e lista 1) quanto os índices de referência. Se um candidato de fusão é identificado por um índice de fusão, as imagens de referência são utilizadas para a predição dos blocos atuais, e também são determinados os vetores de movimento associados. No entanto, no modo AMVP para cada direção de predição potencial, a partir ou da lista 1 ou da lista 0, um índice de referência precisa ser explicitamente sinalizado, juntamente com um índice MVP para a lista de candidatos MV, uma vez que o candidato AMVP contém apenas um vetor de movimento. No modo AMVP, os vetores de movimento preditos podem ser ainda mais refinados.

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

[0037] Conforme descrito abaixo, com relação à Fig. 5, os candidatos espaciais MV são derivados dos blocos vizinhos mostrados na Fig. 5, para uma PU específica (PU0). No entanto, os métodos geram os candidatos a partir de blocos diferentes para modos de fusão e AMVP.

[0038] No modo de fusão, até quatro candidatos espaciais MV podem ser derivados nas ordens mostradas na Fig. 5(a), com números. A ordem é a seguinte: esquerda (0), acima (1), acima à direita (2), abaixo à esquerda (3) e acima à esquerda (4), como mostrado na Fig. 5(a).

[0039] No modo AMVP, os blocos vizinhos são divididos em dois grupos: um grupo à esquerda compreendendo os blocos 0 e 1 e um grupo acima compreendendo os blocos 2, 3 e 4, conforme mostrado na Fig. 5(b). Para cada grupo, o candidato em potencial em um bloco vizinho referindo-se à mesma imagem de referência que a indicada pelo índice de referência sinalizado, tem a prioridade mais alta para ser escolhido para formar um candidato final do grupo. É possível que todos os blocos vizinhos não contenham um vetor de movimento apontando para a mesma imagem de referência. Portanto, se tal candidato não pode ser encontrado, o primeiro candidato disponível será escalado para formar o candidato final, assim, as diferenças de distância temporal podem ser compensadas.

[0040] Como descrito abaixo, com relação a Fig. 6, a Cópia Intra Bloco (BC) foi incluída no SCC. Um exemplo de Intra BC é mostrado na Fig. 6, onde a CU atual é predita a partir de um bloco já 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, outras restrições possam ser aplicadas.

[0041] Em alguns exemplos, um processo de transformada de espaço-cor em loop (in-loop) para sinais residuais pode ser utilizado para sequências em formato de crominância 4:4:4. Este método transforma sinais de erro de predição em formato de crominância RGB/YUV naqueles em um espaço-cor sub-ótimo. Com esta etapa adicional, a correlação entre as componentes de cor poderia ser reduzida ainda mais. A matriz de transformada é derivada a partir de valores de amostras de pixel para cada unidade de codificação por uma decomposição de valor singular (SVD). A transformada de espaço-cor é aplicada a um erro de predição tanto do modo intra quanto do inter.

[0042] Em alguns exemplos, quando o processo de transformada de espaço-cor é aplicado ao inter modo, o resíduo é primeiramente convertido em um domínio diferente com a matriz de transformada derivada. Após a conversão de espaço-cor, as etapas de codificação convencionais, como DCT/DST, quantização e codificação por entropia são realizadas em ordem.

[0043] Em alguns exemplos, quando o processo de transformada de espaço-cor é aplicado ao intra modo, a predição e o bloco atual são primeiramente convertidos em um domínio diferente com a matriz de transformada derivada, respectivamente. Após a conversão de espaço-cor, o resíduo entre o bloco atual e seu preditor é adicionalmente transformado com DCT/DST, quantizado e codificado por entropia.

[0044] Na operação a seguir, uma matriz de transformada de espaço-cor é aplicada aos três planos G, B e R, da seguinte forma:

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

[0045] Os valores resultantes são ajustados para baixo (clipped) dentro da faixa de especificação do HEVC, porque, em alguns exemplos, os valores são aumentados até $\sqrt{3}$ vezes. Na operação inversa, uma matriz de transformada de espaço-cor é aplicada para os três componentes P', Q' e R', da seguinte forma:

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

[0046] Como descrito abaixo com relação a Fig. 7, uma matriz de transformada pode ser derivada de valores da amostra de referência. Amostras de referência diferentes podem ser utilizadas para o caso intra e para o caso inter. Para o caso de um bloco intra codificado, um bloco alvo e amostras de referência são mostrados na Fig. 7. Neste exemplo, o bloco alvo consiste em amostras 8x8 hachuradas com linhas cruzadas e as amostras de referência são listradas e pontilhadas.

[0047] Para o caso de um bloco inter-codificado, as amostras de referência para a derivação da matriz são as mesmas que as amostras de referência para compensação de movimento. Para poder realizar a operação de desvio, as amostras de referência no bloco AMP são sub-amostradas de forma que o número de amostras torna-se potência de dois. Por exemplo, o número de amostras de referência em um bloco de 12 x 16 é reduzido por 2/3.

[0048] De acordo com as técnicas da presente descrição, o processo de transformada de espaço-cor pode ser aplicado para cada CU. Portanto, não há necessidade de sinalizar se o processo de transformada é invocado ou não. Além disso, ambos os lados do codificador e do decodificador derivam a matriz de transformada com o mesmo

método, para evitar o overhead para a sinalização da matriz de transformada.

[0049] De acordo com as técnicas da presente descrição, processos de transformada de espaço-cor, tais como matrizes de transformada de espaço-cor, são usados. Uma dessas matrizes é a matriz de transformada YCbCr, que é:

$$\text{Direta: } \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1172 & -0.3942 & 0.5114 \\ 0.5114 & -0.4645 & -0.0469 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\text{Inversa: } \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.5397 \\ 1 & -0.1831 & -0.4577 \\ 1 & 1.8142 & 0 \end{bmatrix} \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix}$$

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

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

$$\text{Inversa: } \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix}$$

[0051] Outra dessas matrizes é a matriz YCoCg-R, que é a versão revisada da matriz YCoCg, que escalona as componentes Co e Cg por um fator de dois. Utilizando uma técnica de levantamento, as transformadas direta e inversa poderiam ser obtidas através das seguintes equações:

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

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

[0052] Nas equações e matrizes acima, as transformadas diretas podem ser realizadas durante o

processo de codificação (por exemplo, por um codificador de vídeo), e as transformadas inversas podem ser realizadas no processo de decodificação (por exemplo, por um decodificador de vídeo).

[0053] Na codificação de vídeo tradicional, considera-se que as imagens têm tonalidade contínua e são espacialmente suavizadas. Com base nessas considerações, foram desenvolvidas várias ferramentas (ex. transformada baseada em bloco, filtragem, etc.) que têm mostrado bom desempenho para vídeos com conteúdo natural. Entretanto, em certas aplicações (ex., um desktop remoto, visores de trabalho colaborativo, e visores sem fio), o conteúdo gerado pela tela do computador pode ser o conteúdo dominante a ser compactado. Este tipo de conteúdo tende a ter uma tonalidade discreta e linhas nítidas características e os limites do objeto são de alto contraste. A suposição de tonalidade contínua e de suavidade pode não mais ser aplicável e, assim, as técnicas de codificação de vídeo tradicionais podem não funcionar eficientemente.

[0054] A codificação no modo de paleta pode ser utilizada para superar as deficiências acima. Exemplos de técnicas de codificação no modo de paleta são descritos no Pedido de Patente Provisório Norte-Americano No. de série 61/810,649, depositado em 10 de abril de 2013. Para cada CU, uma paleta pode ser derivada, a qual inclui os valores de pixels mais dominantes na CU atual. O tamanho e os elementos da paleta são transmitidos primeiro. Após a transmissão, os pixels na CU são codificados de acordo com uma determinada ordem de varredura. Para cada localização, um elemento de sintaxe, tal como um indicador, palette_flag, é transmitido primeiro para indicar se o

valor de pixel está na paleta (“modo de execução”) ou não ("modo de pixel").

[0055] No “modo de execução”, o índice de paleta pode ser sinalizado, seguido pelo “run”. O “run” é um elemento de sintaxe que indica o número de pixels consecutivos em uma ordem de varredura que tem o mesmo valor do índice de paleta do pixel que está sendo codificado. Se múltiplos pixels, em sucessão imediata na ordem de varredura, têm o mesmo valor de índice de paleta, então o “modo de execução” pode ser indicado pelo elemento de sintaxe, como um palette_flag. Um valor de contador pode ser determinado, o qual iguala o número de pixels que sucedem o pixel atual que têm o mesmo valor do índice de paleta do pixel atual, e o “run” é definido como igual ao valor do contador. Nem o palette_flag nem o índice de paleta precisam ser transmitidos para as posições seguintes que são cobertas pelo “run” uma vez que todos têm o mesmo valor de pixel. Do lado do decodificador, somente o primeiro valor do índice de paleta do pixel atual pode ser decodificado, e o resultado deve ser duplicado para cada pixel no “run” dos pixels indicados no elemento sintaxe “run”.

[0056] No “modo de pixel”, o valor da amostra de pixel é transmitido para esta posição. Se o elemento de sintaxe, como o palette_flag, indica “modo de pixel”, então o valor do índice de paleta só é determinado para um pixel sendo decodificado.

[0057] Técnicas convencionais podem sofrer vários problemas. Por exemplo, pode ser invocada uma transformada de espaço-cor que não leva em consideração as características de sequência e a diversidade local. Por conseguinte, o desempenho de codificação pode ser sub-ótimo. Em outro exemplo, pode ser necessária a derivação de

uma matriz de transformada no decodificador, o que aumenta significativamente a complexidade do decodificador. Além disso, a matriz de transformada pode ser derivada utilizando os pixels espaciais reconstruídos ou o preditor de PUs inter-codificadas. No entanto, a eficiência da matriz de transformada pode ser reduzida quando o tamanho da PU é relativamente pequeno, a predição não é muito precisa, ou os pixels vizinhos não estão disponíveis. Técnicas desta descrição podem superar um ou mais destes problemas.

[0058] Um codificador de vídeo, tal como codificador de vídeo de 20 ou decodificador de vídeo 30, pode realizar as técnicas descritas nesta descrição. Em geral, esta descrição descreve técnicas para a codificação de blocos de vídeo usando um processo de conversão de espaço-cor. Um codificador de vídeo, tal como um codificador de vídeo 20 ou decodificador de vídeo 30, pode determinar um modo de codificação utilizado para codificar os dados de vídeo. O modo de codificação pode ser um dentre um modo de codificação com perdas ou um modo de codificação sem perdas. O codificador de vídeo pode determinar um processo de transformada de espaço-cor dependente do modo de codificação utilizado para codificar os dados de vídeo. O codificador de vídeo pode aplicar o processo de transformada de espaço-cor em codificar os dados de vídeo. Na decodificação dos dados de vídeo, independente do fato de o modo de codificação ser o modo de codificação com perdas ou o modo de codificação sem perdas, o codificador de vídeo pode aplicar o mesmo processo de transformada inversa de espaço-cor em um loop de decodificação do processo de codificação.

[0059] Esta descrição descreve técnicas que podem melhorar o desempenho de codificação de transformada de

espaço-cor em loop e podem reduzir a complexidade do decodificador. A Fig. 1 é um diagrama de blocos ilustrando um exemplo do sistema de codificação e decodificação de vídeo 10 que pode utilizar técnicas para codificação de conteúdo de tela, em que é utilizado um formato de amostragem de alta crominância. Como mostrado na FIG. 1, o sistema 10 compreende um dispositivo de origem 12 que proporciona dados de vídeo codificados para serem decodificados em um momento posterior por um dispositivo de destino 14. Em particular, o dispositivo de origem 12 fornece os dados de vídeo para o dispositivo de destino 14 através de um meio legível por computador 16. O dispositivo de origem 12 e o dispositivo de destino 14 podem compreender qualquer um dentre uma ampla gama de dispositivos, incluindo computadores desktop, computadores portáteis (i.e., laptop), computadores tablet, set-top box, aparelhos de telefone tais como os chamados "Telefones inteligentes", os assim chamados "pads" inteligentes, televisores, câmeras, dispositivos de exibição, reprodutores de mídia digital, consoles de videogame, dispositivo de fluxo contínuo de vídeo, ou outros dispositivos similares. Em alguns casos, o dispositivo de origem 12 e o dispositivo de destino 14 podem ser equipados para comunicação sem fio.

[0060] O dispositivo de destino 14 pode receber os dados de vídeo codificados para serem decodificados através de um meio legível por computador 16. O meio legível por computador 16 pode compreender qualquer tipo de meio ou dispositivo capaz de mover os dados de vídeo codificados do dispositivo de origem 12 para o dispositivo de destino 14. Em um exemplo, o meio legível por computador 16 pode compreender um meio de comunicação para habilitar o dispositivo de origem 12 a transmitir dados de vídeo

codificados diretamente para o 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 transmitidos para o dispositivo de destino 14. O meio de comunicação pode compreender qualquer meio de comunicação sem fio ou com fio, tal como um espectro de rádio frequência (RF) ou uma ou mais linhas de transmissão físicas. O meio de comunicação pode formar parte de uma rede baseada em pacotes, como uma rede de área local, uma rede de área estendida, ou uma rede de área global como a Internet. O meio de comunicação pode compreender roteadores, comutadores, estações base, ou qualquer outro equipamento que possa ser útil para facilitar a comunicação do dispositivo de origem 12 para o dispositivo de destino 14.

[0061] Em alguns exemplos, dados codificados podem ser emitidos da interface de saída 22 para um dispositivo de armazenamento. De modo similar, os dados codificados podem ser acessados a partir do dispositivo de armazenamento por uma interface de entrada. O dispositivo de armazenamento pode compreender qualquer um dentre uma variedade de meios de armazenamento de dados distribuídos ou acessados localmente 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 dados de vídeo codificados. Em outro exemplo, o dispositivo de armazenamento pode corresponder a um servidor de arquivos ou outro dispositivo de armazenamento intermediário que pode armazenar o vídeo codificado gerado pelo dispositivo de 12. O dispositivo de destino 14 pode acessar dados de vídeo armazenados no dispositivo de armazenamento via fluxo contínuo ou download. O servidor de arquivos pode ser qualquer tipo de

servidor capaz de armazenar dados de vídeo codificados e transmitir esses dados de vídeo codificados para o dispositivo de destino 14. Exemplos de servidores de arquivos incluem um servidor de web (por exemplo, para um site), um servidor FTP, dispositivos de armazenamento ligados em rede (NAS) ou uma unidade 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 compreender um canal sem fio (por exemplo, uma conexão Wi-Fi), uma conexão com fio (por exemplo, DSL, modem a cabo, etc.) ou uma combinação de ambos que seja adequada para acessar 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 por fluxo contínuo, uma transmissão por download, ou uma combinação das mesmas.

[0062] As técnicas desta descrição não estão necessariamente limitadas às aplicações ou configurações sem fio. As técnicas podem ser aplicadas para codificar vídeo em apoio a qualquer uma dentre uma variedade de aplicações multimídia, tais como difusões de televisão através do ar, transmissões de televisão por cabo, transmissões de televisão por satélite, transmissões de vídeo por fluxo contínuo de Internet, tal como fluxo contínuo adaptativo dinâmico 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 unidirecional ou bidirecional para suportar aplicativos tais como fluxo contínuo de vídeo, reprodução de vídeo, difusão de vídeo e/ou vídeo telefonia.

[0063] No exemplo da Fig. 1, o dispositivo de origem 12 compreende a fonte de vídeo 18, o codificador de vídeo 20 e a interface de saída 22. O dispositivo de destino 14 compreende a interface de entrada 28, o decodificador de vídeo 30 e o dispositivo de exibição 32. De acordo com esta descrição, o codificador de vídeo 20 do dispositivo de origem 12 pode ser configurado para aplicar as técnicas para codificar blocos de vídeo utilizando um processo de conversão de espaço-cor. Em outros exemplos, um dispositivo de origem e um dispositivo de destino podem compreender outros componentes ou arranjos. Por exemplo, o dispositivo de origem 12 pode receber dados de vídeo de uma fonte de vídeo 18 externa, tal como uma câmera externa. Da mesma forma, o dispositivo destino 14 pode fazer a interface com um dispositivo de exibição externo, ao invés de incluir um dispositivo de exibição integrado.

[0064] O sistema 10 ilustrado na Fig. 1 é apenas um exemplo. As técnicas para codificar blocos de vídeo utilizando um processo de conversão de espaço-cor podem ser realizadas por qualquer dispositivo de codificação e/ou decodificação de vídeo digital. Embora geralmente as técnicas desta descrição sejam realizadas por um dispositivo de codificação de vídeo, as técnicas podem também ser realizadas por um codificador/decodificador de vídeo, normalmente referido como um "CODEC". Além disso, as técnicas desta descrição também podem ser realizadas por um pré-processador de vídeo. O dispositivo de origem 12 e o dispositivo de destino 14 são apenas exemplos desses dispositivos de codificação em que o dispositivo de origem 12 gera dados de vídeo codificados para transmissão para o dispositivo de destino 14. Em alguns exemplos, os dispositivos 12, 14 podem operar de uma forma substancialmente simétrica de forma que cada um dos

dispositivos 12, 14 compreende componentes de codificação e decodificação de vídeo. Assim, o sistema 10 pode suportar a transmissão de vídeo unidirecional ou bidirecional entre dispositivos de vídeo 12, 14, por exemplo, para fluxo contínuo de vídeo, reprodução de vídeo, difusão de vídeo ou vídeo telefonia.

[0065] A fonte de vídeo 18 do dispositivo de origem 12 pode compreender um dispositivo de captura de vídeo, como uma câmera de vídeo, um arquivo de vídeo contendo um vídeo anteriormente capturado e/ou uma interface de alimentação de vídeo para receber o vídeo de um provedor de conteúdos de vídeo. Como uma alternativa adicional, a fonte de vídeo 18 pode gerar dados baseados em gráficos de computador conforme o vídeo de origem, ou uma combinação de vídeo ao vivo, vídeo arquivado e vídeo gerado por computador. Em alguns casos, se a fonte de vídeo 18 é uma câmera de vídeo, o dispositivo de origem 12 e o dispositivo de destino 14 podem formar os assim chamados telefones com câmera ou vídeo telefones. Como mencionado acima, no entanto, as técnicas descritas nesta descrição podem ser aplicáveis à codificação de vídeo em geral e podem ser aplicadas em aplicações sem fio e/ou com fio. Em cada caso, o vídeo capturado, pré-capturado ou gerado por computador pode ser codificado pelo codificador de vídeo 20. A informação de vídeo codificado pode então ser emitida pela interface de saída 22 para um meio legível por computador 16.

[0066] O meio legível por computador 16 pode compreender um meio transitório, como uma transmissão por difusão sem fio ou por uma rede com fio, ou um meio de armazenamento (isto é, meio de armazenamento não transitório), tais como um disco rígido, flash drive, disco compacto (CD), disco de vídeo digital (DVD), disco Blu-ray

ou outra mídia legível por computador. Em alguns exemplos, um servidor de rede (não mostrado) pode receber dados de vídeo codificados do dispositivo de origem 12 e fornecer os dados de vídeo codificados para o dispositivo de destino 14, por exemplo, através de transmissão de rede. De forma similar, um dispositivo de computação de uma instalação produção de mídia, tal como uma instalação de marcação em disco, pode receber dados de vídeo codificados do dispositivo de origem 12 e produzir um disco (disc) que contém os dados de vídeo codificados. Portanto, o meio legível por computador 16 pode ser entendido como compreendendo um ou mais meios legíveis por computador de várias formas, em vários exemplos.

[0067] A interface de entrada 28 do dispositivo de destino 14 recebe informações do meio legível por computador 16. A informação de meio legível por computador 16 pode compreender informação de sintaxe definida pelo codificador de vídeo 20, que também é utilizada pelo decodificador de vídeo 30, que inclui elementos de sintaxe que descrevem as características e/ou processamentos de blocos e outras unidades codificadas, por exemplo, GOPs. O dispositivo de exibição 32 exibe os dados de vídeo decodificados para um usuário, e pode compreender qualquer um dentre uma variedade de dispositivos de exibição, tal como um tubo de raios catódicos (CRT), um monitor de cristal líquido (LCD), um monitor de plasma, um monitor de diodo emissor de luz orgânico (OLED) ou outro tipo de dispositivo de exibição.

[0068] O codificador de vídeo 20 e decodificador de vídeo 30 podem operar de acordo com um padrão de codificação de vídeo, como o padrão Codificação de Vídeo com Alta Eficiência (HEVC), atualmente em desenvolvimento, e podem se conformar com o Modelo de Teste HEVC (HM).

Alternativamente, o codificador de vídeo 20 e o decodificador de vídeo 30 podem operar de acordo com outros padrões proprietários ou industriais, como o padrão ITU-T H.264, alternativamente conhecido como MPEG-4, parte 10, Codificação de Vídeo Avançada (AVC), ou extensões de tais padrões. As técnicas desta descrição, no entanto, não estão limitadas a qualquer padrão de codificação particular. Outros exemplos de padrões de codificação de vídeo incluem o MPEG-2 e o ITU-T H. 263. Embora não mostrado na Fig. 1, em alguns aspectos, cada um entre o codificador de vídeo de 20 e o decodificador de vídeo 30 pode ser integrado com um codificador e um decodificador de áudio e podem compreender unidades MUX-DEMUX adequadas, ou outro hardware e software, para lidar com a codificação tanto de áudio quanto de vídeo em um fluxo de dados em comum ou fluxos de dados separados. Se aplicável, as unidades MUX-DEMUX podem se conformar ao protocolo de multiplexador ITU H.223, ou outros protocolos, como o protocolo de datagrama de usuário (UDP).

[0069] O padrão ITU-T H. 264/MPEG-4 (AVC) foi elaborado pelo Grupo de Experts em Codificação de Vídeo (VCEG) ITU-T juntamente com o Grupo de Experts em Imagem com Movimento (MPEG) ISO/IEC, como produto de uma parceria coletiva conhecida como Time de Vídeo Unido (JVT). Em alguns aspectos, as técnicas descritas nesta descrição podem ser aplicadas aos dispositivos que geralmente se conformam ao padrão H.264. O padrão H.264 está descrito na Recomendação H.264 ITU-T, Codificação de Vídeo Avançada para serviços audiovisuais genéricos, pelo Grupo de Estudo ITU-T, datada de Março de 2005, e pode ser referido aqui tanto como padrão H.264 ou especificação H.264 ou o padrão ou especificação AVC/H.264. O Time de Vídeo Unido (JVT) continua trabalhando em extensões para H.264/MPEG-4 AVC.

[0070] O codificador de vídeo 20 e o decodificador de vídeo 30 pode, cada um, ser implementado como qualquer um dentre uma variedade de circuitos de codificador adequados, como um ou mais microprocessadores, processadores de sinais digitais (DSPs), circuitos integrados de aplicação específica (ASICs), arranjos de portas programadas em campo (FPGAs), lógica discreta, software, hardware, firmware ou qualquer combinação dos mesmos. Quando as técnicas são implementadas parcialmente em software, um dispositivo pode armazenar instruções para o software em um meio legível por computador não transitório adequado e realizar as instruções em hardware utilizando um ou mais processadores para realizar as técnicas desta descrição. Cada um entre codificador de vídeo de 20 e decodificador de vídeo 30 pode ser incluído em um ou mais codificadores ou decodificadores, ou cada um dos quais pode ser integrado como parte de um codificador/decodificador (CODEC) combinado em um respectivo dispositivo.

[0071] O JCT-VC está trabalhando no desenvolvimento do padrão HEVC. Os esforços de padronização de HEVC são baseados em um modelo de evolução de um codificador de vídeo referido como o Modelo de Teste HEVC (HM). O HM presume várias funcionalidades adicionais de codificadores de vídeo em relação aos dispositivos existentes de acordo com, por exemplo, o ITU-T H.264/AVC. Por exemplo, enquanto que o H.264 oferece nove modos de codificação intra-predição, o HM pode fornecer até trinta e três modos de codificação intra-predição.

[0072] Em geral, o modelo de trabalho do HM descreve que um quadro de vídeo ou uma imagem pode ser dividido em uma sequência de treeblocks ou em unidades de codificação maiores (LCU) que incluem tanto amostras de

luminância quanto de crominância. Os dados de sintaxe dentro de um fluxo de bits podem definir um tamanho para a LCU, que é a maior unidade de codificação em termos de número de pixels. Uma fatia compreende um número de treeblocks consecutivos na ordem de codificação. Um quadro de vídeo ou imagem pode ser particionado 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 de quadtree compreende um nó por CU, com um nó raiz correspondente ao treeblock. Se uma CU é dividida em quatro sub-CUs, o nó correspondente à CU inclui quatro nós de folha, cada um dos quais corresponde a uma dentre as sub-CUs.

[0073] Cada nó da estrutura de dados quadtree pode fornecer dados de sintaxe para a CU correspondente. Por exemplo, um nó na quadtree pode compreender um indicador de divisão, indicando se a CU correspondente ao nó é dividida em sub-CUs. Elementos de sintaxe para uma CU podem ser definidos recursivamente e podem depender de se a CU é dividida em sub-CUs. Se uma CU não é mais dividida é referida como uma CU-folha. Nesta descrição, quatro sub-CUs de uma CU-folha serão também referidas como CUs-folha mesmo se não existir nenhuma divisão explícita da CU-folha original. Por exemplo, se uma CU no tamanho de 16x16 não é mais dividida, as quatro sub-CUs 8x8 serão também referidas como CUs-folha embora a CU 16x16 nunca tenha sido dividida.

[0074] Uma CU tem uma finalidade similar a um macrobloco do padrão H.264, exceto que uma CU não tem uma distinção de tamanho. Por exemplo, um treeblock pode ser dividido em quatro nós filho (também referidos como sub-CUs) e cada nó filho pode, por sua vez, ser um nó pai e ser divido em outros quatro nós filho. Um nó filho indivisível final, referido como um nó folha da quadtree, compreende um

nó de codificação, também referido como uma CU-folha. Os dados de sintaxe associados a um fluxo de bits codificado pode definir o número máximo de vezes que um treeblock pode ser dividido, referido como uma profundidade máxima de CU, e também pode definir um tamanho mínimo dos nós de codificação. Por conseguinte, um fluxo de bits também pode definir a menor unidade codificação (SCU). Esta descrição usa o termo "bloco" para se referir a qualquer uma dentre uma CU, PU ou TU, no contexto do HEVC, ou estruturas de dados similares no contexto de outros padrões (por exemplo, macroblocos e sub-blocos do mesmo no H.264/AVC).

[0075] Uma CU inclui um nó de codificação e unidades de predição (PUs) e unidades de transformadas (TUs) associadas com o nó de codificação. Um tamanho de CU corresponde a um tamanho do nó de codificação e a CU deve ter um formato de quadrado. O tamanho da CU pode variar de 8x8 pixels até o tamanho do treeblock com um máximo de 64x64 pixels ou maior. Cada CU pode conter uma ou mais PUs e uma ou mais TUs. Os dados de sintaxe associados a uma CU podem descrever, por exemplo, o particionamento da CU em uma ou mais PUs. Os modos de particionamento podem ser diferentes se a CU é codificada no modo skip ou no modo, codificada no modo intra-predição ou codificada no modo inter-predição. As PUs podem ser particionadas para não serem de forma quadrada. Os dados de sintaxe associados com uma CU podem também descrever, por exemplo, o particionamento da CU em uma ou mais TUs de acordo com um quadtree. Uma TU pode ter a forma quadrada ou não quadrada (por exemplo, retangular).

[0076] O padrão HEVC permite transformações de acordo com as TUs, que podem ser diferentes para diferentes CUs. As TUs são normalmente dimensionadas com base no tamanho de PUs dentro de uma determinada CU definida para

uma LCU particionada, embora isto não seja sempre o caso. As TUs são tipicamente menores ou do mesmo tamanho das PUs. Em alguns exemplos, amostras residuais correspondentes a uma CU podem ser subdivididas em unidades menores, utilizando uma estrutura de quadtree conhecida como "quadtree residual" (RQT). Os nós folha do RQT podem ser referidos como unidades de transformadas (TUs). Os valores da diferença de pixel associados às TUs podem ser transformados para produzir coeficientes de transformadas, que podem ser quantizados.

[0077] Uma CU-folha pode compreender uma ou mais unidades de predição (PUs). Em geral, uma PU representa uma área espacial correspondente à totalidade ou a uma parte da CU correspondente e pode compreender dados para recuperar uma amostra de referência para a PU. Além disso, uma PU compreende dados relacionados à predição. Por exemplo, quando a PU é codificada por intra-modo, os dados para a PU podem ser incluídos em uma quadtree residual (RQT), que pode compreender dados que descrevem um modo de intra-predição para uma TU correspondente à PU. Como outro exemplo, quando a PU é codificada por inter-modo, a PU pode compreender 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, uma componente horizontal do vetor de movimento, uma componente vertical do vetor do movimento, uma resolução para o vetor de movimento (por exemplo, precisão de um quarto de pixel ou precisão de um oitavo pixel), uma imagem de referência para a qual o vetor de movimento aponta, e/ou uma lista de imagens de referência (por exemplo, Lista 0, Lista 1 ou Lista C) para o vetor de movimento.

[0078] Uma CU-folha possuindo uma ou mais PUs também pode compreender uma ou mais unidades de

transformada (TUs). As unidades de transformada podem ser especificadas utilizando uma RQT (também referida como uma estrutura quadtree TU), como discutido acima. Por exemplo, um indicador de divisão pode indicar se uma CU-folha é dividida em quatro unidades de transformada. Então, cada unidade de transformada pode ser dividida ainda mais em outras sub-TUs. Quando uma TU não é mais dividida, pode ser referida como uma TU-folha. Geralmente, para intra-codificação, toda as TUs-folha pertencentes a uma CU-folha compartilham o mesmo modo de intra-predição. Ou seja, o mesmo modo de intra-predicãoo é geralmente aplicado para calcular valores preditos para todas TUs de uma CU-folha. Para a intra-codificação, um codificador de vídeo pode calcular um valor residual para cada TU-folha, utilizando o modo de intra-predição, como uma diferença entre a parte da CU correspondente à TU e o bloco original. Uma TU não é necessariamente limitada ao tamanho de uma PU. Assim, As TUs podem ser maiores ou menores do que uma PU. Para intra-codificação, uma PU pode ser colocado com uma correspondente TU-folha para a mesma CU. Em alguns exemplos, o tamanho máximo de uma TU-folha pode corresponder ao tamanho da CU-folha correspondente.

[0079] Além disso, as TUs de CUs-folha podem também ser associadas com estruturas de dados quadtree, referidas como quadtrees residuais (RQTs). Ou seja, a CU-folha pode compreender uma quadtree indicando como a CU-folha é particionada em TUs. O nó raiz de uma quadtree TU geralmente corresponde a uma CU-folha, enquanto o nó raiz de uma quadtree CU geralmente corresponde a uma treeblock (ou LCU). As TUs do RQT que não são divididas são referidas como TUs-folha. Em geral, esta descrição utiliza os termos CU e TU para se referir a CU-folha e TU-folha, respectivamente, salvo indicação em contrário.

[0080] Uma sequência de vídeo normalmente compreende uma série de quadros de vídeo ou de imagens. Um grupo de imagens (GOP) geralmente compreende uma série de uma ou mais dentre as imagens de vídeo. Um GOP pode incluir os dados de sintaxe em um cabeçalho do GOP, em um cabeçalho de uma ou mais dentre as imagens, ou em outro lugar, que descreve um número de imagens compreendidas no GOP. Cada fatia de uma imagem pode compreender dados de sintaxe da fatia que descrevem um modo de codificação para a respectiva fatia. O codificador de vídeo 20 normalmente opera em blocos de vídeo dentro de fatias de vídeo individuais para codificar os dados de vídeo. Um bloco de vídeo pode corresponder a um nó de codificação dentro de uma CU. Os blocos de vídeo podem ter tamanhos fixos ou variados, e podem diferenciar em tamanho de acordo com um padrão de codificação especificado.

[0081] Como um exemplo, a HM suporta a predição em vários tamanhos de PU. Considerando que o tamanho de uma CU particular é 2Nx2N, a HM suporta intra-predição em tamanhos de PU de 2Nx2N ou NxN, e inter-predição em tamanhos de PU simétricas 2Nx2N, 2NxN, Nx2N ou NxN. A HM também suporta particionamento assimétrico para inter-predição em tamanhos de PU de 2NxnU, 2NxnD, nLx2N e nRx2N. No particionamento assimétrico, uma direção de uma CU não é particionada, enquanto a outra direção é particionada em 25% e 75%. A parte da CU correspondente à partição de 25% é indicada por um "n" seguido de uma indicação de “acima”, “abaixo”, “esquerda”, ou “direita”. Assim, por exemplo, "2NxnU" refere-se a uma CU 2Nx2N que é particionada horizontalmente com uma PU 2Nx0,5N no topo e uma PU 2Nx1,5N na parte inferior.

[0082] Nesta descrição, “NxN” e “N por N” podem ser utilizados intercambiavelmente para se referir 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 de 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 geralmente tem 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 organizados em linhas e 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.

[0083] Após a codificação intra-preditiva ou inter-preditiva utilizando 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 descrevendo um método ou modo de geração de dados de pixel preditivos no domínio espacial (também referido como o domínio de pixel) e as TUs podem compreender coeficientes no domínio de transformada após a aplicação de uma transformada, por exemplo, uma transformada discreta de cosseno (DCT), uma transformada de números inteiros, uma transformada wavelet ou uma transformada conceitualmente similar para dados de vídeo residuais. Os dados residuais podem corresponder às diferenças de pixels entre pixels da imagem não codificada e os valores de predição correspondentes às PUs. O codificador de vídeo 20 pode formar as TUs, incluindo os dados residuais para a CU, e então transformar as TUs para produzir os coeficientes de transformada para a CU.

[0084] Após quaisquer transformadas para produzir coeficientes de transformada, o codificador de vídeo 20 pode realizar quantização dos coeficientes de transformada.

Quantização geralmente se refere a um processo no qual os coeficientes de transformada são quantizados para possivelmente reduzir a quantidade de dados utilizados para representar os coeficientes, fornecendo compressão adicional. O processo de quantização pode reduzir a profundidade de bit associada a alguns ou a todos os coeficientes. Por exemplo, um valor de n bits pode ser arredondado para baixo para um valor de m bits durante a quantização, onde n é maior que m.

[0085] Após a quantização, o codificador de vídeo pode varrer os coeficientes de transformada, produzindo um vetor unidimensional a partir da matriz bidimensional, incluindo os coeficientes de transformada quantizados. A varredura pode ser designada para colocar os coeficientes de energia mais alta (e, portanto, de menor frequência) na frente da matriz e colocar os coeficientes de energia mais baixa (e, portanto, de maior frequência) na parte de trás da matriz. Em alguns exemplos, o codificador de vídeo 20 pode utilizar uma ordem de varredura predefinida para varrer os coeficientes de transformada quantizados para produzir um vetor serializado que pode ser codificado por entropia. Em outros exemplos, o codificador de vídeo 20 pode realizar uma varredura adaptativa. Após a varredura dos coeficientes de transformada quantizados para formar um vetor unidimensional, o codificador de vídeo 20 pode codificar por entropia o vetor unidimensional, por exemplo, de acordo com 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 por entropia de particionamento de intervalo de probabilidade (PIPE) ou outra metodologia de codificação por entropia. O codificador de vídeo 20 pode

também codificar por entropia os elementos de sintaxe associados com os dados de vídeo codificados para uso pelo decodificador de vídeo 30 para decodificar os dados de vídeo.

[0086] Para realizar a CABAC, o codificador de vídeo 20 pode atribuir um contexto dentro de um modelo de contexto para um símbolo a ser transmitido. O contexto pode estar relacionado a, por exemplo, se os valores vizinhos do símbolo são não-nulos ou não. Para realizar a CAVLC, o codificador de vídeo 20 pode selecionar um código de comprimento variável para um símbolo ser transmitido. Palavras código em VLC podem ser construídas de tal forma que códigos relativamente menores correspondam aos símbolos mais prováveis, enquanto que códigos maiores correspondam a símbolos menos prováveis. Desta forma, a utilização de VLC pode alcançar uma economia de bits, por exemplo, utilizando palavras código de comprimentos iguais para cada símbolo a ser transmitido. A determinação da probabilidade pode ser baseada em um contexto atribuído ao símbolo.

[0087] De acordo com as técnicas desta descrição, um codificador de vídeo, tal como o codificador de vídeo 20 ou decodificador de vídeo 30, pode determinar um modo de codificação utilizado para codificar os dados de vídeo. O modo de codificação pode ser um de um modo de codificação com perdas ou um modo de codificação sem perdas. O codificador de vídeo pode determinar um processo de transformada de espaço-cor dependente do modo de codificação utilizado para codificar os dados de vídeo. O codificador de vídeo pode aplicar o processo de transformada de espaço-cor na codificação dos dados de vídeo. Ao decodificar os dados de vídeo, independente do fato de o modo de codificação ser o modo de codificação com perdas ou o modo de codificação sem perdas, o codificador

de vídeo pode aplicar o mesmo processo de transformada inversa de espaço-cor em um loop de decodificação do processo de codificação.

[0088] Um codificador de vídeo, tal como um codificador de vídeo 20 ou decodificador de vídeo 30, pode realizar qualquer uma das técnicas, tal como descrito com relação às Figs. 1-9. Usando a mesma matriz de transformada inversa de espaço-cor, um decodificador de vídeo pode ser capaz de decodificar os dados de vídeo de modo mais eficiente quando a transformação espaço-cor é utilizada. Em casos diferentes, pode ser mais eficiente para um codificador de vídeo utilizar um modo de codificação com perdas. Em outros casos, pode ser mais eficiente para um decodificador de vídeo utilizar um modo de codificação sem perdas. Para qualidade de imagem mais elevada, um modo de codificação sem perdas pode ser aplicada no lado de decodificação, e uma matriz de codificação sem perdas pode ser usada no lado do decodificador, independentemente de os dados de vídeo serem codificados com um modo de codificação com perdas ou um modo de codificação sem perdas. Como tal, pode adicionar-se a eficiência de um decodificador de vídeo para implementar a mesma matriz de transformada inversa de espaço-cor sem perdas independentemente do fato de os dados de vídeo serem codificados com um modo de codificação com perdas ou com um modo de codificação sem perdas, removendo o passo de determinar se o dados de vídeo são codificados com um modo de codificação com perdas ou um modo de codificação sem perdas, aumentando assim a eficiência de codificação global do sistema e redução do consumo de energia.

[0089] Uma variedade de elementos de sintaxe pode ser usada em alinhamento com as técnicas da presente descrição atual. Estes elementos de sintaxe podem incluir:

seq_parameter_set_rbsp( ) {	Descritor
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_rdpcm_enabled_flag	u(1)
explicit_rdpcm_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) {	Descritor
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]	ae(v)
if(CromaArrayType = = 3)	
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
intra_croma_pred_mode[x0 + i][y0 + j]	ae(v)
else if(CromaArrayType != 0)	
intra_croma_pred_mode[x0][y0]	ae(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	ae(v)
if(rqt_root_cbf) {	
if (color_transform_enabled_flag) (new)	
color_transform_flag[x0][y0] (new)	ae(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)	
}	
}	
}	
}	

Tabela 1

[0090] Quando o elemento de sintaxe color_transform_enabled_flag é igual a 1, o processo de transformada de espaço-cor em loop pode ser invocado no processo de decodificação. Quando o elemento de sintaxe color_transform_enabled_flag é igual a 0, a transformada de espaço-cor 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 inferido para ser igual a 0.

[0091] Quando o elemento de sintaxe color_transform_flag[x0][y0] é igual a 1, a unidade de codificação atual pode ser codificada com transformada de espaço-cor. Quando o elemento de sintaxe color_transform_flag[x0][y0] é igual a 0, a unidade de codificação atual é codificada sem transformada de espaço-cor. Quando o elemento sintaxe não está presente, o valor do elemento de sintaxe color_transform_flag pode ser inferido para ser igual a 0. Os índices de matriz x0 e y0 podem especificar o local (x0, y0) de uma amostra de luminância superior à esquerda do bloco de codificação considerado em relação à amostra de luminância superior à esquerda da imagem.

[0092] Em alguns exemplos, codificador de vídeo, tal como o codificador de vídeo 20 ou decodificador de vídeo 30, pode determinar se deve utilizar a conversão de espaço-cor 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 de conversão de espaço-cor. O codificador de vídeo pode aplicar uma matriz de transformada de espaço-cor para 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 a conversão de espaço-cor. 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-cor. O codificador de vídeo pode aplicar uma matriz de transformada inversa de espaço-cor para 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-cor.

[0093] Em um exemplo, para CUs intra-codificadas, um indicador (flag) pode ser diretamente sinalizado. Alternativamente, para CUs intra-codificadas, um indicador pode ser sinalizado apenas quando a CU não é modo I-PCM. Em outro exemplo, para as CUs inter-codificadas e/ou CUs intra BC-codificadas, um indicador pode ser sinalizado apenas quando não existem coeficientes não-nulos na CU corrente, isto é, a rqt_root_cbf é igual a 1. Em outro exemplo, o indicador não é sinalizado quando a CU está codificada com o modo de paleta. Em todos os exemplos acima, quando o indicador de transformada de cor decodificada é igual a 0, ou o indicador não está presente no fluxo de bits para uma CU, o processo de transformada de espaço-cor pode ser ignorado.

[0094] Além disso, um indicador em qualquer cabeçalho de conjunto de parâmetros de sequência (SPS)/conjunto de parâmetros de imagem (PPS)/fatia pode ser sinalizado para controlar o uso da transformada de espaço-cor. Quando o indicador no cabeçalho SPS/PPS/fatia é igual a 0, a sinalização de indicador de nível CU pode ser ignorada em todas as CUs no correspondente sequência/imagem/fatia, respectivamente. Alternativamente, a transformada de espaço-cor pode ser realizada dm nível PU ou nível TU em que cada um de PU/TU tem um indicador para indicar o uso da transformada de espaço-cor.

[0095] Em um exemplo, uma transformada YCoCg modificada pode ser aplicada no processo de codificação e de decodificação. A transformada YCoCg modificada pode ser definida da seguinte forma:

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

Inversa: temp = Y-Cg;

G = (Y + Cg + deslocamento) >> 2;

B = (temp - Co + deslocamento) >> 2;

R = (temp + Co + deslocamento) >> 2, em que o deslocamento é igual a dois.

[0096] Alternativamente, a transformada YCoCg modificada pode ser definida como se segue:

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

Inversa: temp = Y-Cg;

G = (Y + Cg + deslocamento) >> 1;

B = (temp - Co + deslocamento) >> 1;

R = (temp + Co + deslocamento) >> 1;

onde o deslocamento é igual a 1 ou 0.

[0097] O correspondente parâmetro de quantização utilizado pelas CUs ou blocos que são codificados com transformada de cor pode ser inferido para ser igual a (dQP + 2), enquanto o utilizado pelas CUs ou blocos sem transformada de cor pode ser inferido para ser igual a dQP. Simultaneamente, a profundidade de bits pode ser aumentada em 1 em ambos os processos de quantização e de transformada.

[0098] Alternativamente, são aplicadas duas etapas de transformada direta/inversa com a transformada YCoCg original inalterada mais um processo de normalização. Para a transformada direta, a transformada direta YCoCg original é aplicada em primeiro lugar. Então, para cada componente i, ou seja, Y, Co e Cg, a componente é redefinida para (i * fator direto + deslocamento) >> BIT_PRECISION onde BIT_PRECISION é um inteiro não sinalizado e o fator direto depende de BIT_PRECISION. Em um exemplo, o fator direto (forward_factor) é igual a 1/√6*(1 << BITS_TRANS) + 0,5,

BIT_PRECISION é igual a 15, o deslocamento é igual a (1 << (BIT_PRECISION - 1)).

[0099] Para a transformada inversa, a transformada inversa YCoCg original é aplicada em primeiro lugar. Então, para cada componente i, ou seja, Y, Co e Cg, a componente é redefinida para (i * fator inverso + deslocamento)>> BIT_PRECISION onde BIT_PRECISION é o mesmo utilizado na transformada direta, e o fator inverso (backward_factor) é dependente de BIT_PRECISION. Em um exemplo, fator inverso é igual a √6/4*(1 << BITS_TRANS) + 0,5, o deslocamento é igual a (1 <<(BIT_PRECISION - 1)).

[0100] Alternativa/adicionalmente, o indicador CU_level pode ser sinalizado apenas quando pelo menos um dos indicadores de bloco codificados em três componentes de cor (ou seja, cbf_luma, cbf_cb e cbf_cr) é igual a 1. Alternativamente, o resíduo modificado após a transformada de espaço-cor pode ser adicionalmente modificado para ter certeza de que a faixa de resíduo nas CUs ou blocos com transformada de espaço-cor é a mesma do resíduo nas CUs/blocos sem transformada de espaço-cor. Em um exemplo, uma operação de clipe é aplicada.

[0101] A matriz de transformada de espaço-cor pode ser independente dos pixels reconstruídos. Por outro lado, pode ser dependente do modo de codificação com perdas ou sem perdas. Em um exemplo, quando a CU é codificada com o modo com perdas, a YCoCg pode ser aplicada enquanto que a YCoCg-R é aplicada quando a CU é codificada com o modo sem perdas. Além disso, quando a YCoCg-R é aplicada, a profundidade de bits das componentes Co e Cg pode ser aumentada em 1. Em outro exemplo, a matriz de transformada pode depender dos modos intra, inter/intra BC. Neste caso, uma matriz predefinida para cada modo pode ser especificada tanto no codificador como no decodificador e a mesma matriz

pode ser utilizada tanto no codificador quanto no decodificador de acordo com o modo de codificação, quando o indicador CU_level é 1. Além disso, o indicador da CU_level pode ser sinalizado apenas quando pelo menos um dentre os indicadores de bloco codificado em três componentes de cor (ou seja, cbf_luma, cbf_cb e cbf_cr) é igual a 1.

[0102] As técnicas da presente descrição também podem compreender um método de derivação de matriz de transformada e sinalização, como em um nível de quadro. Em um exemplo, para cada quadro, uma matriz de transformada pode ser derivada com base nas estatísticas entre as componentes de cor na imagem original. A matriz de transformada pode ser codificada em um fluxo de bits e transmitida diretamente para o decodificador. Alternativamente, pode ser aplicada a predição da matriz de transformada entre quadros codificados consecutivamente. Por exemplo, um indicador pode ser transmitido para indicar se a matriz é a mesma que a utilizada no quadro anterior. Alternativamente, uma matriz de transformada pode ser derivada com base em um conjunto de quadros originais e sinalizada no fluxo de bits.

[0103] Em outro exemplo, um conjunto de matrizes de transformada é derivado e sinalizado na sintaxe de alto nível, como o PPS. Em um exemplo, a imagem é classificada em múltiplas regiões, com base nas características, e para cada região, uma matriz de transformada é derivada. As matrizes derivadas podem ser codificadas e sinalizadas na sintaxe de alto nível. O índice da matriz de transformada selecionada é adicionalmente sinalizado no nível de CU/TU/PU. Alternativamente, o índice da matriz de transformada selecionada é sinalizado no cabeçalho da fatia ou para cada região delineada (tile). Alternativamente, a

matriz de transformada pode ser derivada no cabeçalho da fatia ou para cada região delineada.

[0104] Quando a profundidade de bits das componentes de luminância e de crominância é diferente, a profundidades de bits de todas as componentes de cor pode ser modificada, em primeiro lugar, para ser a mesma e, então a transformada de cor pode ser aplicada subsequentemente. Alternativamente, o fluxo de bits pode se conformar a uma restrição que, quando a profundidade de bits das componentes de luminância e de crominância é diferente no fluxo de bits codificado, a transformada de cor deve ser desabilitada (isto é, o indicador color_transform_enabled igual a 0).

[0105] Diferentes QPs podem ser aplicadas às CUs/blocos, que são codificados com ou sem transformada de cor, e/ou a profundidade de bits utilizada no processo de quantização e transformada pode ser modificada devido a tal transformada de cor não normalizada. Em um exemplo, quando é utilizada a transformada YCoCg modificada acima, as seguintes etapas (ou seja, escalonamento, transformada e o processo de construção da matriz anterior ao processo de filtrar por desagrupamento) são modificadas com base no color_transform_flag.

[0106] Em um exemplo, quando é utilizada a transformada YCoCg modificada, conforme descrito em [0092], o processo de derivação do parâmetro de quantização correspondente e o processo de transformada inversa são modificados consequentemente. As entradas para este processo podem compreender uma localização de luminância (xCb, yCb) especificando a amostra superior esquerda do bloco de codificação de luminância atual com relação à amostra de luminância superior esquerda da imagem atual. Neste processo, a variável QpY, o parâmetro de quantização

de luminância Qp'Y e os parâmetros de quantização de crominância 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 atual com relação à amostra de luminância superior esquerda da imagem atual. As posições horizontal e vertical xQg e yQg podem ser definidas como igual a xCb-(xCb&((1 << Log2MinCuQpDeltaSize)-1)) e yCb-(yCb&((1 << Log2MinCuQpDeltaSize)-1)), respectivamente. O tamanho de luminância de um grupo de quantização, Log2MinCuQpDeltaSize, pode determinar o tamanho de luminância da menor área dentro de um bloco de árvore de codificação que compartilha o mesmo qPY_PRED.

[0107] A variável QpY pode ser derivada da seguinte forma:

QpY = ((qPY PRED + CuQpDeltaVal + 52 + 2 * QpBdOffsetY)% (52 + QpBdOffsetY)) - QpBdOffsetY

[0108] O parâmetro de quantização de luminância Qp'Y pode ser derivado da seguinte forma:

Qp'Y = QpY + QpBdOffsetY

[0109] Quando o elemento de sintaxe CromaArrayType não é igual a 0, pode-se aplicar o seguinte:

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

qPiCb = Clip3(-QpBdOffsetC, 57, QpY + pps_cb_qp_offset + slice_cb_qp_offset + CuQpAdjValCb)

qPiCr = Clip3(-QpBdOffsetC, 57, QpY + pps_cr_qp_offset + slice_cr_qp_offset + CuQpAdjValCr)

[0110] Se o elemento de sintaxe CromaArrayType é igual a 1, as variáveis qPCb e qPCr podem ser definidas igual 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 definidas iguais a Min(qPi, 51), com base no índice qPi igual a qPiCb e qPiCr, respectivamente.

qPi	< 30	30	31	32	33	34	35	36	37	38	39	40	41	42	43	> 43
Qp_C	= qPi	29	30	31	32	33	33	34	34	35	35	36	36	37	37	= qPi − 6

Tabela 2

[0111] Os parâmetros de quantização de crominância para as componentes, Cb e Cr, Qp'Cb e Qp'Cr, podem ser derivados da seguinte forma:

Qp'Cb = qPCb + QpBdOffsetC

Qp'Cr = qPCr + QpBdOffsetC

[0112] Quando color_transform_flag é igual a 1, pode-se aplicar o seguinte:

Qp 'Y = (Qp 'Y + 8)

Quando CromaArrayType ≠ 0, Qp'Cb = (Qp 'cb + 8) e Qp'Cr = (Qp 'cr + 8)

BitDepthY = (BitDepthY + 2)

BitDepthC = (BitDepthC + 2)

[0113] As entradas para este processo podem compreender uma localização de luminância (xTbY, yTbY) especificando a amostra superior esquerda do bloco de transformada de luminância atual com relação à amostra de luminância superior esquerda do quadro atual, uma variável nTbS especificando o tamanho do bloco de transformada atual, uma variável cIdx especificando a componente de cor do bloco atual e uma variável qP especificando o parâmetro de quantização. A saída deste processo pode incluir a matriz d (nTbS)x(nTbS) de coeficientes de transformada escalonados com elementos d[x][y].

[0114] As variáveis log2TransformRange, bdShift, coeffJVIin e coeffMax podem ser derivadas da seguinte forma:

Quando color_transform_flag = 1, pode-se aplicar o seguinte:

CoeffCTMinY = -(1<<(extended_precision_processing_flag ? Max(15,BitDepthY+6):15))

CoeffCTMinC = -(1<<(extended_precision_processing _fag ? Max(15,BitDepthC+6):15))

CoeffCTMaxY = (1<<(extended-precision_processingj_flag ? Max(15,BitDepthY+6):15)) - 1

CoeffCTMaxC = (1<<(extended_precision_processingj_flag ? Max(15,BitDepthC+6):15)) - 1

Se cIdx = 0,

log2TransformRange = extended_precision_processing_flag ? Max(15,BitDepthY+6):15

bdShift = BitDepthY + Log2(nTbS) + 10-log2TransformRange

coeffMin = (color_transform_flag ? CoeffCTMinY : CoeffMinY)

coeffMax = (color_transform_fag ? CoeffCTMaxY : CoeffMaxY)

Caso contrário,

log2TransformRange = extended_precision_processing_flag ? Max(15,BitDepthC+6): 15

bdShift = BitDepthC + Log2(nTbS) + 10-log2TransformRange

coeffMin = (color_transform_flag ? CoeffCTMinC:CoeffMinC)

coeffMax = (color_transform_fag ? CoeffCTMaxC:CoeffMaxC)

[0115] A lista levelScale[] pode ser especificada como levelScale[k] = {40, 45, 51, 57, 64, 72} com k = 0..5.

[0116] Para a derivação dos coeficientes de transformada escalonados d[x][y] com x = 0..nTbS - 1, y = 0..nTbS - 1, pode-se aplicar o seguinte:

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

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

scaling_list_enabled_flag = 0.

transform_skip_flag[xTbY][yTbY] = 1 e nTbS>4.

Caso contrário aplica-se o seguinte:

m[x][y] = ScalingFactor[sizeId][matrixId][x][y] (8-283)

[0117] Quando o elemento de sintaxe sizeId é especificado para o tamanho da matriz de quantização igual (nTbS)x(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 derivado da seguinte forma:

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

[0118] Quando o color_transform_flag é igual a 1, pode-se aplicar o seguinte:

BitDepthY = BitDepthY - 2

BitDepthC = BitDepthC – 2

[0119] Em um exemplo, quando a transformada de espaço-cor é aplicada para intra-modos, o resíduo é derivado do domínio espaço-cor convertido. Ou seja, antes de invocar o processo de predição de intra-amostra, os pixels vizinhos da CU atual podem ser primeiramente convertidos para outro espaço-cor sub-ótimo com uma transformada direta modificada YCoCg ou YCoCg-R. Os pixels vizinhos modificados podem ser utilizados para derivar o preditor da CU atual. O resíduo (ou seja, o erro de predição) é derivado da CU atual e dos pixels vizinhos da CU atual em um domínio sub-ótimo. O resíduo é codificado da mesma forma que o fluxo de codificação convencional, como predição residual inter-componente, transformada,

quantização e codificação por entropia. Após a codificação por entropia, os coeficientes de transformada podem ainda ser modificados com a YCoCg ou YCoCg-R inversa. Os coeficientes de transformada modificados podem então ser adicionados ao preditor para derivar os pixels reconstruídos da CU atual. Depois que o processo de reconstrução é invocado, a transformada inversa YCoCg ou YCoCg-R pode ser aplicada para os pixels vizinhos modificados. Portanto, no lado do decodificador, antes de invocar o processo de reconstrução, pode-se aplicar um processo adicional onde a transformada inversa de espaço-cor é aplicada para os coeficientes de transformada derivados.

[0120] Quando a transformada de espaço-cor é aplicada para intra-modos, em vez de gerar o resíduo subtraindo-se o preditor do bloco atual no domínio de cor convertido, o resíduo pode ser gerado utilizando os pixels antes da transformada de espaço-cor, seguido pela transformada de espaço-cor. Neste caso, os intra-modos de luminância e de crominância podem ser definidos para o modo idêntico. Alternativamente, a sinalização do modo de crominância pode ser pulada. Alternativamente, o indicador de transformada de cor pode ser sinalizado apenas quando os intra-modos de luminância e crominância forem os mesmos.

[0121] Conforme descrito acima, em alguns exemplos, os métodos de codificação sem perdas e com perdas podem compartilhar a mesma matriz, por exemplo, YCoCg-R. Adiciona ou alternativamente, quando a transformada YCoCg-R é aplicada para codificação com perdas, a profundidade de bits de componentes de crominância de blocos com transformada de cor pode ser aumentada em 1 em comparação com os blocos sem transformada de cor. O QP de componente de luminância de blocos com transformada de cor pode ser

igual ao dos blocos sem transformada de cor menos 4. Em um exemplo, os dois QP de componentes de crominância de blocos com transformada de cor podem ser iguais ao QP da componente de crominânica/luminância dos blocos sem transformada de cor mais 2. Em um exemplo, o QP das componentes de crominância Cg de blocos com transformada de cor podem ser iguais aos QP da componente de crominânica/luminância de blocos sem transformada de cor mais 2. O QP das componentes Co de crominância de blocos com transformada de cor podem ser iguais ao QP das componentes de crominânica/luminância de blocos sem transformada de cor mais 3.

[0122] Em tais exemplos, quando a YCoCg-R é utilizada para ambos os modo com perdas e sem perdas, se o bloco atual é codificado com perdas, a profundidade de bits de luminância e de crominância 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 direta de YCoCg-R, as componentes Co e Cg podem ser escalonadas por desvio à direita de N bits a fim de reduzir o aumento da profundidade de bits devido à transformada YCoCg-R direta e tornar a precisão da profundidade de bits igual a precisão da profundidade de bits da componente Y. Além disso, antes de realizar a transformada inversa YCoCg-R, as componentes Co e Cg podem ser modificadas com desvio à esquerda de N bits. Em um exemplo, N é igual a 1.

[0123] Em um exemplo, pode-se considerar que as entradas de profundidade de bits de componentes de luminância e de crominância são as mesmas. Para o modo de codificação com perdas, podem ser aplicadas operações de desvio para as duas componentes de crominância (i.e., Cg, Co) após a transformada direta e antes da transformada

inversa. Em um exemplo, o seguinte processo é aplicado da seguinte forma:

1. YCoCg-R direta não é alterada: $$\begin{matrix} \mathrm{Co} = \mathrm{R} - \mathrm{B} \\ t = B + \lfloor Co/2 \rfloor \\ Cg = G - t \\ Y = t + \lfloor Cg/2 \rfloor \end{matrix},$$

2. Se o bloco atual é codificado no modo com perdas, pode-se aplicar ainda o seguinte:

```
Cg = Cg>>1;
Co = Co>>1;
```

[0124] Em outro exemplo, um deslocamento pode ser considerado no processo de desvio à direita. Por exemplo, as equações destacadas acima podem ser substituídas por:

```
Cg = (Cg + 1)>>1;
Co = (Co + 1)>>1;
```

[0125] Antes de invocar a transformada inversa de YCoCg-R, pode-se aplicar o seguinte:

Se o bloco atual é codificado no modo com perdas, pode-se aplicar o seguinte:

```
Cg = Cg<<1;
Co = Co<<1;
```

E YCoCg-R inversa pode permanecer inalterada:

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

[0126] Em tais exemplos, um processo de modificação residual para transformar blocos utilizando a transformada de cor adaptativa pode ser invocado quando CromaArrayType é igual a 3. As entradas para este processo podem compreender uma variável blkSize especificando o tamanho do bloco, uma matriz (blkSize)x(blkSize) de amostras residuais rY de luminância com elementos rY[x][y], uma matriz (blkSize)x(blkSize) de amostras residuais rCb de crominância com elementos rCb[x][y] e uma matriz

(blkSize)x(blkSize) de amostras residuais rCr de crominância com elementos rCr [x][y]. As saídas para este processo podem incluir uma matriz (blkSize)x(blkSize) modificada de amostras residuais r_Y de luminância, uma matriz (blkSize)x(blkSize) modificada de amostras residuais r_{Cb} de crominância e uma matriz (blkSize)x(blkSize) modificada de amostras residuais r_{Cr} de crominância.

[0127] As matrizes (BlkSize)x(blkSize) das amostras residuais r_Y, r_{Cb} e r_{Cr}, com x = 0..blkSize - 1, y = 0..blkSize - 1 podem ser alteradas da seguinte forma:

Se cu_transquant_bypass_flag = 0, $r_{Cb}[x][y]=r_{Cb}[x][y]<<1$

e $r_{Cr}[x][y]=r_{Cr}[x][y]<<1$

$tmp = r_Y[x][y]-(r_{Cb}[x][y]>>1)$

$r_Y[x][y]=tmp+r_{Cb}[x][y]$

$r_{Cb}[x][y]=tmp-(r_{Cr}[x][y]>>1)$

$r_{Cr}[x][y]=r_{Cb}[x][y]+r_{Cr}[x][y]$

[0128] Alternativamente, 'se cu_transquant_bypass _flag é igual a 0, rCb[x][y]=rCb[x][y] <<1 e rCr[x][y]= rCr[x][y]<<1' pode ser substituída por 'rCb[x][y]=rCb[x] [y]<<(1-cu_transquant_bypass_flag) e rCr[x][y]=rCr[x][y]<< (1-cu_transquant_bypass_flag)'.

[0129] Em outro exemplo de um método de codificação com perdas e de um método de codificação sem perdas compartilhando uma matriz de transformada de cor, as entradas das profundidades de bits das componentes de luminância e de crominância podem ser diferentes. Nesse exemplo, quando as entradas das profundidades de bits de componentes de luminância e de crominância são diferentes e YCoCg-R é utilizada para ambos os modos de codificação com perdas e sem perdas, antes da transformada inversa, pelo menos um dos valores das componentes Y, Co, Cg pode primeiramente ser desviado N bits para fazer com que todas as três componentes tenham a mesma precisão de profundidade

de bits. Adicionalmente, um ou mais bits podem ser desviados para as duas componentes de crominância (i.e., Co e Cg). Pelo menos um dentre os valores de predição de todas as três componentes pode ser modificado para ter a mesma precisão de profundidade de bits antes de invocar a transformada YCoCg-R direta. E após a transformada direta, as duas componentes de crominância podem ser desviadas à direita em 1 bit. Para os modos de codificação sem perdas, a diferença de profundidade de bits pode não ser considerada e, portanto, a transformada de espaço-cor adaptativa pode ser desabilitada. Em um exemplo, o fluxo de bits pode compreender uma restrição que, quando a profundidade de bits das componentes de luminância e de crominância são diferentes no fluxo de bits codificado, a transformada de cor deve ser desabilitada (ou seja, o color transform_enabled_flag é igual a 0).

[0130] Neste exemplo, um processo de modificação residual para blocos de transformada, utilizando transformada de cor adaptativa pode ser invocado quando CromaArrayType é igual a 3. As entradas para este processo podem compreender uma variável blkSize especificando o tamanho do bloco, uma matriz (blkSize)x(blkSize) de amostras residuais rY de luminância com elementos rY[x][y], uma matriz(blkSize)x(blkSize) de amostras de residuais rCb de crominância com elementos rCb[x][y] e uma matriz (blkSize)x(blkSize) de amostras residuais rCr de crominância com elementos rCr[x][y]. As saídas para este processo podem incluir uma matriz (blkSize)x(blkSize) modificada de amostras residuais rY de luminância, uma matriz (blkSize)x(blkSize) modificada de amostras de residuais rCb de crominância e uma matriz (blkSize)x(blkSize) modificada de amostras residuais rCr de crominância.

[0131] As variáveis deltaBDy e deltaBDc podem ser derivadas da seguinte forma:

```
BitDepthMax= Max(BitDepthY, BitDepthC)
deltaBDY = cu_transquant_bypass_flag ? 0:BitDepthMax -
BitDepthY
deltaBDC = cu_transquant_bypass_flag ? 0:BitDepthMax –
BitDepthC
OY = cu_transquant_bypass_flag||(BitDepthMax == BitDepthY)
? 0:1<<(deltaBDY – 1)
OC = cu_transquant_bypass_flag||(BitDepthMax == BitDepthc)
? 0:1<<(deltaBDc – 1)
```

[0132] As matrizes (blkSize)x(blkSize) de amostras residuais rY, rCb e rCr com x = 0..blkSize-1, y = 0..blkSize-1 podem ser modificadas como segue:

As amostras residuais rY[x][y],rCb[x][y] e rCr [x][y] podem ser modificadas como segue:

```
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
```

[0133] Nos exemplos acima, a função Max pode ser utilizada para derivar o maior valor entre duas variáveis. Como alternativa, pelo menos, um dos deslocamentos arredondados O_Y e O_c pode ser definido como igual a 0.

[0134] Alternativamente, mesmo para codificação sem perdas, quando a profundidade de bits de componentes de luminância e de crominância é diferente, as componentes podem ser desviadas para a mesma precisão de profundidade de bits, se necessário. Do lado do codificador, após a transformada direta, pode não haver necessidade de desviar de volta para a precisão de profundidade de bits original. Em outras palavras, o resíduo codificado das três componentes pode ser da mesma profundidade de bits para se ter a certeza de que é a codificação sem perdas. Diferentemente da codificação com perdas, as entradas não podem ser modificadas antes de invocar a transformada inversa YCoCg-R para codificação sem perdas. O desvio à direita ainda pode ser necessário para ter a certeza de que as saídas estão na mesma profundidade de bits que as entradas originais.

[0135] Neste exemplo, um processo de modificação resíduo para transformar blocos utilizando transformada de cor adaptativa pode ser invocado quando CromaArrayType é igual a 3. As entradas para este processo podem compreender uma variável blkSize especificando o tamanho do bloco, uma matriz (blkSize)x(blkSize) de amostras residuais rY de luminância com os elementos rY[x][y], uma matriz (blkSize)x(blkSize) de amostras residuais rCb de crominância com os elementos rCb[x][y] e uma matriz (blkSize)x(blkSize) de amostras residuais rCr de crominância com os elementos rCr[x][y]. As saídas para este processo podem compreender uma matriz (blkSize)x(blkSize) modificada de amostras de residuais rY de luminância, uma matriz (blkSize)x(blkSize) modificada de amostras residuais rCb de crominância e uma matriz (blkSize)x(blkSize) modificada de amostras residuais rCr de crominância.

[0136] As variáveis deltaBDy e deltaBDc podem ser derivadas da seguinte forma:

```
BitDepthMax = Max(BitDepthY, BitDepthC)
deltaBDY = BitDepthMax-BitDepthY
deltaBDC = BitDepthMax-BitDepthC
OY =(BitDepthMax == BitDepthY)?0:1<<(deltaBDY-1)
OC = (BitDepthMax == BitDepthc)?0:1<<(deltaBDc-1)
```

[0137] As matrizes(blkSize)x(blkSize) de amostras residuais rY, rCb e rCr, com x = 0..blkSize - 1, y = 0..blkSize-1 podem ser modificadas do seguinte modo:

```
Quando cu_transquant_bypass_flag é igual a 0, as amostras
residuais rY[x][y], rCb[x][y] e rCr[x][y] podem ser
modificadas 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
```

[0138] Alternativamente, um entre Oγ e Oc pode ser igual a 0.

[0139] O codificador de vídeo 20 pode ainda enviar dados de sintaxe, como dados de sintaxe baseados em blocos, dados de sintaxe baseados em quadro e dados de sintaxe baseados em GOP, para o decodificador de vídeo 30, por exemplo, em um cabeçalho de quadro, um cabeçalho de bloco, um cabeçalho da fatia ou um cabeçalho do GOP. Os

dados de sintaxe do GOP podem descrever um número de quadros no respectivo GOP, e os dados de sintaxe do quadro podem indicar um modo de codificação/predição utilizado para codificar o quadro correspondente.

[0140] Cada um entre o codificador de vídeo 20 e o decodificador de vídeo 30 pode ser implementado como qualquer um dentre a variedade de circuitos de codificador ou decodificador adequados, conforme o caso, como um ou mais microprocessadores, processadores de sinais digitais (DSPs), circuitos integrados de aplicação específica (ASICs), arranjos de portas programáveis em campo (FPGAs), circuitos de lógica discreta, software, hardware, firmware ou quaisquer combinações dos mesmos. Cada codificador de vídeo 20 ou decodificador de vídeo 30 pode ser incluído em um ou mais codificadores ou decodificadores, ou cada um dos quais pode ser integrado como parte de um codificador/decodificador (CODEC) de vídeo combinado. Um dispositivo compreendendo codificador de vídeo 20 e/ou decodificador de vídeo 30 pode compreender um circuito integrado, um microprocessador, e/ou um dispositivo de comunicação sem fio, como um telefone celular.

[0142] A Fig. 2 é um diagrama de blocos ilustrando um exemplo de codificador de vídeo 20 que pode implementar técnicas para codificar blocos de vídeo utilizando um processo de conversão de espaço-cor. O codificador de vídeo 20 pode realizar intra-codificação e inter-codificação de blocos de vídeo dentro de fatias de vídeo. A intra-codificação se baseia na predição espacial para reduzir ou remover redundância espacial no vídeo dentro de um determinado quadro de vídeo ou imagem. A inter-codificação se baseia na predição temporal para reduzir ou remover redundância temporal em vídeo dentro de quadros ou imagens adjacentes de uma sequência de vídeo. O

intra-modo (modo I) pode se referir a qualquer um dentre os vários modos de codificação com base espacial. Os inter-modos, tais como predição unidirecional (modo P) ou bi-predição (modo B), podem se referir a qualquer um dentre os vários modos de codificação com base temporal.

[0142] Como mostrado na Fig. 2, o codificador de vídeo 20 recebe um bloco de vídeo atual dentro de um quadro de vídeo a ser codificado. No exemplo da Fig. 2, o codificador de vídeo 20 compreende a unidade de seleção de modo 40, a memória de imagem de referência 64, o somador 50, a unidade de processamento de transformada 52, a unidade de quantização 54 e a unidade de codificação por entropia 56. A unidade de seleção de modo 40, por sua vez, compreende a unidade de compensação de movimento 44, a unidade de estimação de movimento 42, a unidade de intra-predição 46 e a unidade de particionamento 48. Para a reconstrução do bloco de vídeo, o codificador de vídeo 20 também compreende a unidade de quantização inversa 58, a unidade de transformada inversa 60 e o somador 62. Um filtro de desagrupamento (não mostrado na Fig. 2) também pode ser incluído para filtrar limites do bloco para remover artefatos de blocagem do vídeo reconstruído. Se desejado, o filtro de desagrupamento normalmente pode filtrar a saída do somador 62. Filtros adicionais (em loop ou após o loop) também podem ser utilizados além do filtro de desagrupamento. Esses filtros não são mostrados por questões de concisão, mas se desejado, pode-se filtrar a saída do somador 50 (como um filtro em loop).

[0143] Durante o processo de codificação, o codificador de vídeo 20 recebe um quadro fatia de vídeo a ser codificado. O quadro ou a fatia pode ser dividido em múltiplos blocos de vídeo. A unidade de estimação de movimento 42 e a unidade de compensação de movimento 44

realizam a codificação inter-preditiva do bloco de vídeo recebido em relação a um ou mais blocos em um ou mais quadros de referência para fornecer a predição temporal. A unidade de intra-predição 46 pode realizar alternativamente a codificação intra-preditiva do bloco de vídeo recebido com relação a um ou mais blocos vizinhos no mesmo quadro ou fatia como o bloco a ser codificado para fornecer predição espacial. O codificador de vídeo 20 pode realizar múltiplas passagens de codificação, por exemplo, para selecionar um modo de codificação apropriado para cada bloco de dados de vídeo.

[0144] Além disso, a unidade de particionamento 48 pode particionar blocos de dados de vídeo em sub-blocos, com base na avaliação de esquemas de particionamento anteriores em passagens de codificação anteriores. Por exemplo, a unidade de particionamento 48 pode inicialmente particionar um quadro ou uma fatia em LCUs, e particionar cada uma das LCUs em sub-CUs com base na análise da taxa-distorção (por exemplo, otimização da taxa-distorção). A unidade de seleção de modo 40 pode ainda produzir uma estrutura de dados quadtree indicativa do particionamento de uma LCU em sub-CUs. As CUs de nó-folha da quadtree podem incluir uma ou mais PUs ou uma ou mais TUs.

[0146] 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 fornecer o bloco intra ou inter codificado resultante ao somador 50 para gerar dados de blocos residuais e ao 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 intra modo, informações de particionamento e

outras informações de sintaxe, para a unidade de codificação por entropia 56.

[0146] A unidade de estimação de movimento 42 e a unidade de compensação de movimento 44 podem ser altamente integradas, mas são ilustradas separadamente para fins conceituais. A estimação de movimento, realizada pela unidade de estimação de movimento 42, é o processo para gerar vetores de movimento, que estimam movimento para os blocos de vídeo. Um vetor de movimento, por exemplo, pode indicar o deslocamento de um PU de um bloco de vídeo dentro de um quadro de vídeo ou imagem atual 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 atual (ou outra unidade codificada). Um bloco preditivo é um bloco que mais se aproxima ao bloco a ser codificado, em termos de diferença de pixels, o que pode ser determinado pela soma da diferença absoluta (SAD), pela soma diferença quadrática (SSD) ou outras medidas de diferença. Em alguns exemplos, o codificador de vídeo 20 pode calcular os valores para as posições de pixel sub-inteiros de imagens de referência armazenadas na memória de imagem de referência 64. Por exemplo, o codificador de vídeo 20 pode interpolar os valores de posições de pixel de um quarto, posições de pixel de um oitavo, ou outra posição de pixel fracionário da imagem de referência. Portanto, a unidade de estimação de movimento 42 pode realizar uma busca de movimento em relação às posições de pixel total e posições de pixel fracionário e liberar um vetor de movimento com precisão de pixel fracionário.

[0147] A unidade de estimação de movimento 42 calcula um vetor de movimento para uma PU de um bloco de vídeo em uma fatia inter-codificada, comparando-se 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 imagens de referência (lista 0) ou de uma segunda lista de imagens de referência (lista 1), cada uma das quais identifica uma ou mais imagens de referência armazenadas na memória de imagem de referência 64. A unidade de estimação de movimento 42 envia o vetor de movimento calculado para a unidade de codificação por entropia 56 e para a unidade de compensação de movimento 44.

[0148] A compensação de movimento, realizada pela unidade de compensação de movimento 44, pode envolver a busca ou a geração do bloco preditivo com base no vetor de movimento determinado pela unidade de estimação de movimento 42. Mais uma vez, a unidade de estimação 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 atual, 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 imagens de referência. O somador 50 forma um bloco vídeo residual, subtraindo os valores de pixel do bloco preditivo dos valores de pixel do bloco de vídeo atual a ser codificado, formando os valores de diferença de pixel, como discutido abaixo. Em geral, a unidade de estimação de movimento 42 realiza a estimação de movimento em relação às componentes de luminância, e a unidade de compensação de movimento 44 utiliza os vetores de movimento calculados com base nas componentes de luminância para ambas as componentes de crominância e componentes de luminância. A unidade de seleção de modo 40 também pode gerar elementos de sintaxe associados com os blocos de vídeo e com a fatia de vídeo

para serem utilizados pelo decodificador de vídeo 30 na decodificação dos blocos de vídeo da fatia de vídeo.

[0149] A unidade de intra-predição 46 pode intra-predizer um bloco atual, como uma alternativa à inter-predição realizada pela unidade de estimação de movimento 42 e pela unidade de compensação de movimento 44, como descrito acima. Em particular, a unidade de intra-predição 46 pode determinar um modo de intra-predição para utilizar para codificar um bloco atual. Em alguns exemplos, a unidade de intra-predição 46 pode codificar um bloco atual utilizando vários modos de intra-predição, por exemplo, durante as passagens de codificação separadas, e unidade de intra-predição 46 (ou a unidade de seleção de modo 40, em alguns exemplos) pode selecionar um modo de intra-predição apropriado para utilizar a partir dos modos testados.

[0150] Por exemplo, a unidade de intra-predição 46 pode calcular valores de taxa-distorção, utilizando uma análise da taxa-distorção para os vários modos de intra-predição testados, e selecionar o modo de intra-predição possuindo as melhores características de taxa-distorção entre os modos testados. A análise da taxa-distorção geralmente determina uma quantidade de distorção (ou erro) entre um bloco codificado e um bloco original, um bloco não codificado que foi codificado para produzir o bloco codificado, bem como uma taxa de bits (ou seja, um número de bits) utilizada para produzir o bloco codificado. A unidade de intra-predição 46 pode calcular taxas a partir das distorções e taxas para os vários blocos codificados para determinar qual o modo de intra-predição exibe o melhor valor de taxa-distorção para o bloco.

[0151] Depois de selecionar um modo de intra-predição para um bloco, a unidade de intra-predição 46 pode fornecer a informação indicativa do modo de intra-predição

selecionado para o bloco à unidade de codificação por entropia 56. A unidade de codificação por entropia 56 pode codificar a informação indicando o modo intra-predição selecionado. O codificador de vídeo 20 pode incluir no fluxo de bits transmitido os dados de configuração que podem compreender uma pluralidade de tabelas de índices de modo de intra-predição e uma pluralidade de tabelas de índices de modo intra-predição modificado (também referidas como tabelas de mapeamento de palavra-código), as definições de contextos de codificação para vários blocos, e as indicações de um modo de intra-predição mais provável, uma tabela de índices de modo intra-predição e uma tabela de índices de modo intra-predição modificado para utilizar em cada um dos contextos.

[0152] O codificador de vídeo 20 forma um bloco de vídeo residual, subtraindo-se os dados de predição da unidade de seleção de modo 40 do bloco de vídeo original que está sendo codificado. O somador 50 representa o componente ou os componentes que realizam esta operação de subtração. A unidade de processamento da transformada 52 aplica uma transformada, tal como uma transformada discreta de cosseno (DCT) ou uma transformada conceitualmente similar, para o bloco residual, produzindo um bloco de vídeo composto por valores do coeficiente de transformada residual. A unidade de processamento da transformada 52 pode realizar outras transformadas que são conceitualmente semelhantes à DCT. Transformada Wavelet, transformada de números inteiros, transformada de sub-banda ou outros tipos de transformadas também podem ser usadas. Em qualquer caso, a unidade de processamento de transformada 52 aplica a transformada ao bloco residual, produzindo um bloco de coeficientes de transformada residual. A transformada pode converter as informações residuais de um domínio de valor

de pixel em um domínio de transformada, tal como um domínio da frequência. A unidade de processamento de transformada 52 pode enviar os coeficientes de transformada resultantes da unidade de quantização 54. A unidade de quantização 54 quantiza os coeficientes de transformada a fim de reduzir ainda mais a taxa de bits. O processo de quantização pode reduzir a profundidade de bits associada com alguns ou com todos os coeficientes. O grau de quantização pode ser modificado ajustando-se um parâmetro de quantização. Em alguns exemplos, a unidade quantização 54 pode, em seguida, realizar uma varredura da matriz incluindo os coeficientes de transformada quantizados. Alternativamente, a unidade de codificação por entropia 56 pode realizar a varredura.

[0153] Após a quantização, a unidade de codificação por entropia 56 codifica por entropia os coeficientes de transformada quantizados. Por exemplo, a unidade de codificação por entropia 56 pode implementar a codificação de comprimento variável adaptativa ao contexto (CAVLC), a codificação da aritmética binária adaptativa ao contexto (CABAC), a codificação da aritmética binária adaptativa ao contexto com base em sintaxe (SBAC), a codificação por entropia do intervalo de probabilidade de particionamento (PIPE) ou outra técnica de codificação. No caso da codificação por entropia baseada em contexto, o contexto pode se basear nos 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, decodificador de vídeo 30) ou arquivado para posterior recuperação ou transmissão.

[0154] De acordo com as técnicas desta descrição, a unidade de codificação por entropia 56 do codificador de vídeo 20 pode realizar uma ou mais técnicas da presente

descrição. Por exemplo, a unidade de codificação por entropia 56 do codificador de vídeo 20 pode determinar se utiliza a conversão de espaço-cor para uma unidade de codificação. O modo de codificação pode ser um de um modo de codificação com perdas ou um modo de codificação sem perdas. A unidade de codificação por entropia 56 do codificador de vídeo 20 pode ainda determinar um processo de transformada de espaço-cor dependente do modo de codificação utilizado para codificar os dados de vídeo. A unidade de codificação por entropia 56 do codificador de vídeo 20 pode adicionalmente aplicar o processo de transformada de espaço-cor para os dados de vídeo. A unidade de codificação por entropia 56 do codificador de vídeo 20 pode adicionalmente aplicar um processo de transformada inversa de espaço-cor em um loop de decodificação do processo de codificação. Nas técnicas da presente descrição, o processo de transformada inversa de espaço-cor é independente do fato de o modo de codificação ser o modo de codificação com perdas ou o modo de codificação sem perdas.

[0155] A unidade de quantização inversa 58 e a unidade de transformada inversa 60 aplicam a quantização inversa e a transformada inversa, respectivamente, para reconstruir o bloco residual no domínio de pixel, por exemplo, para utilização posterior como um bloco de referência. A unidade de compensação de movimento 44 pode calcular um bloco de referência, adicionando-se 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 também pode aplicar um ou mais filtros de interpolação ao bloco residual reconstruído para calcular valores de pixel sub-inteiro para utilizar na estimação de movimento. O somador 62 adiciona o bloco residual

reconstruído ao bloco de predição de movimento compensado 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 utilizado pela unidade de estimação de movimento 42 e pela unidade de compensação de movimento 44 como um bloco de referência para inter-codificar um bloco em um quadro de vídeo subsequente.

[0156] Dessa maneira, a unidade de codificação por entropia 56 do codificador de vídeo 20 pode realizar uma ou mais técnicas desta descrição. Por exemplo, a unidade de codificação por entropia 56 do codificador de vídeo 20 pode executar uma ou mais técnicas da descrição atual. Por exemplo, a unidade de codificação por entropia 56 do codificador de vídeo 20 pode determinar um modo de codificação utilizado para codificar os dados de vídeo. O modo de codificação pode ser um de um modo de codificação com perdas ou um modo de codificação sem perdas. A unidade de codificação por entropia 56 pode adicionalmente determinar um processo de transformada de espaço-cor dependente do modo de codificação utilizado para codificar os dados de vídeo. A unidade de codificação por entropia 56 do codificador de vídeo 20 pode adicionalmente aplicar o espaço-cor- transformar processo para os dados de vídeo. Entropia codifica unidade 56 do codificador de vídeo 20 pode adicionalmente aplicar um processo de transformada inversa de espaço-cor em um loop de decodificação do processo de codificação. Nas técnicas da presente descrição, o processo de transformada inversa de espaço-cor é independente do fato de o modo de codificação ser o modo de codificação com perdas ou o modo de codificação sem perdas.

[0157] Em alguns exemplos, o modo de codificação pode ser o modo de codificação com perdas. Em tais exemplos, o processo de transformada de espaço-cor pode ser uma matriz YCoCg. A unidade de codificação por entropia 56 do codificador de vídeo 20 pode modificar uma componente Co e uma componente CG da matriz YCoCg com um desvio à esquerda de N bits antes de aplicar o processo de transformada inversa de espaço-cor no circuito de decodificação do processo de codificação. Em alguns exemplos, N pode ser igual a 1.

[0158] Em outros exemplos, o modo de codificação pode ser o modo de codificação sem perdas. Em tais exemplos, o processo de transformada de espaço-cor pode ser uma matriz YCoCg-R. A unidade de codificação por entropia 56 do codificador de vídeo 20 pode aumentar uma profundidade de bit de uma componente Co e uma componente Cg da matriz YCoCg-R por um.

[0159] Em qualquer um dos dois exemplos acima, o processo de transformada inversa de espaço-cor pode ser um processo de transformada inversa de espaço-cor sem perdas. Nestes exemplos, o processo de transformada inversa de espaço-cor sem perdas pode compreender uma matriz YCoCg-R.

[0160] Em alguns exemplos, a unidade de codificação por entropia 56 do codificador de vídeo 20 pode adicionalmente determinar utilizar a conversão de espaço-cor para codificar os dados de vídeo. Ao fazê-lo, a unidade de codificação por 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 de espaço-cor em resposta à determinação de utilizar a conversão de espaço-cor para a unidade de codificação. Em alguns exemplos, o elemento de sintaxe compreende um indicador 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 de código intra-pulso (IPCM). Em outros exemplos, o elemento de sintaxe é sinalizado apenas quando existem 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 utilizando a conversão de espaço-cor.

[0161] 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 intra-codificada 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.

[0162] Em alguns exemplos, o elemento de sintaxe é um primeiro elemento de sintaxe e os dados de vídeo é um primeiro conjunto de dados de vídeo. Em tais exemplos, a unidade de codificação por entropia 56 do codificador de vídeo 20 pode determinar um valor de um índice de paleta para um primeiro pixel de uma segunda unidade de codificação de um segundo conjunto de dados de vídeo. A unidade de codificação por entropia 56 de codificador de vídeo 20 pode adicionalmente determinar os valores de índices de paleta para um ou mais pixels seguintes em uma ordem de varredura imediatamente a seguir ao primeiro pixel. Determinar os valores dos índices de paleta pode compreender determinar os valores dos índices de paleta até um pixel ter um índice de paleta com um valor diferente do valor do índice de paleta para o primeiro pixel. A unidade de codificação por entropia 56 do codificador de vídeo 20

pode adicionalmente determinar uma série de valores de índice de paleta definidos para as próximas pixels. Com base no número de valores de índice de paleta, a unidade de codificação por entropia 56 do codificador de vídeo 20 pode codificar o índice de paleta utilizando um primeiro modo ou um segundo modo. Quando o número de valores de índice de paleta é maior que um, a unidade de codificação por entropia 56 do codificador de vídeo 20 pode definir um valor de um segundo elemento de sintaxe para indicar que o primeiro modo foi utilizado para a codificação de índice de paleta e definir um valor de um terceiro elemento de sintaxe igual ao número de valores de índice de paleta. Quando o número de valores de índice de paleta é menor ou igual a um, a unidade de codificação por entropia 56 do codificador de vídeo 20 pode definir o valor do segundo elemento de sintaxe para indicar que o segundo modo foi utilizado para codificação do índice de paleta. Em alguns exemplos, o primeiro modo é um modo de execução, e o segundo modo é um modo de pixel.

[0163] Em alguns exemplos, o elemento de sintaxe de codificação pode compreender a unidade de codificação por entropia 56 do codificador de vídeo 20 para codificar dados de uma mensagem de informação aprimorada suplementar (SEI) que indica se a unidade de codificação foi codificada utilizando a conversão de espaço-cor.

[0164] A Fig. 3 é um diagrama de blocos que ilustra um exemplo de decodificador de vídeo 30 que pode implementar técnicas para decodificação de blocos de vídeo, alguns dos quais foram codificados utilizando um processo de conversão de espaço-cor. No exemplo da Fig. 3, o decodificador de vídeo 30 compreende uma unidade de decodificação 70 por entropia, uma unidade de compensação de movimento 72, uma unidade de intra-predição 74, uma

unidade de quantização inversa 76, uma unidade de transformada inversa 78, uma memória de imagem de referência 82 e um somador 80. O decodificador de vídeo 30 pode, em alguns exemplos, realizar uma passagem de decodificação geralmente recíproca à passagem de codificação descrita com relação ao codificador de vídeo 20 (Fig. 2). A unidade de compensação de Movimento 72 pode gerar dados de predição com base em vetores de movimento recebidos da unidade de decodificação por entropia 70, enquanto a unidade de intra-predição 74 pode gerar dados de predição com base em indicadores de modo de intra-predição recebidos da unidade de decodificação por entropia 70.

[0165] 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 do codificador de vídeo 20. A unidade de decodificação por entropia 70 do decodificador de vídeo 30 decodifica por entropia o fluxo de bits para gerar coeficientes quantizados, vetores de movimento ou indicadores de modo de intra-predição e outros elementos de sintaxe. A unidade de decodificação por entropia 70 encaminha os vetores de movimento e os outros elementos de sintaxe para unidade de compensação de movimento 72. O decodificador de vídeo 30 pode receber os elementos de sintaxe em nível de fatia de vídeo e/ou em nível de bloco de vídeo.

[0166] De acordo com as técnicas desta descrição, a unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode realizar uma ou mais técnicas da presente descrição. Por exemplo, a unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode receber um primeiro bloco codificado de dados de vídeo. O primeiro bloco codificado de dados de vídeo foi codificado usando um

modo de codificação com perdas e um primeiro processo de transformada de espaço-cor. A unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode receber ainda um segundo bloco codificado de dados de vídeo. O segundo bloco codificado de dados de vídeo foi codificado usando um modo de codificação sem perdas e um segundo processo de transformada de espaço-cor. A unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode promover o processo de transformada inversa de espaço-cor para o primeiro bloco codificado de dados de vídeo. A unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode adicionalmente aplicar o mesmo processo de transformada inversa de espaço-cor para o segundo bloco codificado de dados de vídeo. No lado do decodificador, pode não haver necessidade de realizar um processo de transformada de espaço-cor direta, independentemente do modo de codificação. Em alguns exemplos, o processo de transformada inversa de espaço-cor pode ser fixado.

[0167] Quando a fatia de vídeo é codificada como uma fatia intra-codificada (I), a unidade de intra-predição 74 pode gerar predição dados para um bloco de vídeo da fatia de vídeo atual com base em um modo de intra-predição sinalizado e dados de blocos anteriormente decodificados dos quadros ou imagens atuais. Quando o quadro de vídeo é codificado como uma fatia inter-codificada (i.e., B, P ou GPB), a unidade de compensação de movimento 72 produz blocos preditivos para um bloco de vídeo da fatia de vídeo atual com base em vetores de movimento e outros elementos de sintaxe recebidos da unidade de decodificação por entropia 70. Os blocos preditivos podem ser produzidos a partir uma dentre as imagens de referência dentro de uma das listas de imagens de referência. O decodificador de vídeo 30 pode construir as listas do quadro de referência,

lista 0 e lista 1, utilizando as 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 da fatia de vídeo atual analisando 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 atual que está sendo decodificado. Por exemplo, a unidade de compensação de movimento 72 usa alguns elementos de sintaxe recebidos para determinar um modo de predição (por exemplo, intra ou inter predição) utilizado para codificar os blocos de vídeo da fatia de vídeo, um tipo de fatia de inter-predição (por exemplo, fatia B, fatia P ou fatia GPB), informações de construção para uma ou mais dentre as listas de imagem de referência para a fatia, vetores de movimento para cada bloco de vídeo inter-codificado da fatia, estado de inter-predição para cada bloco vídeo inter-codificado da fatia e outras informações para decodificar os blocos de vídeo na fatia de vídeo atual.

[0168] A unidade de compensação de movimento 72 também pode realizar interpolação com base em filtros de interpolação. A unidade de compensação de movimento 72 pode utilizar filtros de interpolação como os usados pelo codificador de vídeo 20 durante a codificação dos blocos de vídeo para calcular valores interpolados para pixels sub-inteiro 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 dos elementos de sintaxe recebidos e utilizar os filtros de interpolação para produzir blocos preditivos.

[0169] A unidade de quantização inversa 76 inverte a quantização, ou seja, dequantiza, os coeficientes

de transformadas quantizados fornecidos no fluxo de bits e decodificados pela unidade de decodificação por entropia 70. O processo de quantização inversa pode compreender a utilização de um parâmetro de quantização QPy calculado pelo decodificador de vídeo 30 para cada bloco de vídeo na fatia de vídeo para determinar um grau de quantização e, da mesma forma, o grau de quantização inversa que deve ser aplicado.

[0170] A unidade de transformada inversa 78 aplica uma transformada inversa, por exemplo, uma DCT inversa, uma transformada inteira inversa ou um processo conceitualmente similar de transformada inversa, aos coeficientes de transformada para produzir blocos residuais no domínio do pixel.

[0171] Após a unidade de compensação de movimento 72 gerar o bloco preditivo para o bloco de vídeo atual com base nos vetores de movimento e outros elementos de sintaxe, o decodificador de vídeo 30 forma um bloco de vídeo decodificado ao somar os 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 os componentes que realizam esta operação de soma. Se desejado, um filtro de desagrupamento também pode ser aplicado para filtrar os blocos decodificados para remover artefatos de blocagem. Outros filtros de loop (no loop de codificação ou após o loop de codificação) também podem ser usados para suavizar as transições de pixels, ou de outra forma melhorar a qualidade de vídeo. Os blocos de vídeo decodificados em um determinado quadro ou imagem são então armazenados na memória de imagem de referência 82, que armazena imagens de referência utilizadas para compensação do movimento subsequente. A memória de imagem de referência

82 também armazena o vídeo decodificado para posterior apresentação em um dispositivo de exibição, como dispositivo de exibição 32 da FIG. 1.

[0172] Dessa maneira, a unidade de codificação por entropia 70 do decodificador de vídeo 30 pode realizar uma ou mais técnicas de presente descrição. Por exemplo, a unidade de decodificação de entropia 70 do decodificador de vídeo 30 pode receber um primeiro bloco codificado de dados de vídeo. O primeiro bloco codificado de dados de vídeo foi codificado usando um modo de codificação com perdas e um primeiro processo de transformada de espaço-cor. A unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode receber mais um segundo bloco codificado de dados de vídeo. O segundo bloco codificado de dados de vídeo foi codificado usando um modo de codificação sem perdas e um segundo processo de transformada de espaço-cor. A unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode promover o processo de transformada inversa de espaço-cor para o primeiro bloco codificado de dados de vídeo. A unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode adicionalmente aplicar o mesmo processo de transformada inversa de espaço-cor para o segundo bloco codificado de dados de vídeo. No lado do decodificador, pode não haver necessidade de realizar um processo de transformada direta de espaço-cor, independentemente do modo de codificação. Em alguns exemplos, o processo de transformada inversa de espaço-cor pode ser fixado. Em alguns exemplos, os blocos de dados de vídeo podem ser unidades de codificação.

[0173] Em alguns exemplos, a unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode modificar adicionalmente uma ou mais componentes do primeiro bloco codificado de dados de vídeo com um desvio à

direita de N bits ou um desvio à esquerda de N bits. Em alguns exemplos, N pode ser igual a 1. Em alguns exemplos, uma ou mais componentes do primeiro bloco codificado de dados de vídeo podem ser duas componentes de crominância.

[0174] Em alguns exemplos, o primeiro processo de transformada de espaço-cor compreende uma matriz YCoCg. Em outros exemplos, o segundo processo de transformada de espaço-cor compreende uma matriz YCoCg-R. Em qualquer exemplo, o processo de transformada inversa de espaço-cor pode compreender um processo de transformada inversa de espaço-cor sem perdas. Em alguns exemplos, o processo de transformada inversa de espaço-cor sem perdas compreende uma matriz YCoCg-R.

[0175] Em alguns exemplos, a unidade de decodificação por 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-cor. Em alguns exemplos, o elemento de sintaxe pode compreender um indicador de um bit. Em alguns exemplos, a unidade de codificação é codificada em um modo diferente do modo de modulação de código intra-pulso (IPCM) e o elemento de sintaxe é sinalizado apenas para unidades de codificação utilizando um modo diferente do modo IPCM. Em alguns exemplos, o elemento de sintaxe indica que a unidade de codificação foi codificada utilizando a conversão de espaço-cor, quando existem 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 utilizando a conversão de espaço-cor. A unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode adicionalmente determinar se um valor do elemento

de sintaxe indica que a unidade de codificação foi codificada usando a conversão de espaço-cor. Em resposta à determinação de que o elemento de sintaxe indica que a unidade de codificação foi codificada usando conversão de espaço-cor, a unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode aplicar o processo de transformada inversa de espaço-cor.

[0176] Em alguns exemplos, o elemento de sintaxe pode indicar que a conversão de espaço-cor não foi usada para codificar a unidade de codificação. Por exemplo, o elemento de sintaxe pode indicar que a conversão de espaço-cor não foi usada para codificar a unidade de codificação, quando a unidade de codificação é intra codificada e quando um modo de predição de luminância e um modo de predição de crominância de uma unidade de predição no interior da unidade de codificação são diferentes. Em outro exemplo, o elemento de sintaxe pode indicar que a conversão de espaço-cor não foi usada para codificar a unidade de codificação, quando a unidade de codificação está 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.

[0177] Em alguns exemplos, o elemento de sintaxe é um primeiro elemento de sintaxe e os dados de vídeo são um primeiro conjunto de dados de vídeo. Em tais exemplos, a unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode adicionalmente determinar um valor de um segundo elemento de sintaxe de um segundo conjunto de dados de vídeo. O segundo elemento de sintaxe pode indicar se um modo de paleta foi usado para codificar o segundo conjunto de dados de vídeo. Em resposta ao segundo elemento de sintaxe indicando que o modo de paleta foi usado para

codificar os dados de vídeo, uma unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode determinar um valor de um terceiro elemento de sintaxe. O terceiro elemento de sintaxe pode indicar se um primeiro modo ou um segundo modo é usado para decodificar um índice de paleta para um pixel na unidade de codificação. Com base no valor determinado do segundo elemento de sintaxe, a unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode decodificar o índice de paleta usando o primeiro modo ou o segundo modo. Quando se utiliza o primeiro modo, a unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode determinar um valor do índice de paleta, determinar um valor de um quarto elemento de sintaxe que indica um número de pixels de uma ordem de varredura imediatamente posterior ao pixel sendo decodificado, e duplicar o resultado da determinação do valor do índice de paleta para os próximos N pixels na ordem de varredura, com N igual ao valor do quarto elemento de sintaxe. Em alguns exemplos, o primeiro modo é um modo de execução. Quando se utiliza o segundo modo, a unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode determinar o valor do índice de paleta de saída e um valor de amostra de pixel para pixel, em que o valor da amostra de pixel é igual ao valor do índice de paleta. Em alguns exemplos, o segundo modo é um modo de pixel.

[0178] Em alguns exemplos, a decodificação do elemento de sintaxe compreende uma unidade de decodificação por entropia 70 do decodificador de vídeo 30, sendo configurada para decodificar os dados de uma mensagem de informação aprimorada suplementar (SEI) que indica se a unidade de codificação foi codificada usando a conversão de espaço-cor.

[0179] A Fig. 4 é um diagrama conceitual ilustrando os 35 modos de predição HEVC de acordo com uma ou mais técnicas da presente descrição. No HEVC atual, para a componente de luminância de cada Unidade de Predição (PU), um método de intra-predição é utilizado com 33 modos de predição angular (indexados a partir de 2 até 34), um modo DC (indexado com 1) e um modo Planar (indexado com 0), conforme descrito em relação à Fig. 4.

[0180] Além dos 35 intra-modos acima citados, um outro modo, chamado 'I-PCM', também é empregado pelo HEVC. No modo I-PCM, predição, transformada, quantização e codificação por 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 de quando o sinal não pode ser eficientemente codificado por outros modos.

[0181] A Fig. 5 é um diagrama conceitual ilustrando candidatos espaciais vizinhos de vetor de movimento para os modos de fusão e avançado de predição de vetor de movimento (AMVP) de acordo com uma ou mais técnicas da presente descrição. Conforme descrito em relação a Fig. 5, candidatos espaciais MV são derivados dos blocos vizinhos mostrados na Fig. 5 para uma PU específica (PU0), embora os métodos que geram os candidatos a partir dos blocos sejam diferentes para os modos de fusão e para os modos AMVPpara os modos de predição avançado de vector de movimento (AMVP) de acordo com uma ou mais das técnicas da presente descrição atual e intercalação. Conforme descrito em relação à FIG. 5, os candidatos MV espaciais são derivados a partir dos blocos vizinhos, mostrados na FIG. 5 para uma PU específica (PU0), embora os métodos geram os candidatos dos blocos diferentes para os modos de fusão e AMVP.

[0182] No modo de fusão, até quatro candidatos espaciais MV podem ser derivados segundo as ordens mostradas na Fig. 5(a) com números, e a ordem é a seguinte: esquerda (0), acima (1), acima à direita (2), abaixo à esquerda (3) e acima à esquerda (4), como mostrado na Fig. 5(a).

[0183] No modo AMVP, os blocos vizinhos são divididos em dois grupos: o grupo esquerdo consistindo dos blocos 0 e 1 e o grupo acima consistindo dos blocos 2, 3 e 4, conforme mostrado na Fig. 5(b). Para cada grupo, o candidato em potencial em um bloco vizinho referindo-se à mesma imagem de referência como o indicado pelo índice de referência sinalizado, tem a prioridade mais alta para ser escolhida para formar um candidato final do grupo. É possível que todos os blocos vizinhos não contenham um vetor de movimento apontando para a mesma imagem de referência. Portanto, se esse candidato não for encontrado, o primeiro candidato disponível será escalado para formar o candidato final, assim as diferenças de distância temporal podem ser compensadas.

[0184] A Fig. 6 é um diagrama conceitual ilustrando um exemplo de cópia intra bloco (BC) de acordo com uma ou mais técnicas da presente descrição. Conforme descrito em relação à Fig. 6, a cópia intra bloco (BC) foi incluída na RExt. Um exemplo de Intra BC é mostrado na Fig. 6, onde a CU atual é predita a partir de um bloco já decodificado da imaginem/fatia atual. O tamanho do intra bloco BC atual pode ser tão grande quanto o tamanho de uma CU, que varia de 8x8 a 64x64, embora em algumas aplicações restrições adicionais podem ser ainda aplicadas.

[0185] A Fig. 7 é um diagrama conceitual ilustrando um exemplo de uma amostra de bloco alvo e uma amostra de referência para um intra bloco 8x8, de acordo

com uma ou mais técnicas da presente descrição. Conforme descrito abaixo com relação à Fig. 7, uma matriz de transformada é derivada de valores da amostra de referência. Amostras de referência diferentes são utilizadas para o caso intra e o caso inter. Para o caso de intra bloco, o bloco alvo e a amostra de referência são mostrados na Fig. 7. Nesta figura, o bloco alvo compreende amostras hachurada com linhas cruzadas 8x8 e as amostras de referências são listradas e pontilhadas.

[0186] Para o caso de inter blocos, as amostras de referência para a determinação da matriz são as mesmas que as para compensação de movimento. Para poder realizar a operação de desvio, as amostras de referência no bloco AMP são sub-amostradas de tal forma que o número de amostras torna-se potência de dois. Por exemplo, o número de amostras de referência em um bloco 12x16 é reduzido para 2/3.

[0187] De acordo com algumas técnicas da presente descrição, o processo de transformada de espaço-cor pode ser aplicado. Em tais exemplos, não há nenhuma necessidade de sinalizar se o processo de transformada é invocado ou não. Além disso, ambos os lados do codificador e do decodificador podem derivar a matriz de transformada com o mesmo método para evitar o overhead na sinalização da matriz de transformada.

[0188] A Fig. 8 é um fluxograma que ilustra um exemplo de método para codificação de um bloco atual. O bloco atual pode compreender uma CU atual ou uma parte da CU atual. Embora descrito em relação ao codificador de vídeo 20 (Figs. 1 e 2), deve ser entendido que outros dispositivos podem ser configurados para realizar um método semelhante ao da Fig. 8.

[0189] Neste exemplo, o codificador de vídeo 20 inicialmente predita o bloco atual (150). Por exemplo, o codificador de vídeo 20 pode calcular uma ou mais unidades de predição (PUs) para o bloco atual. O codificador de vídeo 20 pode então calcular um bloco residual para o bloco atual, por exemplo, para produzir uma unidade de transformada (TU) (152). Para calcular o bloco residual, o codificador de vídeo 20 pode calcular a diferença entre o bloco original, não codificado, e o bloco predito para o bloco atual. A unidade de codificação por entropia 56 do codificador de vídeo 20 pode determinar um modo de codificação usado para codificar dados de vídeo (154). O modo de codificação pode ser um de um modo de codificação com perdas ou um modo de codificação sem perdas. A unidade de codificação por entropia 56 pode adicionalmente determinar um processo de transformada de espaço-cor dependente do modo de codificação utilizado para codificar os dados de vídeo (156). A unidade de codificação por entropia 56 do codificador de vídeo 20 pode adicionalmente aplicar o processo de transformada de espaço-cor para os dados de vídeo (158). O codificador de vídeo 20 pode, em seguida, transformar e quantizar coeficientes do bloco residual (160). Em seguida, o codificador de vídeo 20 pode varrer coeficientes de transformada quantizados do bloco residual (162). Durante a varredura, ou após a varredura, 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). A unidade de codificação por entropia 56 do codificador de vídeo 20 pode adicionalmente aplicar um processo de transformada inversa de espaço-cor em um loop de decodificação do processo de

codificação (168). Nas técnicas da presente descrição, o processo de transformada inversa de espaço-cor é independente do fato de o modo de codificação ser o modo de codificação com perdas ou o modo de codificação sem perdas.

[0190] Em alguns exemplos, o modo de codificação pode ser o modo de codificação com perdas. Em tais exemplos, o processo de transformada de espaço-cor pode ser uma matriz YCoCg. A unidade de codificação por entropia 56 do codificador de vídeo 20 pode modificar uma componente Co e uma componente de CG da matriz YCoCg com um desvio à esquerda de N bits antes de aplicar o processo de transformada inversa de espaço-cor no loop de decodificação do processo de codificação. Em alguns exemplos, N pode ser igual a 1.

[0191] Em outros exemplos, o modo de codificação pode ser o modo de codificação sem perdas. Em tais exemplos, o processo de transformada de espaço-cor pode ser uma matriz YCoCg-R. A unidade de codificação por entropia 56 do codificador de vídeo 20 pode aumentar a uma profundidade de uma componente Co e uma componente Cg da matriz YCoCg-R por um bit.

[0192] Em qualquer um dos dois exemplos acima, o processo de transformada inversa de espaço-cor pode ser um processo de transformada inversa de espaço-cor sem perdas. Nestes exemplos, o processo de transformada inversa de espaço-cor sem perdas pode compreender uma matriz YCoCg-R.

[0193] Em alguns exemplos, a unidade de codificação por entropia 56 do codificador de vídeo 20 pode adicionalmente determinar a utilizar a conversão de espaço-cor para codificar os dados de vídeo. Ao fazê-lo, a unidade de codificação por entropia 56 do codificador de vídeo 20 pode definir um valor de um elemento de sintaxe dos dados de vídeo para indicar o uso da conversão de espaço-cor em

resposta à determinação de utilizar a conversão de espaço-cor para a unidade de codificação. Em alguns exemplos, o elemento de sintaxe compreende um indicador 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 de código intra-pulso (IPCM). Em outros exemplos, o elemento de sintaxe é sinalizado apenas 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-cor.

[0194] 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 intra-codificada 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.

[0195] Em alguns exemplos, o elemento de sintaxe é um primeiro elemento de sintaxe e os dados de vídeo são um primeiro conjunto de dados de vídeo. Em tais exemplos, a unidade de codificação por entropia 56 do codificador de vídeo 20 pode determinar um valor de um índice de paleta para um primeiro pixel de uma segunda unidade de codificação de um segundo conjunto de dados de vídeo. A unidade de codificação por entropia 56 do codificador de

vídeo 20 pode adicionalmente determinar os valores de índices de paleta para um ou mais próximos pixels em uma ordem de varredura imediatamente a seguir do primeiro pixel. Determinar os valores dos índices de paleta pode compreender determinar os valores dos índices de paleta até um pixel ter um índice de paleta com um valor diferente do valor do índice de paleta para o primeiro pixel. A unidade de codificação por entropia 56 do codificador de vídeo 20 pode adicionalmente determinar uma série de valores de índice de paleta definidos para os próximos pixels. Com base no número de valores de índice de paleta, a unidade de codificação por entropia 56 do codificador de vídeo 20 pode codificar o índice de paleta usando um primeiro modo ou um segundo modo. Quando o número de valores de índice paleta é maior do que um, a unidade de codificação por entropia 56 do codificador de vídeo 20 pode definir um valor de um segundo elemento de sintaxe para indicar que o primeiro modo foi utilizado para codificar o índice da paleta e definir um valor de um terceiro elemento de sintaxe igual ao número de valores de índice de paleta. Quando o número de valores de índice de paleta é menor do que ou igual a um, a unidade de codificação por entropia 56 do codificador de vídeo 20 pode definir o valor do segundo elemento de sintaxe para indicar que o segundo modo foi utilizado para codificar o índice de paleta. Em alguns exemplos, o primeiro modo é um modo de execução, e o segundo modo é um modo de pixel.

[0196] Em alguns exemplos, codificar o elemento de sintaxe pode compreender a unidade de codificação por entropia 56 do codificador de vídeo 20 codificar dados de uma mensagem de informação aprimorada suplementar (SEI) que indica se a unidade de codificação foi codificada usando a conversão de espaço-cor.

[0197] A Fig. 9 é um fluxograma que ilustra um exemplo de método para decodificar um bloco de dados de vídeo atual. O bloco atual pode compreender uma CU atual ou uma parte da CU atual. Embora descrito em relação ao decodificador de vídeo 30 (Figs. 1 e 3), deve ser entendido que outros dispositivos podem ser configurados para realizar um método semelhante ao da Fig. 9.

[0198] A unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode receber um primeiro bloco codificado de dados de vídeo (196). O primeiro bloco codificado de dados de vídeo foi codificado usando um modo de codificação com perdas e um primeiro processo de transformada de espaço-cor. A unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode receber ainda um segundo bloco codificado de dados de vídeo (198). O segundo bloco codificado de dados de vídeo foi codificado usando um modo de codificação sem perdas e um segundo processo de transformada de espaço-cor.

[0199] O decodificador de vídeo 30 pode predizer o bloco atual (200), por exemplo, utilizando um intra-modo ou uma inter-predição para calcular um bloco predito para o bloco atual. O codificador de vídeo 30 também pode receber dados codificados por entropia para o bloco atual, como os dados codificados por entropia para coeficientes de um bloco residual correspondente ao bloco atual (202). O decodificador de vídeo 30 pode decodificar por entropia os dados codificados por entropia para reproduzir os coeficientes do bloco residual (204). O decodificador de vídeo 30 pode, então, varrer inversamente os coeficientes reproduzidos (206), para criar um bloco de coeficientes de transformada quantizados. O decodificador de vídeo 30 pode então transformar inversamente e quantizar inversamente os coeficientes (208). A unidade de decodificação por entropia

70 do decodificador de vídeo 30 pode decodificar um elemento sintaxe para o bloco atual (210).

[0200] A unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode adicionalmente aplicar um processo de transformada inversa de espaço-cor para o primeiro bloco codificado de dados de vídeo (212). A unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode adicionalmente aplicar o mesmo processo de transformada inversa de espaço-cor para o segundo bloco codificado de dados de vídeo (214). O decodificador de vídeo 30 pode vir a decodificar o bloco atual, combinando o bloco predito e o bloco residual (216). No lado do decodificador, pode não haver necessidade de realizar um processo de transformada direta de espaço-cor, independentemente do modo de codificação. Em alguns exemplos, o processo de transformada inversa de espaço-cor pode ser fixado. Em alguns exemplos, os blocos de dados de vídeo podem ser unidades de codificação.

[0201] Em alguns exemplos, a unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode modificar adicionalmente uma ou mais componentes do primeiro bloco codificado de dados de vídeo com um desvio à direita de N bits ou um desvio à esquerda de N bits. Em alguns exemplos, N pode ser igual a 1. Em alguns exemplos, as uma ou mais componentes do primeiro bloco codificado de dados de vídeo podem ser duas componentes de crominância.

[0202] Em alguns exemplos, o primeiro processo de transformada de espaço-cor compreende a aplicação de uma matriz YCoCg. Em outros exemplos, o segundo processo de transformada de espaço-cor compreende a aplicação de uma matriz YCoCg-R. Em qualquer exemplo, o processo de transformada inversa de espaço-cor pode compreender um processo de transformada inversa de espaço-cor sem perdas.

Em alguns exemplos, o processo de transformada inversa de espaço-cor compreende a aplicação de uma matriz YCoCg-R.

[0203] Em alguns exemplos, a unidade de decodificação por 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-cor. Em alguns exemplos, o elemento de sintaxe pode compreender um indicador de um bit. Em alguns exemplos, a unidade de codificação é codificada em um modo diferente do modo de modulação de código intra-pulso (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 de espaço-cor 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-cor. A unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode adicionalmente determinar se um valor do elemento de sintaxe indica que a unidade de codificação foi codificada usando a conversão de espaço-cor. Em resposta à determinação de que o elemento de sintaxe indica que a unidade de codificação foi codificada usando conversão de espaço-cor, a unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode aplicar o processo de transformada inversa de espaço-cor.

[0204] Em alguns exemplos, o elemento de sintaxe pode indicar que a conversão de espaço-cor não foi usada para codificar a unidade de codificação. Por exemplo, o elemento de sintaxe pode indicar que a conversão de espaço-cor não foi usada para codificar a unidade de codificação,

quando a unidade de codificação é intra codificada e quando um modo de predição de luminância e um modo de predição de crominância de uma unidade de predição no interior da unidade de codificação são diferentes. Em outro exemplo, o elemento de sintaxe pode indicar que a conversão de espaço-cor não foi usada para codificar a unidade de codificação, quando a unidade de codificação está codificada com um modo de paleta. Nestes exemplos, o fluxo de bits recebido pode não incluir o elemento de sintaxe.

[0205] Em alguns exemplos, o elemento de sintaxe é um primeiro elemento de sintaxe e os dados de vídeo são um primeiro conjunto de dados de vídeo. Em tais exemplos, a unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode adicionalmente determinar um valor de um segundo elemento de sintaxe de um segundo conjunto de dados de vídeo. O segundo elemento de sintaxe pode indicar se um modo de paleta foi usado para codificar o segundo conjunto de dados de vídeo. Em resposta ao segundo elemento de sintaxe indicando que o modo de paleta foi usado para codificar os dados de vídeo, uma unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode determinar um valor de um terceiro elemento de sintaxe. O terceiro elemento de sintaxe pode indicar se um primeiro modo ou um segundo modo é usado para decodificar um índice de paleta para um pixel na unidade de codificação. Com base no valor determinado do segundo elemento de sintaxe, a unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode decodificar o índice de paleta usando o primeiro modo ou o segundo modo. Quando se utiliza o primeiro modo, a unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode determinar um valor do índice de paleta, determinar um valor de um quarto elemento de sintaxe que indica um número de pixels de uma ordem de

varredura imediatamente a seguir ao elemento de imagem sendo decodificado, e duplicar o resultado da determinação do valor do índice de paleta para os próximos N pixels na ordem de varredura, com N igual ao valor do quarto elemento de sintaxe. Em alguns exemplos, o primeiro modo é um modo de execução. Quando se utiliza o segundo modo, a unidade de decodificação por entropia 70 do decodificador de vídeo 30 pode determinar o valor do índice de paleta e emitir um valor de amostra de pixel para o pixel, em que o valor da amostra de pixel é igual ao valor do índice de paleta. Em alguns exemplos, o segundo modo é um modo de pixel.

[0206] Em alguns exemplos, a decodificação do elemento de sintaxe compreende uma unidade de decodificação por entropia 70 do decodificador de vídeo 30, sendo configurado para decodificar os dados de uma mensagem de informação aprimorada suplementar (SEI) que indica se a unidade de codificação foi codificada usando a conversão de espaço-cor.

[0207] É importante ser reconhecido que, dependendo do exemplo, certos atos ou eventos de qualquer uma das técnicas descritas neste documento 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, os atos ou os eventos podem ser realizados simultaneamente, por exemplo, através de vários segmentos de processamento multitarefa (multi-threaded), processamento de interrupção ou de 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 legível por computador e executadas por uma unidade de processamento baseada em hardware. O meio legível por computador pode compreender o meio de armazenamento legível por computador, que corresponde a um meio tangível tal como um meio de armazenamento de dados, ou um meio de comunicação, incluindo qualquer meio que facilita a transferência de um programa de computador de um lugar para outro, por exemplo, de acordo com um protocolo de comunicação. Desta forma, o meio legível por computador geralmente pode corresponder a (1) meio de armazenamento legível por computador tangível que é não transitório ou (2) um meio de comunicação tal como um sinal ou uma onda portadora. O meio de armazenamento de dados pode ser qualquer meio disponível que pode ser acessado por um ou mais computadores ou um ou mais processadores para recuperar instruções, códigos e/ou estruturas de dados para a implementação das técnicas descritas nesta descrição. Um produto de programa de computador pode incluir um meio legível por computador.

[0209] A título de exemplo e não de limitação, tal meio de armazenamento legível por computador pode incluir RAM, ROM, EEPROM, CD-ROM ou outro armazenamento em disco ótico, armazenamento em disco magnético, ou outros dispositivos de armazenamento magnéticos, memória flash, ou qualquer outro meio que pode ser utilizado para armazenar o código do programa desejado na forma de instruções ou estruturas de dados e que pode ser acessado por um computador. Ainda, qualquer conexão é corretamente denominada como um meio legível por computador. Por exemplo, se as instruções forem transmitidas por um site, servidor ou outra fonte remota utilizando um cabo coaxial, cabo de fibra óptica, par trançado, linha de assinante

digital (DSL) ou tecnologias sem fio como infravermelho, rádio e microondas, então o cabo coaxial, o cabo de fibra óptica, par trançado, DSL ou tecnologias sem fio como infravermelho, rádio e microondas estão incluídos na definição de meio. Deve ser entendido, no entanto, que meio de armazenamento legível por computador e meio de armazenamento de dados não compreendem conexões, ondas portadoras, sinais ou outros meios de comunicação transitórios, mas em vez disso são direcionados para os meios de armazenamento tangíveis não transitórios. Disco, como utilizado aqui, inclui disco compacto (CD), disco laser, disco ótico, disco versátil digital (DVD), disquetes e discos Blu-ray, onde discos rígidos normalmente reproduzem dados magneticamente, enquanto discos compactos reproduzem dados oticamente com lasers. As combinações desses últimos também devem ser incluídas no âmbito do meio legível por computador.

[0210] As instruções podem ser realizadas por um ou mais processadores, tais como um ou mais processadores de sinal digital (DSPs), microprocessadores de uso geral, circuitos integrados de aplicação específica (ASICs), matrizes de campos lógicos programáveis (FPGAs) ou outros circuitos equivalente de lógica integrada ou discreta. Nesse sentido, o termo "processador", como aqui utilizado pode se referir a qualquer um dentre as estruturas acima ou qualquer outra estrutura adequada para a aplicação das técnicas descritas neste documento. Além disso, em alguns aspectos, a funcionalidade descrita neste documento pode ser fornecida dentro de módulos de hardware e/ou software dedicado configurados para codificação e decodificação, ou incorporados em um codec combinado. Além disso, as técnicas podem ser totalmente implementadas em um ou mais circuitos ou elementos de lógica.

[0211] As técnicas desta descrição podem ser implementadas em uma ampla variedade de dispositivos ou aparelhos, compreendendo um aparelho de telefone sem fio, um circuito integrado (IC) ou um conjunto de ICs (por exemplo, um conjunto de chips). Vários componentes, módulos ou unidades são descritos nesta descrição para enfatizar os aspectos funcionais dos dispositivos configurados para realizar as técnicas descritas, mas que não necessariamente exigem a realização por diferentes unidades de hardware. Pelo contrário, como descrito acima, diversas unidades podem ser combinadas em uma unidade de hardware de codec ou fornecidas por uma coleção de unidades de hardware interoperativas, incluindo um ou mais processadores, como descrito acima, em conjunto com o software e/ou com o firmware adequados.

[0212] Vários exemplos da descrição foram apresentados. Qualquer combinação dos sistemas, operações ou funções descritos está contemplada. Estes e outros exemplos estão compreendidos no escopo das seguintes reivindicações.

REIVINDICAÇÕES

1. Método de codificar dados de vídeo em um processo de codificação, o método **caracterizado** pelo fato de que compreende:

determinar um modo de codificação utilizado para codificar um bloco atual de dados de vídeo, em que o modo de codificação é um dentre um modo de codificação com perdas ou um modo de codificação sem perdas;

determinar um processo de transformada de espaço-cor dependente do modo de codificação utilizado para codificar os dados de vídeo;

aplicar um processo de transformada de espaço-cor YCoCg a um bloco residual para o bloco atual quando o modo de codificação é determinado a ser o modo de codificação com perdas;

aplicar um processo de transformada de espaço-cor YCoCg-R a um bloco residual quando o modo de codificação é determinado a ser o modo de codificação sem perdas;

aplicar um processo de transformada inversa de espaço-cor YCoCg-R em um laço de decodificação do processo de codificação, independentemente de se o modo de codificação é o modo de codificação com perdas ou o modo de codificação sem perdas; e

quando o modo de codificação é determinado a ser o modo de codificação com perdas, modificar um componente Co e um componente Cg da matriz YCoCg com um deslocamento para esquerda de 1 bit antes de aplicar o processo de transformada inversa de espaço-cor no laço de decodificação do processo de codificação.

2. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que compreende adicionalmente aumentar uma profundidade de bit de um componente Co e um

componente Cg por 1 quando o modo de codificação é determinado a ser o modo de codificação sem perdas.

3. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que compreende adicionalmente determinar utilizar um processo de transformada de espaço-cor para codificar os dados de vídeo, em que determinar utilizar um processo de transformada de espaço-cor para codificar os dados de vídeo compreende:

definir um valor de um elemento de sintaxe que indica que uma unidade de codificação que compreende o bloco atual dos dados de vídeo foi codificada usando um processo de transformada de espaço-cor, em que o elemento de sintaxe compreende uma sinalização de um bit,

e em que um valor de 1 para o elemento de sintaxe indica que um processo de transformada de espaço-cor é usado para codificar a unidade de codificação.

4. Método, de acordo com a reivindicação 3, **caracterizado** pelo fato de que a unidade de codificação não é codificada no modo de modulação de código intra-pulso (IPCM), e

em que o elemento de sintaxe indica que a unidade de codificação foi codificada usando um processo de transformada de espaço-cor quando a unidade de codificação não é codificada usando um modo de modulação de código intra-pulso (IPCM); e/ou

em que o elemento de sintaxe indica que a unidade de codificação foi codificada usando um processo de transformada de espaço-cor quando não há coeficientes não-nulos em uma unidade de transformada da unidade de codificação.

5. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que um elemento de sintaxe que indica que uma unidade de codificação compreendendo o bloco

atual do dado de vídeo foi codificado usando um processo de transformada de espaço-cor não é sinalizado quando a unidade de codificação é intra codificada e quando um modo de predição de luma e um modo de predição de croma de uma unidade de predição dentro da unidade de codificação são diferentes; e/ou

em que em elemento de sintaxe que indica que a unidade de codificação compreende o bloco atual do dado de vídeo foi codificado usando um processo de transformada de espaço-cor não está sinalizado quando a unidade de codificação é codificada com um modo de paleta.

6. 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 um modo de codificação utilizado para codificar um bloco atual de dados de vídeo, em que o modo de codificação é um dentre um modo de codificação com perdas ou um modo de codificação sem perdas;

determinar um processo de transformada de espaço-cor dependente do modo de codificação utilizado para codificar os dados de vídeo;

aplicar um processo de transformada de espaço-cor YCoCg ao bloco residual de dados para o bloco atual quando o modo de codificação é determinado a ser o modo de codificação com perdas;

aplicar um processo de transformada de espaço-cor YCoCg-R ao bloco residual quando o modo de codificação é determinado a ser o modo de codificação sem perdas;

aplicar um processo de transformada inversa de espaço-cor YCoCg-R em um laço de decodificação do processo de codificação, independente de se o modo de codificação é o modo de codificação sem perdas ou o modo de codificação com perdas; e

quando o modo de codificação é determinado para ser o modo de codificação com perdas, modificar um componente Co e um componente Cg da matriz YCoCg com um deslocamento à esquerda de 1-bit antes de aplicar o processo de transformada inversa de espaço-cor no laço de decodificação do processo de codificação.

7. Dispositivo de codificação de vídeo, de acordo com a reivindicação 6, **caracterizado** pelo fato de que o um ou mais processadores são ainda configurados para aumentar uma profundidade de bit de um componente Co e um componente Cg por 1 quando o modo de codificação é determinado a ser um modo de codificação sem perdas.

8. Método de decodificação de dados de vídeo, o método **caracterizado** pelo fato de que compreende:

receber um primeiro bloco codificado de dados de vídeo, em que o primeiro bloco codificado de dados de vídeo foi codificado usando um modo de codificação com perdas e um processo de transformada de espaço-cor YCoCg;

receber um segundo bloco codificado de dados de vídeo, em que o segundo bloco de dados de vídeo codificado é um bloco residual, e em que o segundo bloco codificado de dados de vídeo foi codificado usando um modo de codificação sem perdas e um processo de transformada espaço-cor YCoCg-R;

aplicar um processo de transformada inversa de espaço-cor YCoCg-R ao primeiro bloco codificado de dados de vídeo; e

aplicar um processo de transformada inversa de espaço-cor YCoCg-R ao segundo bloco codificado de dados de vídeo;

em que o componente Co e o componente Cg do primeiro bloco codificado de dados de vídeo são modificados com um deslocamento para esquerda de 1 bit antes de aplicar o processo de transformada inversa de espaço-cor.

9. Método, de acordo com a reivindicação 8, **caracterizado** pelo fato de que compreende adicionalmente:

decodificar um elemento de sintaxe de uma unidade de codificação de dados de vídeo, em que a unidade de codificação compreende o primeiro bloco codificado ou o segundo bloco codificado, e em que o elemento de sintaxe indica se a unidade de codificação foi codificada usando um processo de transformação de espaço-cor;

determinar se um valor do elemento de sintaxe indica que a unidade de codificação foi codificada usando um processo de transformada de espaço-cor; e

em resposta a determinar que o elemento de sintaxe indica que a unidade de codificação foi codificada usando um processo de transformada de espaço-cor, aplicar o processo de transformada inversa de espaço-cor, em que o elemento de sintaxe compreende um sinalizador de um bit,

e em que um valor de 1 para o elemento de sintaxe indica que a unidade de codificação foi codificada usando um processo de transformada de espaço-cor.

10. Método, de acordo com a reivindicação 8, **caracterizado** pelo fato de que:

uma unidade de codificação compreendendo o primeiro bloco codificado ou o segundo bloco codificado não é codificada em um modo de modulação de código de intra-pulso (IPCM), e em que o elemento de sintaxe que indica se a unidade de codificação foi codificada usando um processo

de transformada de espaço-cor é sinalizado apenas para as unidades de codificação que não são codificadas usando um modo IPCM; e/ou em que um elemento de sintaxe indica que a unidade de codificação foi codificada usando um processo de transformada de espaço-cor quando os coeficientes não-zero em uma unidade de transformada da unidade de codificação; e/ou

em que um elemento de sintaxe indica que a unidade de codificação não foi codificada usando um processo de transformada de espaço-cor quando a unidade de codificação é intra codificada e quando um modo de previsão luma e o modo de previsão croma de uma unidade de previsão dentro da unidade de codificação são diferentes, em que o elemento de sintaxe não está presente em um fluxo de bits recebido que compreende o dado de vídeo, e em que decodificar o elemento de sintaxe compreende inferir o valor do elemento de sintaxe; e/ou

em que um elemento de sintaxe indica que a unidade de codificação não foi codificada usando um processo de transformada de espaço-cor 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 decodificar o elemento de sintaxe compreende inferir o valor do elemento de sintaxe.

11. 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:

receber um primeiro bloco codificado de dados de vídeo, em que o primeiro bloco codificado de dados

de vídeo foi codificado usando um modo de codificação com perdas e um processo de transformada de espaço-cor YCoCg;

receber um segundo bloco codificado de dados de vídeo, em que o segundo bloco de dados de vídeo codificado foi codificado usando um modo de codificação sem perdas e um processo de transformada de espaço-cor YCoCg-R;

aplicar um processo de transformada inversa de espaço-cor YCoCg-R para o primeiro bloco codificado de dados de vídeo; e

aplicar um processo de transformada inversa de espaço-cor YCoCg-R para o segundo bloco codificado de dados de vídeo,

em que o componente Co e o componente Cg do primeiro bloco codificado de dados de vídeo são modificados com um deslocamento de 1 bit para a esquerda antes de aplicar o processo de transformada inversa de espaço-cor.

12. Dispositivo de decodificação de vídeo, de acordo com a reivindicação 11, **caracterizado** pelo fato de que os um ou mais processadores são adicionalmente configurados para:

decodificar um elemento de sintaxe de uma unidade de codificação de dados de vídeo, em que a unidade de codificação compreende o primeiro bloco codificado ou o segundo bloco codificado, em que o elemento de sintaxe indica se a unidade de codificação foi codificada usando o processo de transformada de espaço-cor;

determinar se um valor do elemento de sintaxe indica que a unidade de codificação foi codificada usando um processo de transformada de espaço-cor; e

em resposta a determinar que o elemento de sintaxe indica que a unidade de codificação foi codificada usando um processo de transformada de espaço-cor, aplicar o

processo de transformada inversa de espaço-cor, em que o elemento de sintaxe compreende um sinalizador de um bit,

e em que um valor de 1 para o elemento de sintaxe indica que a unidade de codificação foi codificada usando um processo de transformada de espaço-cor.

13. Memória **caracterizada** pelo fato de que compreende instruções armazenadas na mesma, as instruções sendo executadas por um computador para realizar o método conforme definido em qualquer uma das reivindicações 1-5 ou 8-10.

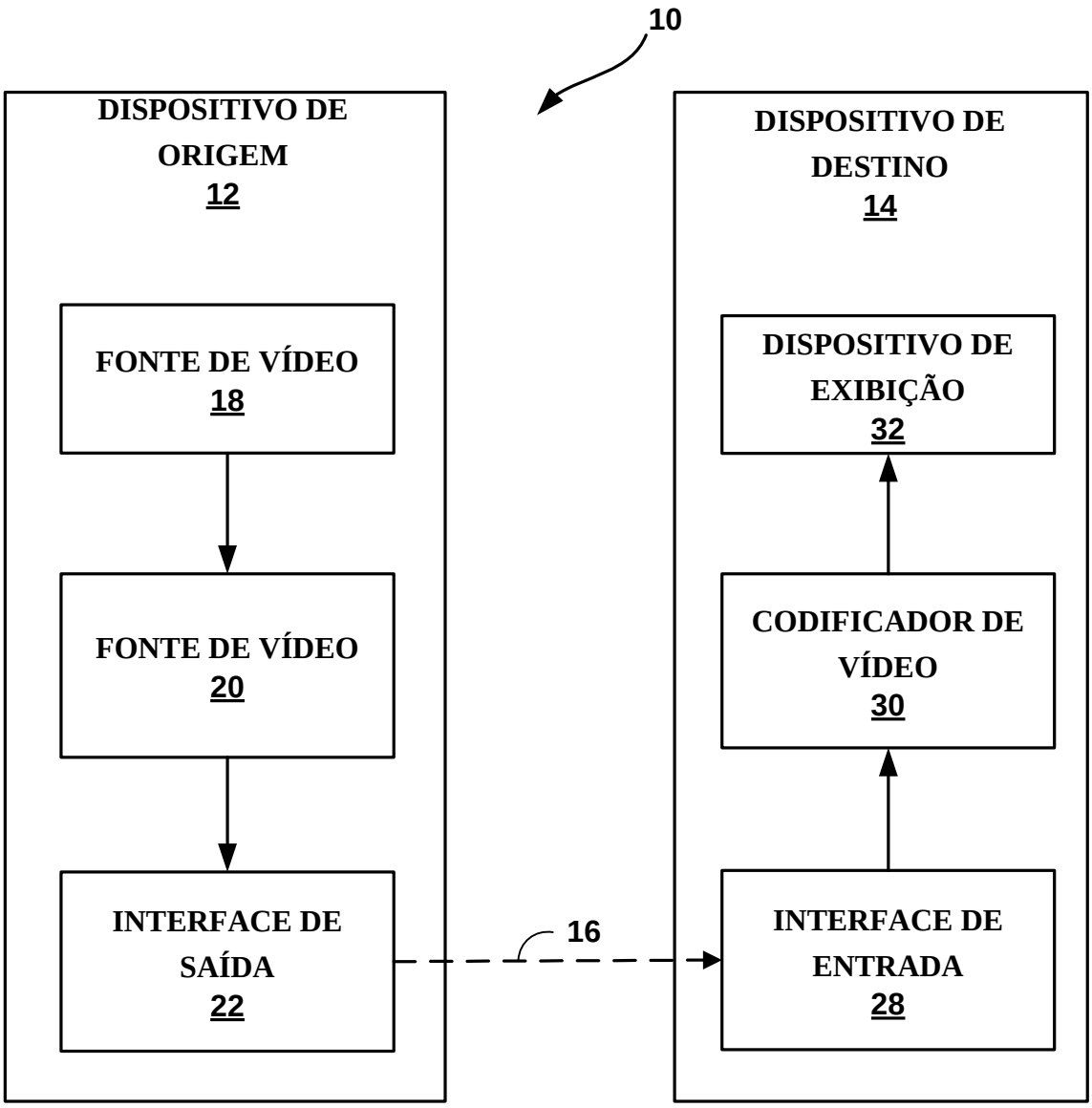

10
DISPOSITIVO DE ORIGEM
12
FONTE DE VÍDEO
18
FONTE DE VÍDEO
20
INTERFACE DE SAÍDA
22
16
DISPOSITIVO DE DESTINO
14
DISPOSITIVO DE EXIBIÇÃO
32
CODIFICADOR DE VÍDEO
30
INTERFACE DE ENTRADA
28

FIG. 1

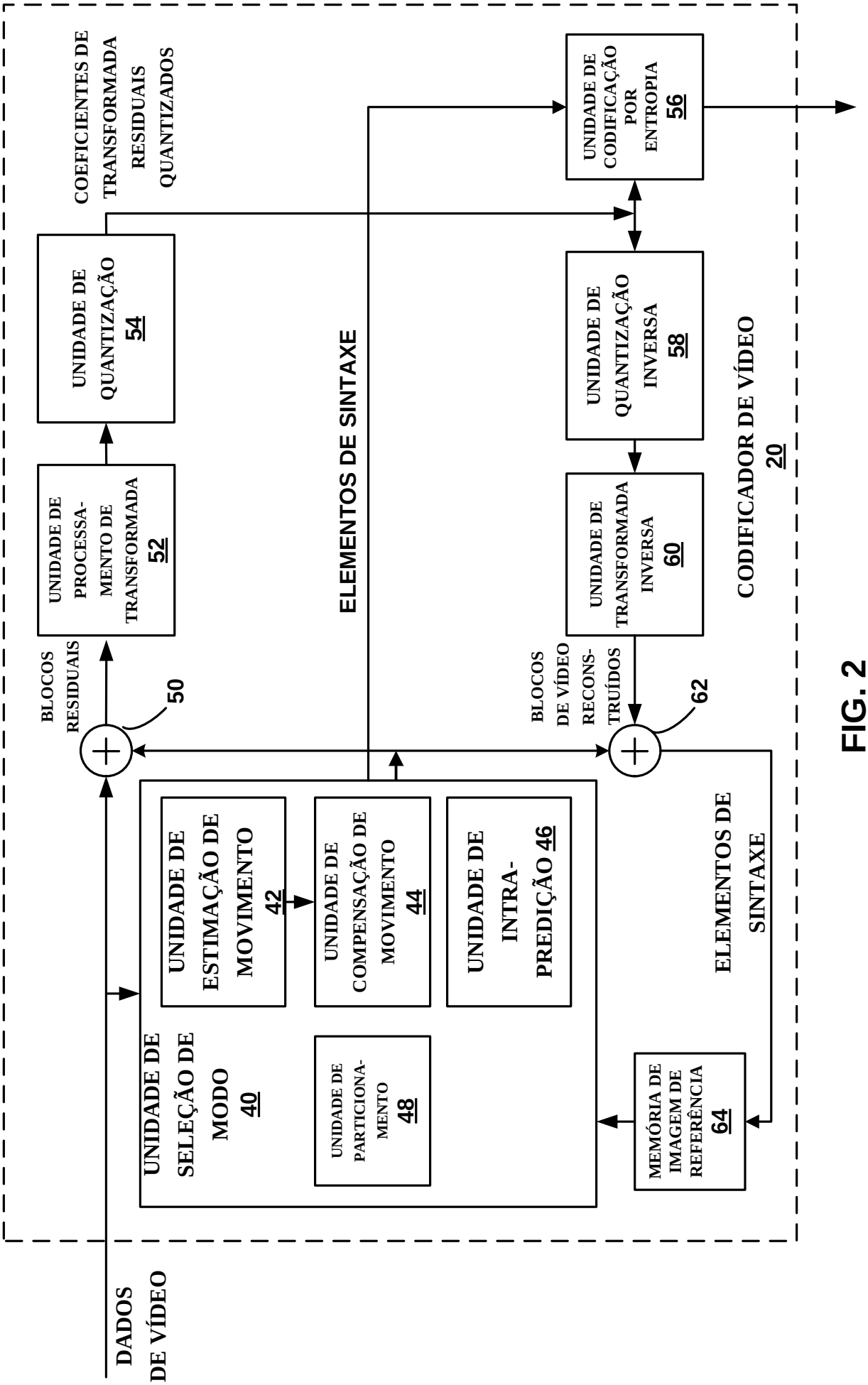

DADOS DE VÍDEO
CODIFICADOR DE VÍDEO 20
UNIDADE DE SELEÇÃO DE MODO 40
UNIDADE DE ESTIMAÇÃO DE MOVIMENTO 42
UNIDADE DE COMPENSAÇÃO DE MOVIMENTO 44
UNIDADE DE INTRA-PREDIÇÃO 46
UNIDADE DE PARTICIONA-MENTO 48
MEMÓRIA DE IMAGEM DE REFERÊNCIA 64
BLOCOS RESIDUAIS
50
UNIDADE DE PROCESSA-MENTO DE TRANSFORMADA 52
UNIDADE DE QUANTIZAÇÃO 54
COEFICIENTES DE TRANSFORMADA RESIDUAIS QUANTIZADOS
ELEMENTOS DE SINTAXE
UNIDADE DE CODIFICAÇÃO POR ENTROPIA 56
UNIDADE DE QUANTIZAÇÃO INVERSA 58
UNIDADE DE TRANSFORMADA INVERSA 60
BLOCOS DE VÍDEO RECONS-TRUÍDOS
62
ELEMENTOS DE SINTAXE

FIG. 2

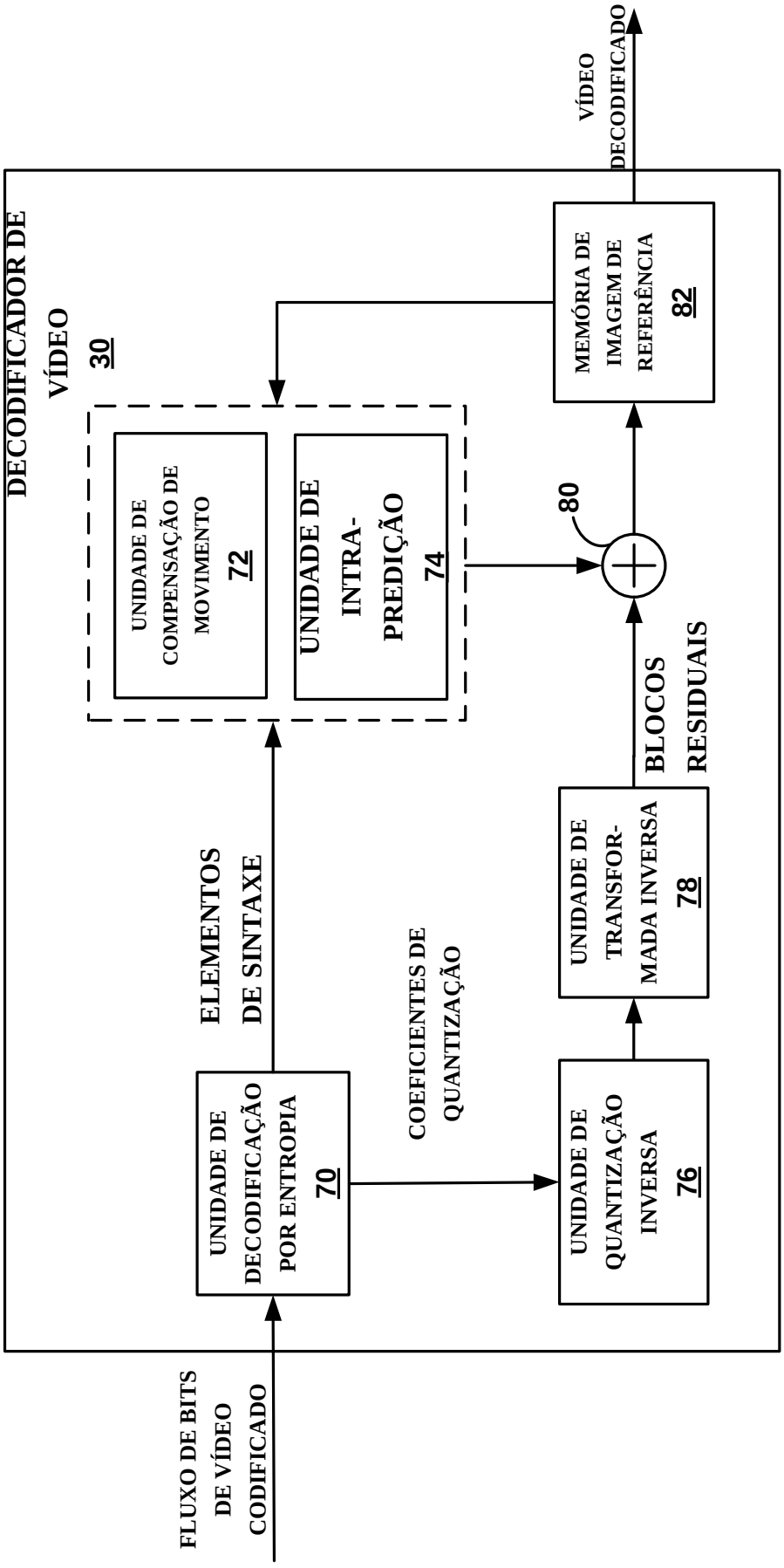

FLUXO DE BITS DE VÍDEO CODIFICADO
DECODIFICADOR DE VÍDEO 30
UNIDADE DE DECODIFICAÇÃO POR ENTROPIA 70
ELEMENTOS DE SINTAXE
UNIDADE DE COMPENSAÇÃO DE MOVIMENTO 72
UNIDADE DE INTRA-PREDIÇÃO 74
COEFICIENTES DE QUANTIZAÇÃO
UNIDADE DE QUANTIZAÇÃO INVERSA 76
UNIDADE DE TRANSFOR-MADA INVERSA 78
BLOCOS RESIDUAIS
80
MEMÓRIA DE IMAGEM DE REFERÊNCIA 82
VÍDEO DECODIFICADO

FIG. 3

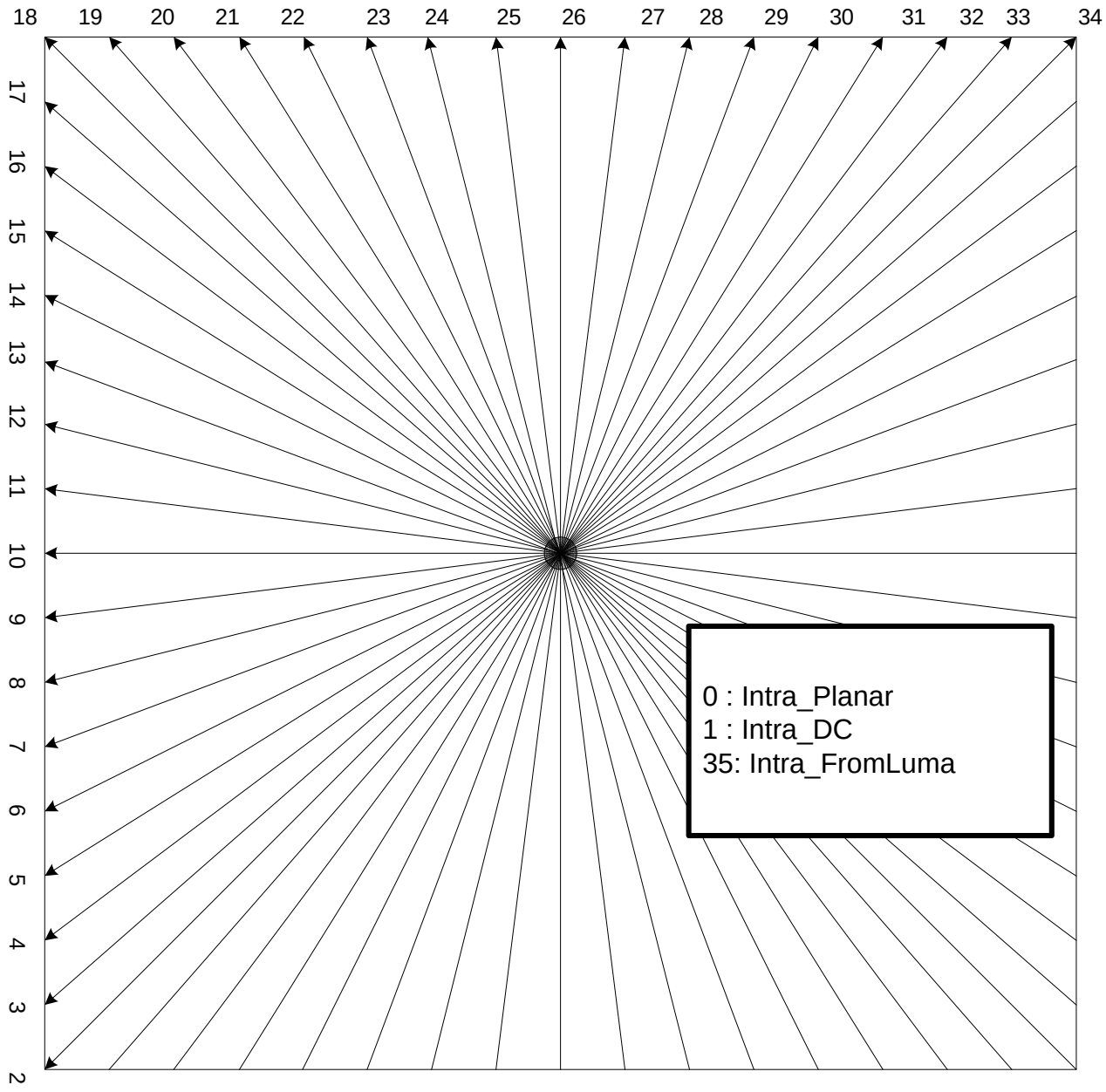
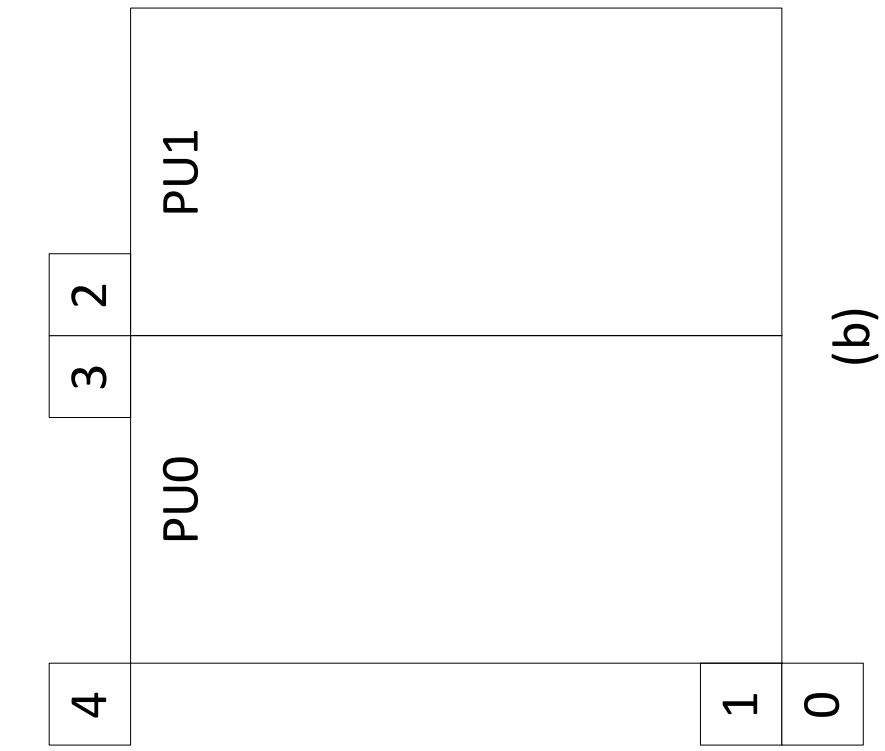

18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2
0 : Intra_Planar
1 : Intra_DC
35: Intra_FromLuma

FIG. 4



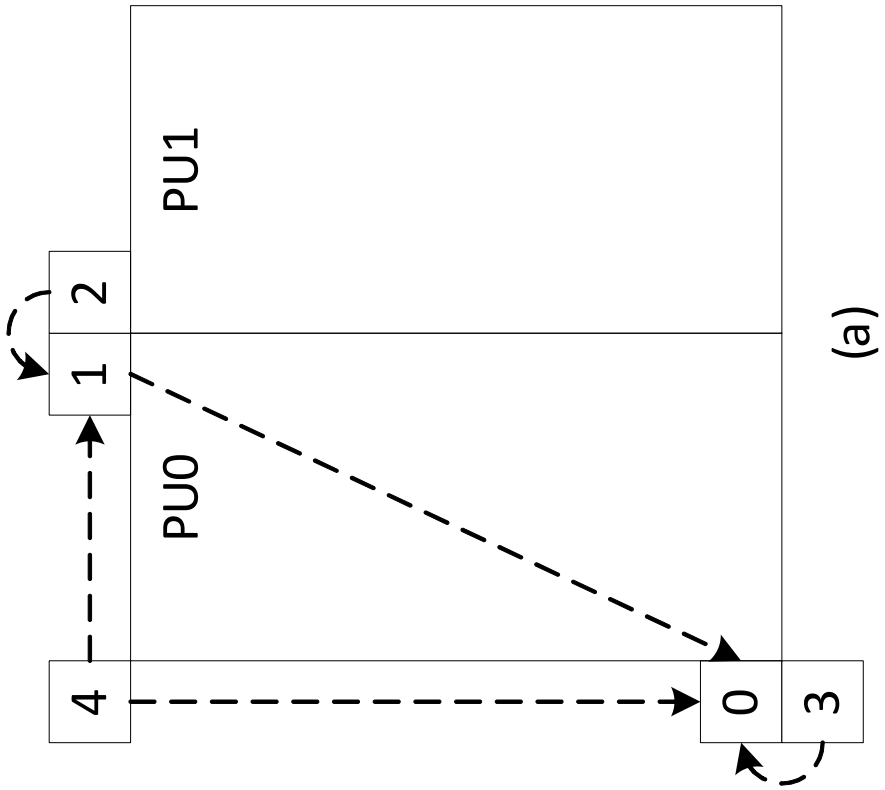


FIG. 5

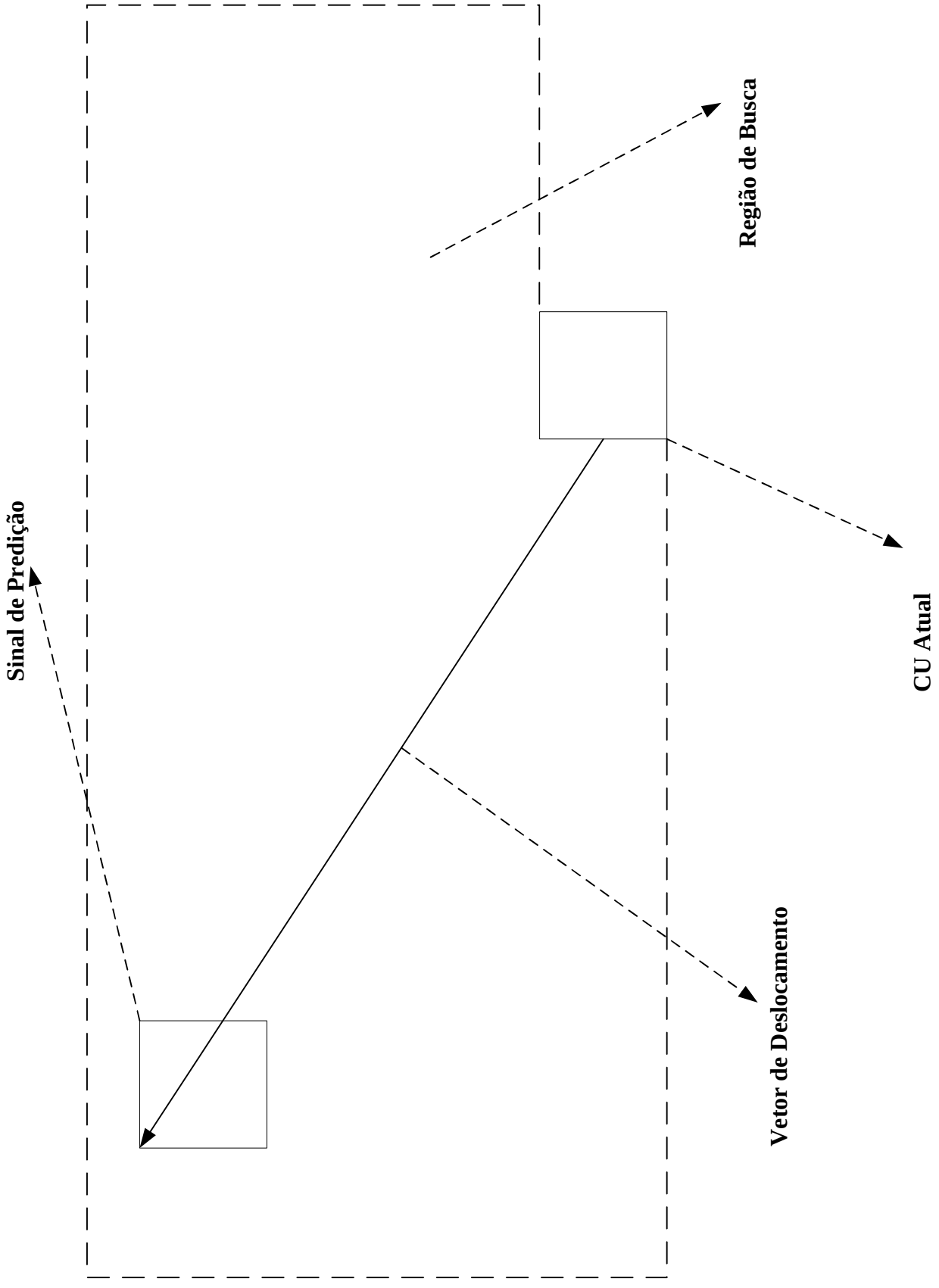

Região de Busca
CU Atual
Sinal de Predição
Vetor de Deslocamento

FIG. 6

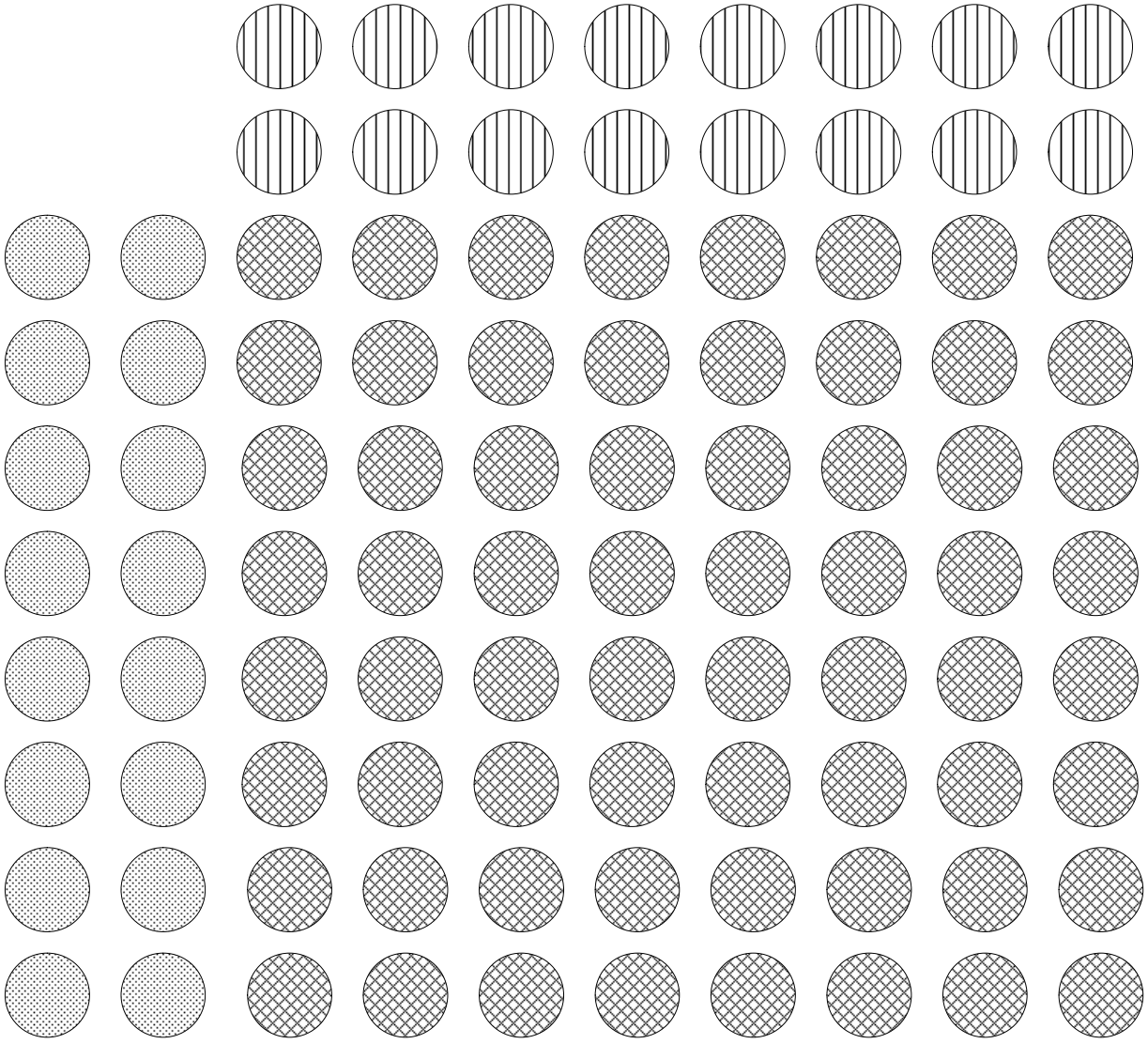


FIG. 7

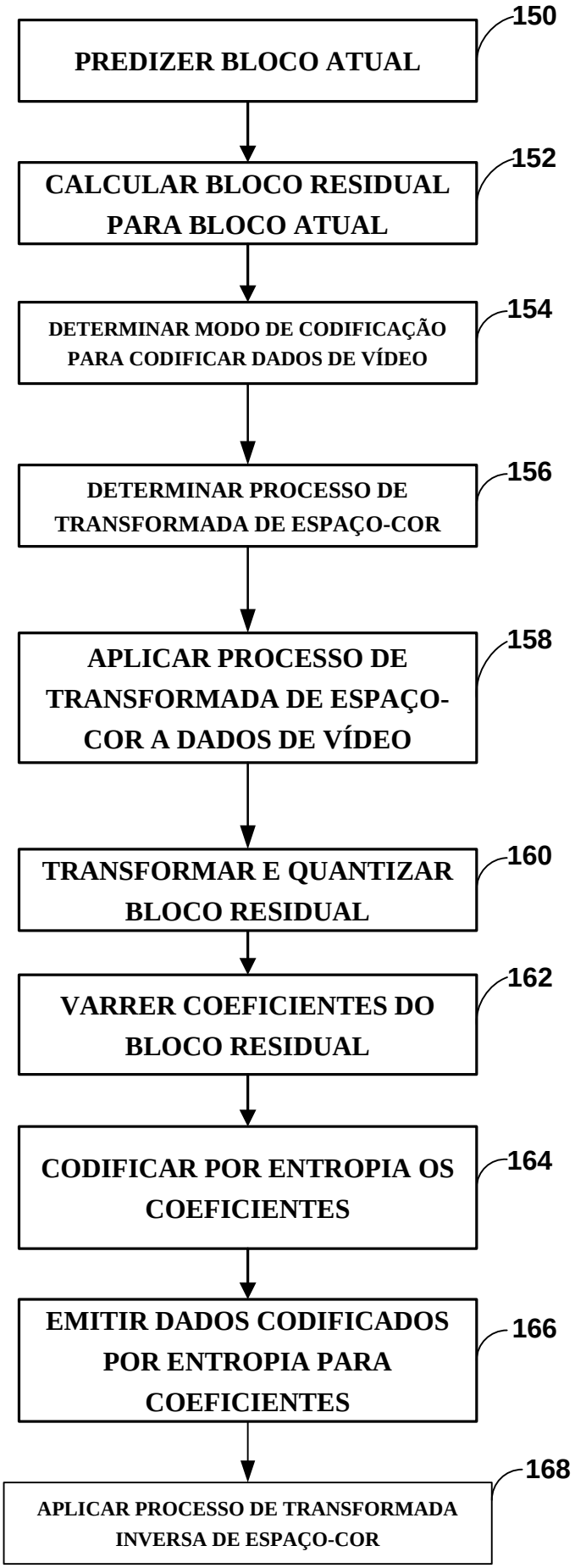

PREDIZER BLOCO ATUAL
150
CALCULAR BLOCO RESIDUAL PARA BLOCO ATUAL
152
DETERMINAR MODO DE CODIFICAÇÃO PARA CODIFICAR DADOS DE VÍDEO
154
DETERMINAR PROCESSO DE TRANSFORMADA DE ESPAÇO-COR
156
APLICAR PROCESSO DE TRANSFORMADA DE ESPAÇO-COR A DADOS DE VÍDEO
158
TRANSFORMAR E QUANTIZAR BLOCO RESIDUAL
160
VARRER COEFICIENTES DO BLOCO RESIDUAL
162
CODIFICAR POR ENTROPIA OS COEFICIENTES
164
EMITIR DADOS CODIFICADOS POR ENTROPIA PARA COEFICIENTES
166
APLICAR PROCESSO DE TRANSFORMADA INVERSA DE ESPAÇO-COR
168

FIG. 8

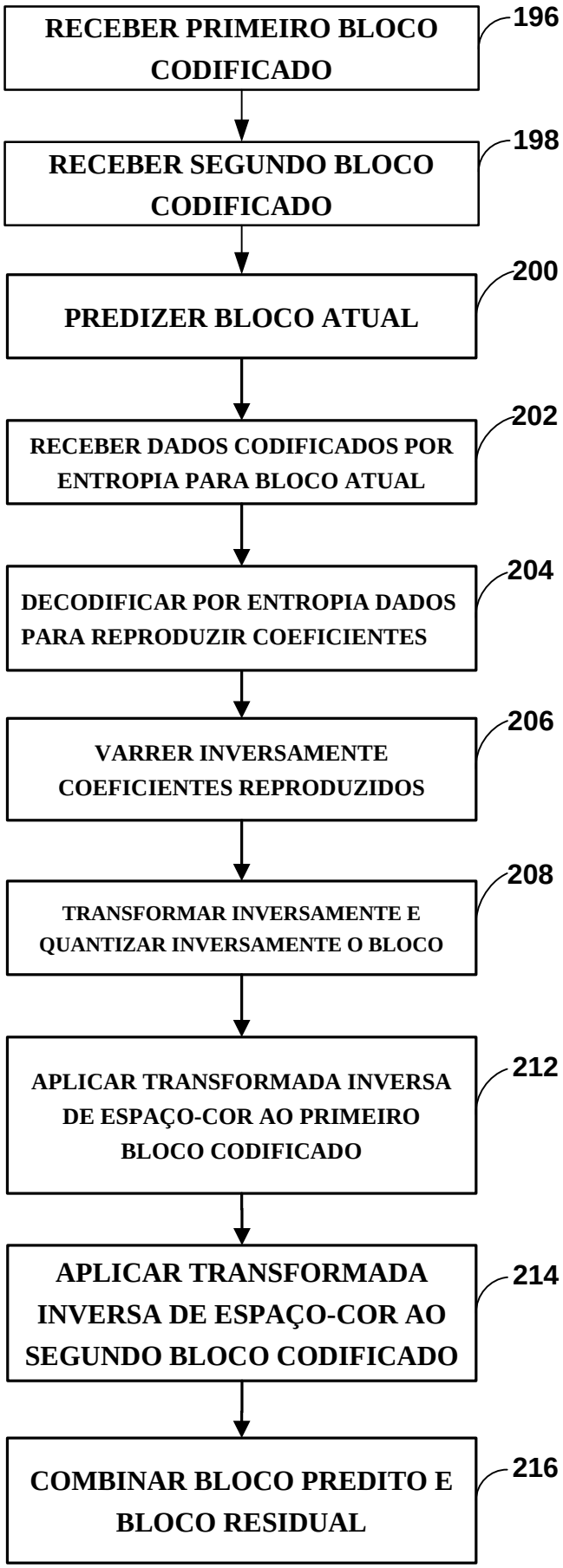

RECEBER PRIMEIRO BLOCO CODIFICADO
196
RECEBER SEGUNDO BLOCO CODIFICADO
198
PREDIZER BLOCO ATUAL
200
RECEBER DADOS CODIFICADOS POR ENTROPIA PARA BLOCO ATUAL
202
DECODIFICAR POR ENTROPIA DADOS PARA REPRODUZIR COEFICIENTES
204
VARRER INVERSAMENTE COEFICIENTES REPRODUZIDOS
206
TRANSFORMAR INVERSAMENTE E QUANTIZAR INVERSAMENTE O BLOCO
208
APLICAR TRANSFORMADA INVERSA DE ESPAÇO-COR AO PRIMEIRO BLOCO CODIFICADO
212
APLICAR TRANSFORMADA INVERSA DE ESPAÇO-COR AO SEGUNDO BLOCO CODIFICADO
214
COMBINAR BLOCO PREDITO E BLOCO RESIDUAL
216

FIG. 9