



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

(11) BR 112016025100-8 B1



(22) Data do Depósito: 01/05/2015

(45) Data de Concessão: 06/12/2022

(54) Título: MÉTODO E DISPOSITIVO PARA DECODIFICAR DADOS DE VÍDEO DE VÁRIAS CAMADAS, ASSIM COMO MEMÓRIA LEGÍVEL POR COMPUTADOR

(51) Int.Cl.: H04N 19/30; H04N 19/44; H04N 19/127; H04N 21/2343; H04N 21/45; (...).

(30) Prioridade Unionista: 01/05/2014 US 61/987,463; 15/05/2014 US 61/993,980; 30/04/2015 US 14/701,004.

(73) Titular(es): QUALCOMM INCORPORATED.

(72) Inventor(es): ADARSH KRISHNAN RAMASUBRAMONIAN; YE-KUI WANG; FNU HENDRY; KRISHNAKANTH RAPAKA.

(86) Pedido PCT: PCT US2015028841 de 01/05/2015

(87) Publicação PCT: WO 2015/168591 de 05/11/2015

(85) Data do Início da Fase Nacional: 26/10/2016

(57) Resumo: MÉTODO E DISPOSITIVO PARA DECODIFICAR DADOS DE VÍDEO DE MULTICAMADAS POR DETERMINAR A CAPACIDADE DO DECODIFICADOR COM BASE EM PERFIL, PLANO E NÍVEL ASSOCIADO À DIVISÃO CONTENDO UMA OU MAIS CAMADAS. Um decodificador de vídeo de multicamadas é configurado para determinar, com base em uma lista de entradas de tripleto, se o decodificador de vídeo de multicamadas é capaz de decodificar um fluxo de bits que compreende uma representação codificada dos dados de vídeo de multicamadas. O número de entradas de tripleto na lista é igual a um número de decodificadores de camada única no decodificador de vídeo de multicamadas. Cada entrada de tripleto respectiva na lista de entradas de tripleto indica um perfil, um plano e um nível para um decodificador de camada única respectiva no decodificador de vídeo de multicamadas. O decodificador de vídeo de multicamadas é configurado de modo que, com base no decodificador de vídeo de multicamadas sendo capaz de decodificar o fluxo de bits, o decodificador de vídeo de multicamadas decodifica o fluxo de bits.

"MÉTODO E DISPOSITIVO PARA DECODIFICAR DADOS DE VÍDEO DE VÁRIAS CAMADAS, ASSIM COMO MEMÓRIA LEGÍVEL POR COMPUTADOR"

[0001] O presente pedido reivindica o benefício do pedido de patente provisional US 61/987.463, depositado em 1 de maio de 2014, e pedido de patente provisional US 61/993.980, depositado em 15 de maio de 2014, o teor integral de cada um é incorporado aqui por referência.

CAMPO TÉCNICO

[0002] A presente revelação refere-se à codificação de vídeo e decodificação de vídeo.

ANTECEDENTES

[0003] Capacidades de vídeo digital podem ser incorporadas em uma ampla gama de dispositivos, incluindo televisões digitais, sistemas de broadcast direto digital, sistemas de broadcast sem fio, assistentes pessoais digitais (PDAs), computadores de mesa ou laptop, computadores de tablet, leitores de e-book, câmeras digitais, dispositivos de gravação digital, tocadores de mídia digital, dispositivos de jogos de vídeo, consoles de videogame, telefones de rádio satélite ou celulares, os denominados "smart phones", dispositivos de teleconferência de vídeo, dispositivos de streaming de vídeo e similares. Dispositivos de vídeo digital implementam técnicas de compressão de vídeo, como aquelas descritas nos padrões definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, parte 10, Codificação avançada de vídeo (AVC), padrão de Codificação de vídeo de eficiência elevada (HEVC) atualmente em desenvolvimento e extensões de tais padrões. Os dispositivos de vídeo podem transmitir, receber, codificar, decodificar e/ou armazenar informações de vídeo digital mais eficientemente por implementar tais técnicas de compressão de vídeo.

[0004] Técnicas de compressão de vídeo executam predição espacial (intra-imagem) e/ou temporal (inter-imagem) para reduzir ou remover redundância inerente em sequências de vídeo. Para codificação de vídeo baseada em bloco, uma fatia de vídeo (isto é, um quadro de vídeo ou uma porção de um quadro de vídeo) pode ser dividida em blocos de vídeo. Blocos de vídeo em uma fatia intra-codificada (I) de uma imagem são codificadas usando predição espacial com relação a amostras de referência em blocos vizinhos na mesma imagem. Blocos de vídeo em uma fatia inter-codificada (P ou B) de uma imagem são codificados usando predição espacial com relação a amostras de referência em blocos vizinhos na mesma imagem. Blocos de vídeo em uma fatia inter-codificada (P ou B) de uma imagem podem usar predição espacial com relação a amostras de referência em blocos vizinhos na mesma imagem ou predição temporal com relação a outras imagens de referência. Imagens podem ser mencionadas como quadros, e imagens de referência podem ser mencionadas como quadros de referência.

[0005] Predição espacial ou temporal resulta em um bloco preditivo para um bloco a ser codificado. Dados residuais representam diferenças de pixel entre o bloco original a ser codificado e o bloco preditivo. Um bloco inter-codificado é codificado de acordo com um vetor de movimento que aponta para um bloco de amostras de referência formando o bloco preditivo, e os dados residuais 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 os dados residuais. Para compressão adicional, os dados residuais podem ser transformados a partir do domínio espacial em um domínio de transformada, resultando em coeficientes de transformada

residual, que então podem ser quantizados. Os coeficientes quantizados, inicialmente dispostos em uma disposição bidimensional, podem ser varridos em uma ordem específica para produzir um vetor unidimensional de coeficientes e codificação por entropia pode ser aplicada para obter ainda mais compressão.

[0006] Um fluxo de bits de codificação de multivistas pode ser gerado por codificar vistas, por exemplo, a partir de múltiplas perspectivas. Alguns padrões de vídeo tridimensional (3D) foram desenvolvidos que fazem uso de aspectos de codificação de multivistas. Por exemplo, vistas diferentes podem transmitir vistas do olho esquerdo e direito para suportar vídeo 3D. alternativamente, alguns processos de codificação de vídeo 3D podem aplicar a denominada codificação de multivistas mais profundidade. Em codificação de multivistas mais profundidade, um fluxo de bits de vídeo 3D pode conter não somente componentes de vista de textura, como também componentes de vista de profundidade. Por exemplo, cada vista pode compreender um componente de vista de textura e um componente de vista de profundidade.

SUMÁRIO

[0007] Em geral, a presente revelação descreve técnicas de descrever propriedades de fluxo de bit de vídeo usadas para determinar, com base em informação de perfil, plano e nível especificada para divisões de fluxo de bit de vídeo respectivas, a capacidade de um decodificador de vídeo para decodificar um fluxo de bits de vídeo.

[0008] Em outro exemplo, essa revelação descreve um método de decodificar dados de vídeo de multicamadas, o método compreendendo: determinar, com base em uma lista de entradas de triplete, se um decodificador de vídeo de multicamadas é capaz de decodificar um fluxo de

bits que compreende uma representação codificada dos dados de vídeo de multicamadas, em que cada entrada de tripleto respectiva na lista de entradas de tripleto indica um perfil, um plano e um nível para um decodificador de camada única respectivo no decodificador de vídeo de multicamadas; e com base no decodificador de vídeo de multicamadas sendo capaz de decodificar o fluxo de bits, decodificar, pelo decodificador de vídeo de multicamadas, o fluxo de bits.

[0009] Em outro exemplo, essa revelação descreve um dispositivo para decodificar dados de vídeo de multicamadas, o dispositivo compreendendo: uma ou mais mídias de armazenagem legível em computador configuradas para armazenar os dados de vídeo de multicamadas; e um ou mais processadores configurados para: determinar com base em uma lista de entradas de tripleto, se um decodificador de multicamadas é capaz de decodificar um fluxo de bits que compreende uma representação codificada dos dados de vídeo de multicamadas, em que cada entrada de tripleto respectiva na lista de entradas de tripleto indica um perfil, um plano e um nível para um decodificador de camada única respectivo no decodificador de vídeo de multicamadas.

[0010] Em outro exemplo, essa revelação descreve um dispositivo para decodificar dados de vídeo, o dispositivo compreendendo: meio para determinar, com base em uma lista de entradas de tripleto, se o dispositivo é capaz de decodificar um fluxo de bits que compreende uma representação codificada dos dados de vídeo de multicamadas, em que cada entrada de tripleto respectiva na lista de entradas de tripleto indica um perfil, um plano e um nível para um decodificador de camada única respectivo no dispositivo; e meio para decodificar, com base no dispositivo sendo capaz de decodificar o fluxo de bits, o fluxo de bits.

[0011] Em outro exemplo, essa revelação descreve uma mídia de armazenagem de dados legível em computador compreendendo instruções armazenadas na mesma que quando executadas fazem com que um decodificador de vídeo de multicamadas: determine, com base em uma lista de entradas de triplete, se o decodificador de vídeo de multicamadas é capaz de decodificar um fluxo de bits que compreende uma representação codificada dos dados de vídeo de multicamadas, em que cada entrada de triplete respectiva na lista de entradas de triplete indica um perfil, um plano e um nível para um decodificador de camada única respectivo no decodificador de vídeo de multicamadas; e com base no decodificador de vídeo de multicamadas sendo capaz de decodificar o fluxo de bits, decodificar o fluxo de bits.

[0012] Os detalhes de um ou mais exemplos da revelação são expostos nos desenhos em anexo e descrição abaixo. Outras características, objetivos e vantagens serão evidentes a partir da descrição, desenhos e reivindicações.

BREVE DESCRIÇÃO DOS DESENHOS

[0013] A figura 1 é um diagrama de blocos ilustrando um sistema de codificação de vídeo de exemplo que pode utilizar as técnicas descritas nessa revelação.

[0014] A figura 2 é um diagrama de blocos ilustrando uma implementação de exemplo de um decodificador de vídeo compreendendo uma pluralidade de decodificadores de camada única, de acordo com uma técnica dessa revelação.

[0015] A figura 3 é um diagrama de blocos ilustrando um codificador de vídeo de exemplo que pode implementar técnicas descritas nessa revelação.

[0016] A figura 4 é um diagrama de blocos ilustrando um decodificador de vídeo de exemplo que pode implementar técnicas descritas nessa revelação.

[0017] A figura 5A é um fluxograma ilustrando

uma operação de exemplo de um codificador de vídeo, de acordo com uma técnica dessa revelação.

[0018] A figura 5B é um fluxograma ilustrando uma operação de exemplo de um decodificador de vídeo, de acordo com uma técnica dessa revelação.

[0019] A figura 6A é um fluxograma ilustrando uma operação de exemplo de um codificador de vídeo, de acordo com uma técnica dessa revelação.

[0020] A figura 6B é um fluxograma ilustrando uma operação de exemplo de um decodificador de vídeo, de acordo com uma técnica dessa revelação.

[0021] A figura 7 é um fluxograma ilustrando uma operação de exemplo de um decodificador de vídeo, de acordo com uma técnica dessa revelação.

DESCRIÇÃO DETALHADA

[0022] Um fluxo de bits (isto é, um fluxo de bits de vídeo) pode compreender uma representação codificada de dados de vídeo. Mais especificamente, o fluxo de bits pode compreender uma série de unidades de Camada de Abstração de rede (NAL). Cada das unidades NAL pode incluir um cabeçalho de unidade NAL e uma Carga útil de Sequência de byte bruta (RBSP). Na codificação de vídeo de multicamadas, unidades NAL podem corresponder a várias camadas de dados de vídeo. Em codificação de vídeo escalonável, as camadas podem incluir uma camada de base e uma ou mais camadas de aperfeiçoamento. Em codificação de multivistas, cada das camadas pode corresponder a vistas diferentes, como vistas diferentes de uma cena de vídeo.

[0023] Certos padrões de codificação de vídeo, como Codificação de vídeo de eficiência elevada (HEVC), definem vários perfis, planos e níveis diferentes. Em geral, um "perfil" de um padrão de codificação de vídeo é um subconjunto das características e ferramentas presentes

no padrão de codificação de vídeo. Em outras palavras, um perfil define quais ferramentas de codificação podem ser usadas. Por exemplo, para um codificador de vídeo, um perfil pode ser definido de ferramentas de codificação que o codificador de vídeo pode usar para gerar fluxos de bits codificados que se conformam ao perfil. Para um decodificador de vídeo, um perfil pode significar o conjunto de ferramentas de codificação que o decodificador de vídeo deve ter para ser capaz de decodificar fluxos de bits que se diz se conformarem ao perfil.

[0024] Um nível é um conjunto definido de limitações sobre os valores que podem ser tomados pelos elementos de sintaxe e variáveis de um padrão de codificação de vídeo. Um plano é uma categoria especificada de limitações de nível impostas em valores dos elementos de sintaxe no fluxo de bits ou valores de variáveis, onde as limitações de nível são encaixadas em um plano e um decodificador se conformando a certo plano e nível seria capaz de decodificar todos os fluxos de bits que se conformam ao mesmo plano ou ao plano inferior daquele nível ou qualquer nível abaixo daquele nível. Desse modo, um nível de um plano é um conjunto especificado de limitações impostas em valores dos elementos de sintaxe no fluxo de bits ou variáveis usadas na decodificação do fluxo de bits.

[0025] Decodificadores de vídeo diferentes podem ter capacidades diferentes. Por exemplo, um decodificador de vídeo pode ser capaz de decodificar fluxos de bits se conformando a um primeiro perfil, porém pode não ser capaz de decodificar fluxos de bits se conformando a um segundo perfil. Nesse exemplo, outro decodificador de vídeo pode ser capaz de decodificar fluxos de bits se conformando ao primeiro perfil e ao segundo perfil. Por exemplo, com camadas escalonáveis de codificação de vídeo, o nível de

qualidade de vídeo que pode ser obtido pode ser aumentado através de camadas de modo que alguns dispositivos podem ser capazes somente de decodificar e apresentar um nível base de qualidade de vídeo enquanto outros dispositivos podem ser capazes de decodificar e apresentar qualidade de vídeo aperfeiçoada. Também, camadas escalonáveis podem permitir recepção de dados de vídeo mais confiáveis quando a largura de banda de canal é limitada, por exemplo, um canal de largura de banda limitada pode ainda ser suficiente para comunicação de dados de vídeo da camada de base e canais de largura de banda mais robustos podem permitir comunicação de dados de vídeo de camadas de base e aperfeiçoamento. Em geral, se um primeiro processador for capaz de decodificar fluxos de bits se conformando a perfis, planos e níveis que são um superconjunto de perfis, planos e níveis que podem ser decodificados por um segundo processador, o primeiro processador é mais complexo e portanto tende a ser mais caro.

[0026] Similarmente, com codificação de multivistas, múltiplas vistas podem ser decodificáveis por alguns dispositivos, enquanto outros dispositivos podem ter somente capacidades de decodificar um número limitado de vistas. Limitações de largura de banda de canal podem ser também mais efetivamente utilizadas por codificação de multivista de modo que um número limitado de vistas seja decodificável quando largura de banda é limitada, e um número maior de vistas pode ser decodificável quando largura de banda permite.

[0027] Em algumas instâncias, um decodificador de vídeo compreende múltiplos núcleos de hardware (por exemplo, unidades de processamento ou módulos de processamento). Em outras palavras, um decodificador de vídeo pode ser construído usando uma pluralidade de núcleos

de hardware. Cada dos núcleos de hardware pode ser capaz de independentemente decodificar pelo menos elementos de sintaxe de baixo nível de um fluo de bits. Os núcleos diferentes podem ser similares ou diferentes em suas capacidades de processamento, porém independentemente, a presença de múltiplos núcleos podem fornecer vantagens de processamento em relação a decodificadores de núcleo único. Em alguns exemplos, um decodificador de vídeo compreendendo múltiplos núcleos de hardware pode se capaz de decodificar múltiplas porções de um fluxo de bits em paralelo. Por exemplo, alguns núcleos de processamento podem implementar um decodificador de camada única. Um decodificador de camada única pode ser capaz de decodificar uma única camada de um fluxo de bits, como uma camada de base de vídeo escalonável ou uma vista específica em vídeo de multivistas. Em contraste, um decodificador de multicamadas pode ser capaz de decodificar múltiplas camadas de um fluxo de bits simultaneamente (isto é, em paralelo). Em um exemplo, um núcleo de hardware único é capaz de decodificar mais de uma camada.

[0028] Núcleos de hardware diferentes de um decodificador de vídeo podem ter capacidades diferentes. Por exemplo, um núcleo de hardware de um decodificador de vídeo pode ser capaz de decodificar fluxos de bits se conformando a um primeiro perfil, porém não um segundo perfil, enquanto um segundo núcleo de hardware de um decodificador de vídeo pode ser capaz de decodificar fluxos de bit se conformando ao primeiro perfil e ao segundo perfil. Um decodificador de vídeo pode ter núcleos de hardware tendo capacidades diferentes para manter o custo geral do decodificador de vídeo tão baixo quanto possível, dada a finalidade pretendida do decodificador de vídeo.

[0029] Tipicamente, em instâncias onde um

decodificador de vídeo compreende múltiplos núcleos de hardware, o decodificador de vídeo determina, com base em informação de perfil, plano e nível sinalizada em um fluxo de bits, se um núcleo de hardware mais capaz do decodificador de vídeo é capaz de decodificar o fluxo de bits. Em caso positivo, o decodificador de vídeo determina que é capaz de decodificar o fluxo de bits. Esse processo não permite que o decodificador de vídeo determine se quaisquer dos núcleos de hardware menos capazes do decodificador de vídeo é capaz de auxiliar no processo de decodificação do fluxo de bits. Como resultado, os núcleos de hardware menos capazes podem ser inativos enquanto o núcleo de hardware mais capaz decodifica todo o fluxo de bits. Entretanto, os núcleos de hardware menos capazes do decodificador de vídeo podem, na realidade, ser capazes de decodificar camadas específicas do fluxo de bits em paralelo com o núcleo de hardware mais capaz do decodificador de vídeo. Como resultado, não usar os núcleos de hardware menos capazes pode desnecessariamente prolongar tempos de decodificação.

[0030] Algumas técnicas dessa revelação pode facilitar reduções em tempos de decodificação em decodificadores de vídeo tendo núcleos de hardware com capacidades diferentes. Por exemplo, algumas técnicas dessa revelação podem diminuir tempos de decodificação em decodificadores de vídeo construídos com base em múltiplos decodificadores de camada única tendo capacidades diferentes. Essa redução em tempos de decodificação em relação a técnicas mais convencionais pode ser realizada sem alterar os próprios núcleos de hardware (por exemplo, sem modificar as estruturas físicas dos núcleos de hardware). Adicionalmente, quando núcleos de hardware têm as mesmas capacidades, algumas técnicas dessa revelação

podem habilitar decodificadores de vídeo a decodificar mais camadas em paralelo.

[0031] De acordo com uma técnica de exemplo dessa revelação, um decodificador de vídeo pode decodificar um fluxo de bits que compreende uma representação codificada de dados de vídeo. Nesse exemplo, o fluxo de bits compreende uma pluralidade de camadas. Como parte de decodificar o fluxo de bits, o decodificador de vídeo pode obter, a partir do fluxo de bits, um ou mais elementos de sintaxe que indicam um ou mais esquemas de divisão. Para cada esquema de divisão respectivo de um ou mais esquemas de divisão, o esquema de divisão respectivo especifica um conjunto respectivo de divisões desconexas cuja união forma um conjunto de camadas de saída. Cada divisão respectiva do conjunto respectivo de divisões desconexas contém uma ou mais das camadas. Consequentemente, uma divisão pode ser um tipo específico de subfluxo de bits do fluxo de bits. O conjunto de camada de saída pode consistir em todas as camadas da pluralidade de camadas de um subconjunto adequado da pluralidade de camadas. Além disso, o decodificador de vídeo pode decodificar cada das divisões de um esquema de divisão específico usando núcleos de processamento diferentes em uma pluralidade de núcleos de hardware, o esquema de divisão específico sendo um de um ou mais esquemas de divisão.

[0032] Em alguns exemplos, para cada divisão respectiva de cada esquema de divisão respectivo de um ou mais esquemas de divisão, o decodificador de vídeo obtém, a partir do fluxo de bits, um elemento de sintaxe respectivo que indica informação de perfil, plano e nível (PTL) respectiva para a respectiva divisão. Em algumas instâncias, o decodificador de vídeo pode selecionar, com base na informação PTL para divisões dos esquemas de

divisão, um esquema de divisão a partir de um ou mais esquemas de divisão. Em algumas instâncias, o decodificador de vídeo pode receber o fluxo de bits de acordo com um esquema de divisão específico como determinado pelos recursos de rede disponíveis. O decodificador de vídeo pode decodificar cada divisão respectiva do esquema de divisão específico usando um núcleo respectivo dos núcleos de hardware. Cada núcleo respectivo dos núcleos de hardware pode implementar um decodificador de camada única respectivo. Em algumas instâncias, um núcleo de hardware do decodificador de vídeo pode ser capaz de decodificar mais de uma camada.

[0033] Técnicas adicionais dessa revelação podem fornecer teste de conformidade de fluxo de bit usando as divisões de fluxo de bit. Por exemplo, um dispositivo de processamento de vídeo (como um codificador de vídeo, decodificador de vídeo ou outro dispositivo) pode decodificar um fluxo de bits que compreende uma representação codificada de dados de vídeo. O fluxo de bits pode compreender uma pluralidade de camadas. Como parte de decodificar o fluxo de bits, o decodificador de vídeo pode obter, a partir do fluxo de bits, um ou mais elementos de sintaxe indicando um ou mais esquemas de divisão para um conjunto de camadas de saída. Nesses tipos de exemplos, o conjunto de camada de saída pode consistir em todas as camadas da pluralidade de camadas ou um subconjunto adequado da pluralidade de camadas (por exemplo, não todas as camadas da pluralidade de camadas). Além disso, para cada esquema de divisão respectivo de um ou mais esquemas de divisão, o respectivo esquema de divisão pode especificar uma divisão do conjunto de camada de saída em uma ou mais divisões. Cada camada no conjunto de camadas de saída pode pertencer exatamente a uma divisão no esquema de

divisão respectivo. Para cada esquema de divisão respectivo de um ou mais esquemas de divisão, o decodificador de vídeo pode obter, a partir de uma mensagem de Informação de Aperfeiçoamento suplementar (SEI) no fluxo de bits ou informação de capacidade de uso de vídeo (VUI) em um conjunto de parâmetros de vídeo (VPS) no fluxo de bits, um conjunto respectivo de elementos de sintaxe indicando conjuntos de parâmetro de decodificador de referência hipotético (HRD) para o esquema de divisão. Além disso, para cada divisão respectiva do esquema de divisão, o dispositivo de processamento de vídeo pode testar, com base nos conjuntos de parâmetro HRD para o esquema de divisão respectivo, a divisão respectiva para conformidade com um padrão de codificação de vídeo, como MV-HEVC ou SHVC.

[0034] De acordo com algumas técnicas dessa revelação, um decodificador de vídeo de multicamadas pode determinar, com base em uma lista de entradas de triplete, se o decodificador de vídeo de multicamadas é capaz de decodificar (isto é, configurado para decodificar) um fluxo de bits que compreende uma representação codificada dos dados de vídeo de multicamadas. O número de entradas de triplete na lista pode ser menor ou igual a um número de decodificadores de camada única no decodificador de vídeo de multicamadas. Cada entrada de triplete respectivo na lista de entradas de triplete indica um perfil, um plano e um nível que um decodificador de camada única respectivo no decodificador de vídeo de multicamadas é capaz de decodificar (isto é, e configurado para decodificar). Com base no decodificador de vídeo de multicamadas sendo capaz de decodificar o fluxo de bits, o decodificador de vídeo de multicamadas pode decodificar o fluxo de bits. Em alguns exemplos, o dispositivo de processamento de vídeo pode determinar que o decodificador de multicamadas seja capaz

de decodificar o fluxo de bits por determinar que o fluxo de bits especifica um esquema de divisão onde a informação PTL de cada divisão do esquema de divisão indica que a divisão pode ser decodificada por pelo menos um dos decodificadores de camada única. Isso pode permitir que um decodificador de vídeo de multicamadas tendo múltiplos decodificadores de camada única determine se o decodificador de vídeo de multicamadas é capaz de decodificar o fluxo de bits.

[0035] A figura 1 é um diagrama de blocos que ilustra um sistema de codificação de vídeo de exemplo 10 que pode utilizar as técnicas dessa revelação. Como usado aqui, o termo "codificador de vídeo" se refere em geral tanto a codificadores de vídeo como decodificadores de vídeo. Nessa revelação, os termos "codificação de vídeo" ou "codificação" podem se referir genericamente a codificação de vídeo ou decodificação de vídeo.

[0036] Como mostrado na figura 1, o sistema de codificação de vídeo 10 inclui um dispositivo de fonte 12, um dispositivo de destino 14 e um elemento de rede 15. O dispositivo de fonte 12 gera dados de vídeo codificados. Por conseguinte, o dispositivo de fonte 12 pode ser mencionado como um dispositivo de codificação de vídeo ou um aparelho de codificação de vídeo. O dispositivo de destino 14 pode decodificar os dados de vídeo codificados gerados pelo dispositivo de fonte 12. Por conseguinte, o dispositivo de destino 14 pode ser mencionado como um dispositivo de decodificação de vídeo ou um aparelho de decodificação de vídeo. O dispositivo de fonte 12 e o dispositivo de destino 14 podem ser exemplos de dispositivos de codificação de vídeo ou aparelhos de codificação de vídeo.

[0037] O dispositivo de fonte 12 e o

dispositivo de destino 14 podem compreender uma ampla gama de dispositivos, incluindo computadores de mesa, dispositivos de computação móvel, computadores notebook (por exemplo, laptop), computadores tablet, conversores de sinais de frequência, aparelhos telefônicos como os denominados "smart phones", televisões, câmeras, dispositivos de exibição, tocadores de mídia digital, consoles de jogo de vídeo, computadores no carro, ou similares.

[0038] O elemento de rede 15 pode receber dados de vídeo codificados e transmitir dados de vídeo codificados processados. O elemento de rede 15 pode ser um elemento de rede ciente de mídia (MANE), dispositivo de rede de fornecimento de conteúdo (CDN), ou outro tipo de dispositivo (por exemplo, dispositivo de computação). O dispositivo de rede 15 pode executar uma ou mais técnicas dessa revelação. Por exemplo, o dispositivo de rede 15 pode executar um ou mais testes de conformidade de fluxo de bits, de acordo com uma ou mais técnicas dessa revelação. O elemento de rede 15, dispositivo de fonte 12, dispositivo de destino 14 e outros tipos de dispositivo que processam dados de vídeo podem ser considerados dispositivos de processamento de vídeo.

[0039] O dispositivo de destino 14 pode receber dados de vídeo codificados a partir do dispositivo de fonte 12 através de um canal 16. O canal 16 pode compreender uma ou mais mídias ou dispositivos capazes de mover os dados de vídeo codificados a partir do dispositivo de fonte 12 para o dispositivo de destino 14. Em um exemplo, o canal 16 pode compreender uma ou mais mídias de comunicação que permitem que o dispositivo de fonte 12 transmita dados de vídeo codificados diretamente para o dispositivo de destino 14 em tempo real. Nesse exemplo, o

dispositivo de fonte 12 pode modular os dados de vídeo codificados de acordo com um padrão de comunicação, como um protocolo de comunicação sem fio, e pode transmitir os dados de vídeo modulados para o dispositivo de destino 14. Uma ou mais mídias de comunicação pode incluir mídia de comunicação sem fio e/ou cabeada, como um espectro de radiofrequência (RF) ou uma ou mais linhas de transmissão física. Uma ou mais mídias de comunicação podem fazer parte de uma rede baseada em pacote, como uma rede de área local, uma rede de área remota, ou uma rede global (por exemplo, a Internet). Uma ou mais mídias de comunicação podem incluir roteadores, comutadores, estações base, ou outro equipamento que facilite comunicação a partir do dispositivo de fonte 12 para o dispositivo de destino 14.

[0040] Em outro exemplo, o canal 16 pode incluir uma mídia de armazenagem que armazena dados de vídeo codificados gerados pelo dispositivo de fonte 12. Nesse exemplo, o dispositivo de destino 14 pode acessar a mídia de armazenagem, por exemplo, através do acesso de disco ou acesso de cartão. A mídia de armazenagem pode incluir uma variedade de mídia de armazenagem de dados localmente acessados como discos Blu-ray, DVDs, CD-ROMs, memória flash, ou outra mídia de armazenagem digital adequada para armazenar dados de vídeo codificados.

[0041] Em um exemplo adicional, o canal 16 pode incluir um servidor de arquivo ou outro dispositivo de armazenagem intermediária que armazena dados de vídeo codificados gerados pelo dispositivo de fonte 12. Nesse exemplo, o dispositivo de destino 14 pode acessar dados de vídeo codificados armazenados no servidor de arquivo ou outro dispositivo de armazenagem intermediário através de streaming ou download. O servidor de arquivo pode ser um tipo de servidor capaz de armazenar dados de vídeo

codificados e transmitir os dados de vídeo codificados para o dispositivo de destino 14. Servidores de arquivo de exemplo incluem servidores de rede (por exemplo, para um website), servidores de protocolo de transferência de arquivo (FTP), dispositivos de armazenagem ligados em rede (NAS) e unidades de disco local.

[0042] O dispositivo de destino 14 pode acessar os dados de vídeo codificados através de uma conexão de dados padrão, como uma conexão da Internet. Tipos de exemplo de conexões de dados podem incluir canais sem fio (por exemplo, conexões Wi-Fi), conexões cabeadas (por exemplo, linha de assinante digital (DSL), modem de cabo, etc.), ou combinações de ambos que são adequadas para acessar dados de vídeo codificados armazenados em um servidor de arquivo. A transmissão de dados de vídeo codificados a partir do servidor de arquivo pode ser uma transmissão streaming, uma transmissão de download, ou uma combinação de ambos.

[0043] As técnicas dessa revelação não são limitadas a aplicações ou cenários sem fio. As técnicas podem ser aplicadas à codificação de vídeo em suporte de uma variedade de aplicações de multimídia, como broadcasts de televisão pelo ar, transmissões de televisão a cabo, transmissões de televisão por satélite, transmissões de vídeo streaming, por exemplo, através da Internet, codificação de dados de vídeo para armazenagem em uma mídia de armazenagem de dados, decodificação de dados de vídeo armazenados em uma mídia de armazenagem de dados, ou outras aplicações. Em alguns exemplos, o sistema de codificação de vídeo 10 pode ser configurado para suportar transmissão de vídeo simples ou duplex para suportar aplicações como streaming de vídeo, reprodução de vídeo, broadcasting de vídeo e/ou telefonia de vídeo.

[0044] A figura 1 é meramente um exemplo e as técnicas dessa revelação podem se aplicar a definições de codificação de vídeo (por exemplo, codificação de vídeo ou decodificação de vídeo) que não incluem necessariamente qualquer comunicação de dados entre os dispositivos de codificação e decodificação. Em outros exemplos, dados (por exemplo, dados de vídeo) são recuperados de uma memória local, streamed através de uma rede, ou similar. Um dispositivo de codificação de vídeo pode codificar e armazenar dados (por exemplo, dados de vídeo) para memória, e/ou um dispositivo de decodificação de vídeo pode recuperar e decodificar dados (por exemplo, dados de vídeo) a partir da memória. Em muitos exemplos, a codificação e decodificação é realizada por dispositivos que não se comunicam entre si, porém simplesmente codificam dados para memória e/ou recuperam e decodificam dados (por exemplo, dados de vídeo) a partir da memória.

[0045] No exemplo da figura 1, o dispositivo de fonte 12 inclui uma fonte de vídeo 18, um codificador de vídeo 20 e uma interface de saída 22. Em alguns exemplos, a interface de saída 22 pode incluir um modulador/demodulador (modem) e/ou um transmissor. A fonte de vídeo 18 pode incluir um dispositivo de captura de vídeo, por exemplo, uma câmera de vídeo, um arquivo de vídeo contendo dados de vídeo anteriormente capturados, uma interface de feed de vídeo para receber dados de vídeo a partir de um provedor de conteúdo de vídeo, e/ou um sistema de gráfico de computador para gerar dados de vídeo, ou uma combinação de tais fontes de dados de vídeo. Desse modo, em alguns exemplos, o dispositivo de fonte 12 compreende uma câmera configurada para capturar dados de vídeo.

[0046] O codificador de vídeo 20 pode codificar dados de vídeo a partir da fonte de vídeo 18. Em

alguns exemplos, o dispositivo de fonte 12 transmite diretamente os dados de vídeo codificados para o dispositivo de destino 14 através da interface de saída 22. Em outros exemplos, os dados de vídeo codificados podem ser também armazenados sobre uma mídia de armazenagem ou um servidor de arquivo para acesso posterior pelo dispositivo de destino 14 para decodificação e/ou reprodução.

[0047] No exemplo da figura 1, o dispositivo de destino 14 inclui uma interface de entrada 28, um decodificador de vídeo 30 e um dispositivo de exibição 32. Em alguns exemplos, a interface de entrada 28 inclui um receptor e/ou um modem. A interface de entrada 28 pode receber dados de vídeo através do canal 16. O decodificador de vídeo 30 pode decodificar dados de vídeo codificados. O dispositivo de exibição 32 pode exibir os dados de vídeo decodificados. O dispositivo de exibição 32 pode ser integrado com ou pode ser externo ao dispositivo de destino 14. O dispositivo de exibição 32 pode compreender uma variedade de dispositivos de exibição como um display de cristal líquido (LCD), um display de plasma, um display de diodo de emissão de luz orgânico (OLED), ou outro tipo de dispositivo de exibição.

[0048] O codificador de vídeo 20 e decodificador de vídeo 30 podem ser, individualmente, implementados como qualquer de uma variedade de conjunto de circuitos de codificador adequado, como um ou mais microprocessadores, processadores de sinais digitais (DSPs), circuitos integrados de aplicação específica (ASICs), disposições de porta programável em campo (FPGAs), lógica discreta, hardware, ou quaisquer combinações dos mesmos. Se as técnicas forem implementadas parcialmente em software, um dispositivo pode armazenar instruções para o software em uma mídia de armazenagem legível em computador,

não transitória, adequada e pode executar as instruções em hardware usando um ou mais processadores para executar as técnicas dessa revelação. Qualquer dos acima (incluindo hardware, software, uma combinação de hardware e software, etc.) pode ser considerada como um ou mais processadores. Cada do codificador de vídeo 20 e decodificador de vídeo 30 pode ser incluído em um ou mais codificadores ou decodificadores, qualquer um dos quais pode ser integrado como parte de um codificador/decodificador combinado (CODEC) em um dispositivo respectivo.

[0049] Essa revelação pode se referir em geral ao codificador de vídeo 20 "sinalizando" certas informações para outro dispositivo, como decodificador de vídeo 30. O termo "sinalização" pode se referir em geral à comunicação de elementos de sintaxe e/ou outros dados usados para decodificar os dados de vídeo comprimidos. Tal comunicação pode ocorrer em tempo real ou quase real. Alternativamente, tal comunicação pode ocorrer através de um período de tempo, como poderia ocorrer ao armazenar elementos de sintaxe para uma mídia de armazenagem legível em computador em um fluxo de bits codificado no momento de codificação, que pode ser então recuperado por um dispositivo de decodificação a qualquer momento após ser armazenado para essa mídia.

[0050] Em alguns exemplos, o codificador de vídeo 20 e o decodificador de vídeo 30 operam de acordo com um padrão de compressão de vídeo, como ISO/IEC MPEG-4 Visual e ITU-T H.264 (também conhecido como ISO/IEC MPEG-4 AVC), incluindo sua extensão de Codificação de vídeo escalonável (SVC), extensão de Codificação de vídeo de multivistas (MVC) e extensão 3DV baseada em MVC. Em algumas instâncias, qualquer fluxo de bits se conformando à extensão 3DV baseada em MVC de H.264/AVC sempre contém um

sub-fluxo de bits que está em conformidade com a extensão MVC de h.264/AVC. A minuta de junção mais recente de MVC é descrita em "Advanced video coding for generic audiovisual services," ITU-T Recommendation H.264, março de 2010. Além disso, há um esforço contínuo para gerar uma extensão de codificação de vídeo tridimensional (3DV) para H.264/AVC, a saber 3DV baseado em AVC. Em outros exemplos, o codificador de vídeo 2D e decodificador de vídeo 3D pode operar de acordo com ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 ou ISO/IEC MPEG-2 Visual, ITU-T H.263, e ISO/IEC MPEG-4 Visual.

[0051] Em outros exemplos, o codificador de vídeo 2D e o decodificador de vídeo 3D podem operar de acordo com o padrão de Codificação de vídeo de eficiência elevada (HEVC) desenvolvido pelo Joint Collaboration Team on Video coding (JCT-VC) de ITU-T Video Coding Experts Group (VCEG) e ISO/IEC Motion Pictures Experts Group (MPEG). Uma minuta do padrão HEVC, mencionado como a "especificação de minuta de HEVC" é descrita em Bross e outros, "High Efficiency video coding (HEVC) Defect Report 3," Joint Collaborative Team on video coding (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 16ª Reunião, San Jose, EUA, janeiro de 2014, documento no. JCTVC-P1003_v1. A especificação de minuta HEVC é disponível de http://phenix.it-sudparis.eu/jct/doc_end_user/documents/16_San%20Jose/wg11/JCTVC-P1003-v1.zip.

[0052] Além disso, há esforços contínuos para produzir uma extensão de codificação de vídeo escalonável para HEVC. A extensão de vídeo escalonável de HEVC pode ser mencionada como SHEVC ou SHVC. Adicionalmente, um Joint Collaboration Team on 3D video coding (JCT-3C) of VCEG and MPEG está desenvolvendo um padrão 3DV com base em HEVC.

Parte dos esforços de padronização para o padrão 3DV com base em HEVC inclui a padronização de um codec de vídeo de multivistas com base em HEVC (isto é, MV-HEVC). Outra parte dos esforços de padronização para o padrão 3DV com base em HEVC inclui a padronização e uma codificação de vídeo 3D com base em HEVC (isto é, 3D-HEVC). Para 3D-HEVC, novas ferramentas de codificação, incluindo aquelas na unidade de codificação (CU) ou nível de unidade de predição (PU), para vistas tanto de textura como de profundidade podem ser incluídas e suportadas.

[0053] Uma Minuta de trabalho recente (WD) de MV-HEVC, mencionada como MV-HEVC WD8 a seguir, é disponível a partir de http://phenix.it-sudparis.eu/jet2/doc_end_user/documents/8_Valencia/wg11/JCT3V-H1002-v1.zip. A extensão escalonável para HEVC, denominada SHVC, está também sendo desenvolvida pelo HCT-VC. Uma Minuta de trabalho recente (WD) de SHVC, mencionada como SHVC WD6 a seguir, é disponível de http://phenix.it-sudparis.eu/jet/doc_end_user/documents/17_Valencia/wg11/JCTVC-Q1008-v1.zip.

[0054] Em HEVC e outras especificações de codificação de vídeo, uma sequência de vídeo tipicamente inclui uma série de imagens. Imagens também podem ser mencionadas como "quadros". Uma imagem pode incluir três conjuntos de amostras, indicados S_L , S_{Cb} e S_{Cr} . S_L é um conjunto bidimensional (isto é, um bloco) de amostras de luma. S_{Cb} é um conjunto bidimensional de amostras de croma Cb. S_{Cr} é um conjunto bidimensional de amostras de croma Cr. Amostras de croma podem ser também mencionados aqui como amostras de "croma". Em outras instâncias, uma imagem pode ser monocromo e pode somente incluir um conjunto de amostras de luma.

[0055] Para gerar uma representação codificada

de uma imagem, o codificador de vídeo 20 pode gerar um conjunto de unidades de árvore de codificação (CTUs). Cada CTU pode compreender um bloco de árvore de codificação de amostras de luma, dois blocos de árvore de codificação correspondentes de amostras de croma, e estruturas de sintaxe usadas para codificar as amostras dos blocos de árvore de codificação. Em imagens de monocromo ou uma imagem que tenha três planos de cor separados, uma CTU pode compreender um bloco de árvore de codificação único e estruturas de sintaxe usadas para codificar as amostras do bloco de árvore de codificação. Um bloco de árvore de codificação pode ser um bloco NxN de amostras. Uma CTU pode ser também mencionada como um "bloco de árvore" ou uma "unidade de codificação maior" (LCU). As CTUs de HEVC podem ser amplamente análogas aos macroblocos de outros padrões, como H.264/AVC. Entretanto, uma CTU não é necessariamente limitada a um tamanho específico e pode incluir uma ou mais unidades de codificação (CUs). Uma fatia pode incluir um número inteiro de CTUs ordenadas consecutivamente em uma ordem de varredura.

[0056] Essa revelação pode usar o termo "unidade de vídeo" ou "bloco de vídeo" ou "bloco" para se referir a um ou mais blocos de amostras e estruturas de sintaxe usadas para codificar amostras de um ou mais blocos de amostras. Tipos de exemplo de unidades de vídeo podem incluir CTUs, CUs, PUs, unidades de transformada (TUs), macroblocos, divisões de macrobloco, e etc. Em alguns contextos, a discussão de PUs pode ser trocada com discussão de macroblocos ou divisões de macrobloco.

[0057] Para gerar uma CTU codificada, o codificador de vídeo 20 pode recursivamente executar divisão de quad-tree nos blocos de árvore de codificação de uma CTU para dividir os blocos de árvore de codificação nos

blocos de codificação, conseqüentemente o nome "unidades de árvore de codificação." Um bloco de codificação é um bloco NxN de amostras. Uma CU pode compreender um bloco de codificação de amostras de luma e dois blocos de codificação correspondentes de amostras de croma de um quadro que tem um conjunto de amostras de luma, um conjunto de amostra Cb e um conjunto de amostra Cr, e estruturas de sintaxe usadas para codificar as amostras dos blocos de codificação. Em imagens de monocromo ou imagens que tenham três planos de cores separadas, uma CU pode compreender um bloco de codificação único e estruturas de sintaxe usadas para codificar as amostras do bloco de codificação.

[0058] O codificador de vídeo 20 pode dividir um bloco de codificação de uma CU em um ou mais blocos de predição. Um bloco de predição é um bloco retangular (isto é, quadrado ou não quadrado) de amostras nas quais a mesma predição é aplicada. Uma unidade de predição (PU) de uma CU pode compreender um bloco de predição de amostras de luma, dois blocos de predição correspondentes de amostras de croma, e estruturas de sintaxe usadas para prever os blocos de predição. Em imagens de monocromo ou imagens que tenham três planos de cores separadas, uma PU pode compreender um bloco de predição único e estruturas de sintaxe usadas para prever as amostras de bloco de predição. O codificador de vídeo 20 pode gerar blocos preditivos, (por exemplo, blocos preditivos de luma, Cb e Cr) para blocos de predição (por exemplo, blocos de predição de luma, Cb e Cr) de cada PU da CU.

[0059] O codificador de vídeo 20 pode usar intra predição ou inter predição para gerar os blocos preditivos para uma PU. Se o codificador de vídeo 20 usar intra predição para gerar os blocos preditivos de uma PU, o codificador de vídeo 20 pode gerar os blocos preditivos da

PU com base em amostras decodificadas da imagem que inclui a PU.

[0060] Após o codificador de vídeo 20 gerar blocos preditivos (por exemplo, blocos preditivos de luma, Cb e Cr) para uma ou mais PUs de uma CU, o codificador de vídeo 20 pode gerar um ou mais blocos residuais para a CU. Por exemplo, o codificador de vídeo 20 pode gerar um bloco residual de luma para a CU. Cada amostra no bloco residual de luma da CU indica uma diferença entre uma amostra de luma em um dos blocos de luma preditivos da CU e uma amostra correspondente no bloco de codificação de luma original da CU. Além disso, o codificador de vídeo 20 pode gerar um bloco residual de Cb para a CU. Cada amostra no bloco residual de Cb da CU pode indicar uma diferença entre uma amostra de Cb em um dos blocos de Cb preditivos da CU e uma amostra correspondente no bloco de codificação de Cb original da CU. O codificador de vídeo 20 pode também gerar um bloco residual de Cr para a CU. Cada amostra no bloco residual de Cr da CU pode indicar uma diferença entre uma amostra de Cr em um dos blocos de Cr preditivos da CU e uma amostra correspondente no bloco de codificação de Cr original da CU.

[0061] Além disso, o codificador de vídeo 20 pode usar divisão de quad-tree para decompor os blocos residuais (por exemplo, os blocos residuais de luma, Cb e Cr) de uma CU em um ou mais blocos de transformada (por exemplo, blocos de transformada de luma, Cb e Cr). Um bloco de transformada é um bloco retangular (por exemplo, quadrado ou não quadrado) de amostras no qual a mesma transformada é aplicada. Uma unidade de transformada (TU) de uma CU pode compreender um bloco de transformada de amostras de luma, dois blocos de transformada correspondentes de amostras de croma, e estruturas de

sintaxe usadas para transformar as amostras de bloco de transformada. Desse modo, cada TU de uma CU pode ter um bloco de transformada de luma, um bloco de transformada de Cb, e um bloco de transformada de Cr. O bloco de transformada de luma da TU pode ser um sub-bloco do bloco residual de luma da CU. O bloco de transformada de Cb pode ser um sub-bloco do bloco residual de Cb da CU. O bloco de transformada de Cr pode ser um sub-bloco do bloco residual de Cr da CU. Em imagens de monocromo ou imagens tendo três planos de cor separados, uma TU pode compreender um único bloco de transformada e estruturas de sintaxe usadas para transformadas as amostras do bloco de transformada.

[0062] O codificador de vídeo 20 pode aplicar uma ou mais transformadas em um bloco de transformada de uma TU para gerar um bloco de coeficiente para a TU. Por exemplo, o codificador de vídeo 20 pode aplicar uma ou mais transformadas a um bloco de transformada de luma de uma TU para gerar um bloco de coeficiente de luma para a TU. Um bloco de coeficiente pode ser um conjunto bidimensional de coeficientes de transformada. Um coeficiente de transformada pode ser uma quantidade escalar. O codificador de vídeo 20 pode aplicar uma ou mais transformadas em um bloco de transformada de Cb de uma TU para gerar um bloco de coeficiente de Cb para a TU. O codificador de vídeo 20 pode aplicar uma ou mais transformadas em um bloco de transformada de Cr de uma TU para gerar um bloco de coeficiente de Cr para a TU.

[0063] Após gerar um bloco de coeficiente (por exemplo, um bloco de coeficiente de luma, um bloco de coeficiente de Cb ou um bloco de coeficiente de Cr), o codificador de vídeo 20 pode quantizar o bloco de coeficiente. A quantização se refere em geral a um processo no qual coeficientes de transformada são quantizados para

possivelmente reduzir a quantidade de dados usados para representar os coeficientes de transformada, fornecendo compressão adicional. Após o codificador de vídeo 20 quantizar um bloco de coeficiente, o codificador de vídeo 20 pode codificar por entropia elementos de sintaxe indicando os coeficientes de transformada quantizados. Por exemplo, o codificador de vídeo 20 pode executar Codificação aritmética binária adaptável ao contexto (CABAC) nos elementos de sintaxe indicando os coeficientes de transformada quantizados.

[0064] O codificador de vídeo 20 pode transmitir um fluxo de bits que inclui uma sequência de bits que forma uma representação de quadros codificados e dados associados. Desse modo, o fluxo de bits compreende uma representação codificada de dados de vídeo. O fluxo de bits pode compreender uma sequência de unidades de camada de abstração de rede (NAL). Uma unidade NAL é uma estrutura de sintaxe contendo uma indicação do tipo de dados na unidade NAL e bytes contendo aqueles dados na forma de uma carga útil de sequência de bytes brutos (RBSP) intercalados conforme necessário com bits de prevenção de emulação. Cada das unidades NAL inclui um cabeçalho de unidade NAL e encapsula uma RBSP. O cabeçalho de unidade NAL pode incluir um elemento de sintaxe que indica um código do tipo de unidade NAL. O código do tipo de unidade NAL especificado pelo cabeçalho de unidade NAL de uma unidade NAL indica o tipo da unidade NAL. Um RBSP pode ser uma estrutura de sintaxe contendo um número inteiro de bytes que é encapsulado em uma unidade NAL. Em algumas instâncias, um RBSP inclui zero bits.

[0065] Tipos diferentes de unidades NAL podem encapsular tipos diferentes de RBSPs. Por exemplo, tipos diferentes de unidades NAL podem encapsular RBSPs

diferentes para conjuntos de parâmetro de vídeo (VPSs), conjuntos de parâmetro de sequência (SPSs), conjuntos de parâmetro de quadro (PPSs), fatias codificadas, Informações de Aperfeiçoamento suplementares (SEI) e etc. Unidades NAL que encapsulam RBSPs para dados de codificação de vídeo (ao contrário de RBSPs para conjuntos de parâmetros e mensagens SEI) podem ser mencionadas como unidades NAL de camada de codificação de vídeo (VCL). Em HEVC (isto é, HEVC não de multicamadas), uma unidade de acesso pode ser um conjunto de unidades NAL que são consecutivas em ordem de decodificação e contêm exatamente uma imagem codificada. Além das unidades de NAL de fatia codificada da imagem codificada, a unidade de acesso pode conter também outras unidades NAL não contendo fatias da imagem codificada. Em alguns exemplos, a decodificação de uma unidade de acesso sempre resulta em uma imagem decodificada. Informação de aperfeiçoamento suplementar (SEI) contém informações que não são necessárias para decodificar as amostras de imagens codificadas a partir das unidades NAL VCL. Um RBSP de SEI contém uma ou mais mensagens SEI.

[0066] O decodificador de vídeo 30 pode receber um fluxo de bits gerado pelo codificador de vídeo 20. Além disso, o decodificador de vídeo 30 pode analisar o fluxo de bits para obter elementos de sintaxe a partir do fluxo de bits. O decodificador de vídeo 30 pode reconstruir os quadros dos dados de vídeo com base pelo menos em parte nos elementos de sintaxe obtidos a partir do fluxo de bits. O processo para reconstruir os dados de vídeo pode ser genericamente recíproco ao processo executado pelo codificador de vídeo 20. Por exemplo, o decodificador de vídeo 30 pode usar vetores de movimento de PUs para determinar blocos preditivos para as PUs de uma CU atual. Além disso, o decodificador de vídeo 30 pode quantizar

inverso blocos de coeficiente associados a TUs da CU atual. O decodificador de vídeo 30 pode executar transformadas inversas nos blocos de coeficiente para reconstruir blocos de transformada associados às TUs da CU atual. O decodificador de vídeo 30 pode reconstruir os blocos de codificação da CU atual por adicionar as amostras dos blocos de amostra preditivos para PUs da CU atual a amostras correspondentes dos blocos de transformada das TUs da CU atual. Por reconstruir os blocos de codificação para cada CU de um quadro, o decodificador de vídeo 30 pode reconstruir o quadro.

[0067] As técnicas dessa revelação são potencialmente aplicáveis a padrões de codificação de multivista e/ou 3DV e especificações, incluindo MV-HEVC e 3D-HEVC. Em codificação de multivista, como aquela definida em MV-HEVC e 3D-HEVC, pode haver múltiplas vistas da mesma cena a partir de pontos de vista diferentes. No contexto de codificação de multivista e codificação de vídeo escalonável, o termo “unidade de acesso” pode ser usado para se referir ao conjunto de imagens que corresponde à mesma instância de tempo. Em algumas instâncias, no contexto de codificação de multivista e codificação de vídeo escalonável, uma unidade de acesso pode compreender um conjunto de unidades NAL que são associadas entre si de acordo com uma regra de classificação especificada, são consecutivas em ordem de decodificação, e contêm as unidades NAL VCL de todas as imagens codificadas associadas ao mesmo tempo de saída e suas unidades NAL não VCL associadas. Desse modo, dados de vídeo podem ser conceptualizados como uma série de unidades de acesso que ocorrem ao longo do tempo.

[0068] Em codificação 3DV, como aquela definida em 3D-HEVC, um “componente de vista” pode ser uma

representação codificada de uma vista em uma unidade de acesso única. Um componente de vista pode conter um componente de vista de profundidade e um componente de vista de textura. Um componente de vista de profundidade pode ser uma representação codificada da profundidade de uma vista em uma unidade de acesso único. Um componente de vista de textura pode ser uma representação codificada da textura de uma vista em uma unidade de acesso único. Nessa revelação, uma "vista" pode se referir a uma sequência de componentes de vista associados ao mesmo identificador de vista.

[0069] O componente de vista de textura e o componente de vistas de profundidade em um conjunto de imagens de uma vista podem ser considerados como correspondendo entre si. Por exemplo, o componente de vista de textura em um conjunto de imagens de uma vista é considerado como correspondendo ao componente de vista de profundidade no conjunto das imagens da vista, e vice-versa (isto é, o componente de vista de profundidade corresponde a seu componente de vista de textura no conjunto e vice-versa). Como usado nessa revelação, um componente de vista de textura que corresponde a um componente de vista de profundidade pode ser considerado como o componente de vista de textura e o componente de vista de profundidade sendo parte de uma mesma vista de uma unidade de acesso único.

[0070] O componente de vista de textura inclui o conteúdo de imagem efetiva que é exibido. Por exemplo, o componente de vista de textura pode incluir componentes luma (Y) e croma (Cb e Cr). O componente de vista de profundidade pode indicar profundidades relativas dos pixels em seu componente de vista de textura correspondente. como exemplo, o componente de vista de

profundidade pode compreender uma imagem em escala cinza que inclui apenas valores de luma, porém os valores de luma em um componente de vista de profundidade podem representar um nível de profundidade de um pixel ou conjunto de pixels, ao contrário de um nível de intensidade de pixel que seria transmitido por um componente de luma de uma vista de textura. Em outras palavras, o componente de vista de profundidade pode não transmitir um conteúdo de imagem, porém ao invés fornecer uma medição das profundidades relativas dos pixels no componente de vista de textura. O componente de vista de profundidade usando somente valores de luma (por exemplo, valores de intensidade) para identificar profundidade é fornecido para fins de ilustração e não deve ser considerado limitador. Em outros exemplos, qualquer técnica pode ser utilizada para indicar profundidades relativas dos pixels no componente de vista de textura.

[0071] Em codificação de multivistas e codificação SVC, um fluxo de bits pode ter uma pluralidade de camadas. Em codificação de multivistas, cada das camadas pode corresponder a uma vista diferente. Uma camada pode ser mencionada como uma "camada base" se um decodificador de vídeo (por exemplo, decodificador de vídeo 30) pode decodificar imagens na camada sem referência a imagens em qualquer outra camada. Uma camada pode ser mencionada como uma "camada não base" ou uma "camada dependente" se a decodificação da camada for dependente da decodificação de imagens em uma ou mais outras camadas. Ao codificar uma imagem em uma das camadas não base, um codificador de vídeo (como codificador de vídeo 20 ou decodificador de vídeo 30) pode adicionar uma imagem em uma lista de imagens de referência se a imagem estiver em uma vista diferente, porém em uma mesma instância de tempo (isto é, unidade de

acesso) que a imagem que o codificador de vídeo está atualmente codificando. Como outras imagens de referência inter predição, o codificador de vídeo pode inserir uma imagem de referência de predição intervista em qualquer posição de uma lista de imagem de referência.

[0072] Em SVC, camadas diferentes da camada de base podem ser mencionadas como "camadas de aperfeiçoamento" e podem fornecer informações que aperfeiçoam a qualidade visual de dados de vídeo decodificados a partir do fluxo de bits. SVC pode aumentar a resolução espacial, relação de sinal para ruído (isto é, qualidade) ou taxa temporal. Em codificação de vídeo escalonável (por exemplo, SHVC), uma "representação de camada" pode ser uma representação codificada de uma camada espacial em uma unidade de acesso único. Para facilidade de explicação, essa revelação pode se referir a componentes de vista e/ou representações de camada como "componentes de vista/representações de camada" ou "simplesmente "imagens."

[0073] Para implementar as camadas, cabeçalhos de unidades NAL podem incluir elementos de sintaxe `nuh_reserved_zero_6bits` que também podem ser mencionados como elementos de sintaxe `"nuh_layer_id"`. Unidades NAL que têm elementos de sintaxe `nuh_reserved_zero_6bits` que especificam valores diferentes pertencem a camadas diferentes de um fluxo de bits. Desse modo, em codificação de multivista (por exemplo, MV-HEVC), 3DV (por exemplo, 3D-HEVC), ou SVC (por exemplo, SHVC), o elemento de sintaxe `nuh_reserved_zero_6bits` de uma unidade NAL especifica um identificador de camada (isto é, um ID de camada) da unidade NAL. Em alguns exemplos, o elemento de sintaxe `nuh_reserved_zero_6bits` de uma unidade NAL é igual a 0 se a unidade NAL se referir a uma camada base em codificação de multivista (por exemplo, MV-HEVC), codificação 3DV (por

exemplo, 3D-HEVC), ou SVC (por exemplo, SHVC). Dados em uma camada de base de um fluxo de bits podem ser decodificados sem referência a dados em qualquer outra camada do fluxo de bits. Se a unidade NAL não se referir a uma camada base em codificação de multivista, 3DV, ou SVC, o elemento de sintaxe `nuh_reserved_zero_6bits` do elemento de sintaxe pode ter um valor não zero. Como indicado acima, em codificação de multivista e codificação 3DV, camadas diferentes de um fluxo de bits podem corresponder a vistas diferentes.

[0074] Além disso, alguns componentes de vista/representações de camada em uma camada podem ser decodificadas sem referência a outros componentes de vista/representações de camada na mesma camada. Desse modo, unidades NAL encapsulando dados de certos componentes de vista/representações de camada de uma camada podem ser removidos do fluxo de bits sem afetar a capacidade de decodificação de outros componentes de vista/representações de camada na camada. A remoção de unidades NAL encapsulando dados de tais componentes de vista/representações de camada pode reduzir a taxa de quadro do fluxo de bits. Um subconjunto de componentes de vista/representações de camada em uma camada que pode ser decodificada sem referência a outros componentes de vista/representações de camada na camada pode ser mencionado aqui como uma "subcamada" ou uma "subcamada temporal."

[0075] Unidades NAL podem incluir elementos de sintaxe `temporal_id` que especificam identificadores temporais (isto é, `TemporalIds`) das unidades NAL. O identificador temporal de uma unidade NAL identifica uma subcamada a qual a unidade NAL pertence. Desse modo, cada subcamada de um fluxo de bits pode ter um identificador temporal diferente. Em geral, se o identificador temporal de uma primeira unidade NAL for menor que o identificador

temporal de uma segunda unidade NAL, os dados encapsulados pela primeira unidade NAL podem ser decodificados sem referência aos dados encapsulados pela segunda unidade NAL.

[0076] Como brevemente indicado acima, unidades NAL podem encapsular RBSP para conjuntos de parâmetro de vídeo (VPSs), conjuntos de parâmetro de sequência (SPSs), conjuntos de parâmetro de imagem (PPSs). Um VPS é uma estrutura de sintaxe compreendendo elementos de sintaxe que se aplicam a zero ou mais sequências de vídeo codificadas inteiras (CVSs). Um SPS também é uma estrutura de sintaxe compreendendo elementos de sintaxe que se aplicam a zero ou mais CVSs inteiros. Um SPS pode incluir um elemento de sintaxe que identifica um VPS que é ativo quando o SPS está ativo. Desse modo, os elementos de sintaxe de um VPS podem ser mais genericamente aplicáveis do que os elementos de sintaxe de um SPS. Um PPS é uma estrutura de sintaxe compreendendo elementos de sintaxe que se aplicam a zero ou mais imagens codificadas. Um PPS pode incluir um elemento de sintaxe que identifica um SPS que é ativo quando o PPS está ativo. Um cabeçalho de fatia de uma fatia pode incluir um elemento de sintaxe que indica um PPS que está ativo quando a fatia está sendo codificada.

[0077] Conjuntos de parâmetro, como VPSs, SPSs e PPSs podem incluir estruturas de sintaxe de extensão específica para extensões para o padrão HEVC. Por exemplo, conjuntos de parâmetro podem incluir extensões específicas para SHVC, MV-HEVC, 3D-HEVC e etc. As estruturas de sintaxe de extensão podem incluir elementos de sintaxe aplicáveis a extensões ao padrão HEVC. A tabela 1, abaixo, mostra uma porção de extensão VPS definida em SHVC WD6 e MV-HEVC WD8.

Tabela 1

vps_extension() {	Descriptor
avc_base_layer_flag	u(1)
...	u(1)
vps_num_profile_tier_level_minus1	ue(v)
for(ptIdx = 1; ptIdx <= vps_num_profile_tier_level_minus1; ptIdx ++)	
{	
vps_profile_present_flag[ptIdx]	u(1)
profile_tier_level(vps_profile_present_flag[ptIdx],	
vps_max_sub_layers_minus1)	
}	
...	
for(i = 1; i < NumOutputLayerSets; i ++) {	
if(i > vps_num_layer_sets_minus1)	
output_layer_set_idx_minus1[i]	u(v)
if(i > vps_num_layer_sets_minus1 defaultTargetOutputLayerIdx ==	
2)	
for(j = 0; j < NumLayersInIdList[LayerSetIdxForOutputLayerSet[i]];	
j ++)	
output_layer_flag[i][j]	u(1)
profile_level_tier_idx[i]	u(v)

if(NumOutputLayersInOutputLayerSet[i] == 1	
&& NumDirectRefLayers[OlsHighestOutputLayerId[i]] > 0)	
alt_output_layer_flag[i]	u(1)
}	
...	
vps_vui_present_flag	
if(vps_vui_present_flag) {	
while(!byte_aligned())	
vps_vui_alignment_bit_equal_to_one	u(1)
vps_vui()	
}	
}	

[0078] No exemplo da tabela 1, acima, e outras tabelas de sintaxe dessa revelação, elementos de sintaxe com descritor de tipo ue(v) podem ser inteiros não

sinalizados de comprimento variável codificados usando 0° codificação Golomb exponencial (Exp-Golomb) com bit esquerdo primeiro. No exemplo da tabela 1 e nas seguintes tabelas, elementos de sintaxe tendo descritores da forma $u(n)$, onde n é um inteiro não negativo, são valores não sinalizados de comprimento n .

[0079] A porção da extensão VPS definida na Tabela 1 inclui elementos de sintaxe definindo um ou mais conjuntos de camada de saída. Um conjunto de camada de saída é um conjunto de camadas consistindo nas camadas de um dos conjuntos de camada especificados no VPS, onde uma ou mais camadas no conjunto de camadas são indicados para serem camadas de saída. Como definido em SHVC WD6, uma camada de saída é uma camada de um conjunto de camadas de saída que é transmitido quando TargetOptLayerSetIdx é igual ao índice do conjunto de camadas de saída. TargetOptLayerSetIdx especifica uma lista de identificador de camada de um ponto de operação em uso e um conjunto de camadas de saída alvo. Um ponto de operação pode se referir a um subconjunto (Por exemplo, um subconjunto adequado) de unidades NAL no fluxo de bits. Além disso, como definido em SHVC WD6, uma camada de saída alvo é uma camada que deve ser transmitida e é uma das camadas de saída do conjunto de camadas de saída com índice olsIdx de modo que TargetOptLayerSetIdx seja igual a olsIdx. Desse modo, em geral, o conjunto de camadas de saída alvo pode ser um específico dos conjuntos de camada de saída. Cada camada do conjunto de camadas de saída alvo é transmitida (por exemplo, fornecida como saída de um decodificador de vídeo). Em algumas instâncias, o decodificador de vídeo 30 pode decodificar uma camada específica de um fluxo de bits, porém não transmite dados de vídeo decodificados de saída da camada específica. Ao invés, o decodificador de vídeo 30

pode usar os dados de vídeo decodificados da camada específica para decodificar dados de vídeo de camadas que estão no conjunto de camada de saída alvo.

[0080] Além disso, a porção da extensão VPS definida na Tabela 1 inclui elementos de sintaxe definindo uma ou mais estruturas de sintaxe de perfil, plano e nível (PTL). Cada estrutura de sintaxe PTL especifica um perfil, plano e nível para cada sequência de vídeo codificado. Como mostrado adicionalmente na Tabela 1, a extensão VPS inclui elementos de sintaxe `profile_level_tier_idx` que identificam estruturas de sintaxe PTL aplicáveis a conjuntos de camada de saída respectivos. Desse modo, a extensão VPS especifica informação PTL para cada dos conjuntos de camada de saída.

[0081] Em geral, um “perfil” é um subconjunto das características e ferramentas presentes em um padrão de codificação de vídeo (por exemplo, HEVC ou uma extensão do mesmo). Compreendido nos limites impostos por um perfil dado é ainda possível exigir uma variação muito grande no desempenho de codificadores e decodificadores dependendo dos valores assumidos por elementos de sintaxe no fluxo de bits como o tamanho especificado das imagens decodificadas. Em muitas aplicações, pode ser não prático nem econômico implementar um decodificador capaz de lidar com todos os usos hipotéticos da sintaxe em um perfil específico. Para lidar com esse problema, “planos” e “níveis” são especificados em cada perfil.

[0082] Em alguns exemplos, um nível é um conjunto definido de limitações sobre os valores que podem ser assumidos pelos elementos de sintaxe e variáveis de um padrão de codificação de vídeo (por exemplo, HEVC ou uma extensão do mesmo). Além disso, em alguns exemplos, um plano é uma categoria especificada de limitações de nível impostas sobre valores dos elementos de sintaxe no fluxo de

bits ou valores de variáveis, onde as limitações de nível são encaixadas em um plano e um decodificador se conformando a um certo plano e nível seria capaz de decodificar todos os fluxos de bits que se conformam ao mesmo plano ou ao plano inferior daquele nível ou qualquer nível abaixo do mesmo. Desse modo, um nível de um plano é um conjunto especificado de limitações impostas sobre valores dos elementos de sintaxe no fluxo de bits ou valores de variáveis. Essas limitações podem ser limites simples sobre valores. Alternativamente, podem ter a forma de limitações em combinações aritméticas de valores (por exemplo, largura de imagem multiplicada por altura de imagem multiplicada pelo número de imagens decodificadas por segundo). Um nível especificado para um plano inferior é mais limitado do que um nível especificado para um plano mais elevado. Em alguns contextos, o termo “nível” pode ser também usado para se referir ao valor de um coeficiente de transformada antes do aumento.

[0083] Como adicionalmente mostrado na tabela 1, acima, a extensão VPS pode incluir uma estrutura de sintaxe `vps_vui()`. A estrutura de sintaxe `vps_vui()` contém parâmetros de informação de capacidade de uso de vídeo (VUI). Parágrafo F.7.3.2.1.4 de SHVC WD6 descreve a sintaxe de uma estrutura de sintaxe `vps_vui()`. Como especificado na especificação de minuta HEVC, parâmetros VUI não são exigidos para construir as amostras de luma ou croma pelo processo de decodificação. Decodificadores de conformação não são exigidos processar essa informação para conformidade de ordem de transmissão para essa Especificação (vide anexo C para a especificação de conformidade de ordem de transmissão). Alguns parâmetros VUI são exigidos para checar conformidade de fluxo de bits e para conformidade de decodificador de temporização de

transmissão. em algumas instâncias, parâmetros VUI podem ser também especificados em SPSs.

[0084] Uma estrutura de sintaxe `vps_vui()` pode incluir uma estrutura de sintaxe `vps_vui_bsp_hrd_parameters()`. A estrutura de sintaxe `vps_vui_bsp_hrd_parameters()` inclui elementos de sintaxe que especificam parâmetros de decodificador de referência hipotética (HRD) para divisões de fluxo de bits. Como definido em SHVC WD6, uma divisão de fluxo de bits é uma sequência de bits, na forma de um fluxo de unidade NAL ou um fluxo de bytes, que é um subconjunto de um fluxo de bits de acordo com uma divisão.

[0085] Pelo menos alguns padrões de codificação de vídeo especificam modelos de armazenagem de vídeo. Em H.264/AVC e HEVC, um modelo de armazenagem é mencionado como um "decodificador de referência hipotético" ou "HRD." Na especificação de minuta HEVC, o HRD é descrito no Anexo C. O HRD descreve como dados devem ser armazenados para decodificação e como dados decodificados são armazenados para transmissão. por exemplo, o HRD descreve a operação de um buffer de imagem codificada ("CPB"), um buffer de imagem decodificada ("DPB"), e um processo de decodificação de vídeo. O CPB é um buffer primeiro a entrar primeiro a sair contendo unidades de acesso em uma ordem de decodificação especificada pelo HRD. O DPB é um buffer contendo imagens decodificadas para referência reordenação de saída ou atraso de saída especificado pelo HRD. Os comportamentos do CPB e DPB podem ser matematicamente especificados. O HRD pode impor diretamente limitações sobre temporização, tamanhos de buffer e taxas de bit. Além disso, o HRD pode indiretamente impor limitações sobre várias características e estatísticas de fluxo de bits.

[0086] Em H.264/AVC e HEVC, conformidade de fluxo de bits e conformidade de decodificador são especificadas como partes da especificação de HRD. Em outras palavras, o modelo de HRD especifica testes para determinar se um fluxo de bits se conformar a um padrão e testes para determinar se um decodificador se conforma ao padrão. Embora o HRD seja citado como algum tipo de decodificador, codificadores de vídeo usam tipicamente o HRD para assegurar conformidade de fluxo de bits, enquanto decodificadores de vídeo tipicamente não necessitam do HRD.

[0087] H.264/AVC e HEVC especificam, ambos, dois tipos de fluxo de bits ou conformidade de HRD, a saber Tipo I e Tipo II. Um fluxo de bits do tipo I é um fluxo de unidade NAL contendo somente as unidades NAL VCL e unidade NAL de dados de enchimento para todas as unidades de acesso no fluxo de bits. Pelo menos em alguns exemplos, um fluxo de unidade NAL é uma sequência de unidades NAL. Um fluxo de bits do tipo II é um fluxo de unidade NAL que contém, além das unidades NAL VCL e unidades NAL de dados de enchimento para todas as unidades de acesso no fluxo de bits, pelo menos um dos seguintes: unidades NAL não VCL adicionais diferentes de unidades NAL de dados de enchimento; e todos os elementos de sintaxe `leading_zero_8bits`, `zero_byte`, `start_coded_prefix_one_3bytes`, e `trailing_zero_8bits` que formam um fluxo de bytes a partir do fluxo de unidade NAL.

[0088] Quando um dispositivo executa um teste de conformidade de fluxo de bits que determina se um fluxo de bits se conforma a um padrão de codificação de vídeo, o dispositivo pode selecionar um ponto de operação do fluxo de bits. O dispositivo pode então determinar um conjunto de parâmetros HRD aplicáveis ao ponto de operação selecionado. O dispositivo pode usar o conjunto de parâmetros HRD aplicáveis ao ponto de operação selecionado para configurar

o comportamento do HRD. Mais particularmente, o dispositivo pode usar o conjunto aplicável de parâmetros HRD para configurar os comportamentos de componentes particulares do HRD, como um programador de fluxo hipotético (HSS), o CPB, um processo de decodificação, o DPB, e etc. Subsequentemente, o HSS pode injetar dados de vídeo codificados do fluxo de bits no CPB do HRD de acordo com um programa específico. Além disso, o dispositivo pode invocar um processo de decodificação que decodifica os dados de vídeo codificados no CPB. O processo de decodificação pode transmitir imagens decodificadas para o DPB. À medida que o dispositivo move dados através do HRD, o dispositivo pode determinar se um conjunto específico de limitações permanece atendido. por exemplo, o dispositivo pode determinar se uma condição de transbordamento ou subfluxo ocorre no CPB ou DPB enquanto o HRD está decodificando o ponto de operação do ponto de operação selecionado. O dispositivo pode selecionar e processar cada ponto de operação do fluxo de bits desse modo. Se nenhum ponto de operação do fluxo de bits fizer com que as limitações sejam violadas, o dispositivo pode determinar que o fluxo de bits se conforme ao padrão de codificação de vídeo.

[0089] Tanto H.264/AVC como HEVC especificam dois tipos de conformidade de decodificador, a saber conformidade de decodificador de temporização de transmissão e conformidade de decodificador de ordem de transmissão. Um decodificador reivindicando conformidade a um perfil, plano e nível específico é capaz de decodificar com sucesso todos os fluxos de bits que se conformam às exigências de conformidade de fluxo de bits de um padrão de codificação de vídeo, como HEVC.

[0090] Quando um dispositivo executa um teste de conformidade de decodificador para determinar se um

decodificador em teste (DUT) se conforma a um padrão de codificação de vídeo, o dispositivo pode fornecer, tanto ao HRD como ao DUT, um fluxo de bits que se conforma ao padrão de codificação de vídeo. O HRD pode processar o fluxo de bits no modo descrito acima com relação ao teste de conformidade de fluxo de bits. O dispositivo pode determinar que o DUT se conforma ao padrão de codificação de vídeo se a ordem de imagens decodificadas transmitidas pelo DUT casa com a ordem de imagens decodificadas transmitidas pelo HRD. Além disso, o dispositivo pode determinar que o DUT se conforma ao padrão de codificação de vídeo se a temporização com a qual o DUT transmite imagens decodificadas casa com a temporização com a qual o HRD transmite as imagens decodificadas.

[0091] Nos modelos H.264/AVC e HEVC HRD, a decodificação ou remoção de CPB pode ser baseada em unidade de acesso. Isto é, o HRD é assumido como decodificando unidades de acesso completas em um momento e remover unidades de acesso completas a partir do CPB. Além disso, nos modelos H.264/AVC e HEVC HRD, é assumido que decodificação de imagem é instantânea. O codificador de vídeo 20 pode sinalizar, em mensagens SEI de temporização de imagem, tempos de decodificação para indicar a decodificação de unidades de acesso. Em aplicações práticas, se um decodificador de vídeo em conformação seguir rigorosamente os tempos de decodificação sinalizados para iniciar a decodificação de unidades de acesso, o tempo mais cedo possível para transmitir uma imagem decodificada específica é igual ao tempo de decodificação daquela imagem específica mais o tempo necessário para decodificar aquela imagem específica. Entretanto, no mundo real, o tempo necessário para decodificar uma imagem não pode ser igual a zero.

[0092] Parâmetros HRD podem controlar vários aspectos do HRD. Em outras palavras, o HRD pode se basear nos parâmetros HRD. Os parâmetros HRD podem incluir um atraso de remoção de CPB inicial, um tamanho CPB, uma taxa de bits, um atraso de saída de DPB inicial e um tamanho de DPB. O codificador de vídeo 20 pode sinalizar esses parâmetros de HRD em uma estrutura de sintaxe `hrd_parameters()` especificada em um conjunto de parâmetros de vídeo (VPS) e/ou um conjunto de parâmetros de sequência (SPS). VPSs e/ou SPSs individuais podem incluir múltiplas estruturas de sintaxe `hrd_parameters()` para conjuntos diferentes de parâmetros HRD. Em alguns exemplos, o codificador de vídeo 20 pode sinalizar parâmetros HRD em mensagens SEI de período de armazenagem ou mensagens SEI de temporização de imagem.

[0093] Um ponto de operação de um fluxo de bits é associado a um conjunto de identificadores de camada (isto é, um conjunto de valores `nuh_reserved_zero_6bits`, que são também conhecidos como valores `nuh_layer_id`) e um identificador temporal. Um ponto de operação pode incluir cada unidade NAL que é associada ao ponto de operação. Um ponto de operação pode ter uma taxa de quadro e/ou uma taxa de bits diferente de um fluxo de bit original. Isso é porque o ponto de operação pode não incluir algumas imagens e/ou alguns dos dados do fluxo de bits original. Consequentemente, se o decodificador de vídeo 30 fosse para remover dados a partir do CPB e/ou DPB em uma taxa específica ao processar o fluxo de bits original e se o decodificador de vídeo 30 fosse para remover dados a partir do CPB e/ou DPB na mesma taxa ao processar um ponto de operação, o decodificador de vídeo 30 pode remover dados em demasia ou muito poucos dados a partir do CPB e/ou DPB. Por conseguinte, o codificador de vídeo 20 pode sinalizar

conjuntos diferentes de parâmetros HRD para pontos de operação diferentes. Por exemplo, o codificador de vídeo 20 pode incluir, em um VPS, múltiplas estruturas de sintaxe `hrd_parameters()` que incluem parâmetros HRD para pontos de operação diferentes.

[0094] A tabela 2, abaixo é uma sintaxe de exemplo para uma estrutura de sintaxe `hrd_parameters()` em HEVC.

Tabela 2 - Parâmetros HRD

<code>hrd_parameters(commonInfPresentFlag,</code> <code>MaxNumSubLayersMinus1) {</code>	Descritor
<code>if(commonInfPresentFlag) {</code>	
timing_info_present_flag	u(1)
<code>if(timing_info_present_flag) {</code>	
num_units_in_tick	u(32)
time_scale	u(32)
<code>}</code>	
nal_hrd_parameters_present_flag	u(1)
vcl_hrd_parameters_present_flag	u(1)
<code>if(nal_hrd_parameters_present_flag </code> <code>vcl_hrd_parameters_present_flag){</code>	
sub_pic_cpb_params_present_flag	u(1)
<code>if(sub_pic_cpb_params_present_flag) {</code>	
tick_divisor_minus2	u(8)
du_cpb_removal_delay_length_minus1	u(5)
<code>}</code>	
bit_rate_scale	u(4)
cpb_size_scale	u(4)

initial_cpb_removal_delay_length_minus1	u(5)
cpb_removal_delay_length_minus1	u(5)
dpb_output_delay_length_minus1	u(5)
}	
}	
for(i = 0; i <= MaxNumSubLayersMinus1; i++) {	
fixed_pic_rate_flag[i]	u(1)
if(fixed_pic_rate_flag[i])	
pic_duration_in_tc_minus1[i]	ue(v)
low_delay_hrd_flag[i]	u(1)
cpb_cnt_minus1[i]	ue(v)
if(nal_hrd_parameters_present_flag)	
sub_layer_hrd_parameters(i)	
if(vcl_hrd_parameters_present_flag)	
sub_layer_hrd_parameters(i)	
}	
}	

[0095] Em alguns exemplos, o decodificador de vídeo 30 compreende uma pluralidade de núcleos de processamento. Cada dos núcleos de processamento compreende uma unidade de processamento independente, como um ASIC ou microprocessador. Cada dos núcleos de processamento pode implementar um decodificador de camada única. Um decodificador de camada única é um codificador de vídeo capaz de decodificar unidades NAL que pertencem a uma camada única de um fluxo de bits. Em tais exemplos, como o decodificador de vídeo 30 compreende múltiplos núcleos de processamento, e desse modo múltiplos decodificadores de camada única, o decodificador de vídeo 30 pode ser capaz de decodificar múltiplas camadas em paralelo. Em algumas instâncias, um núcleo de hardware individual de decodificador de vídeo 30 pode ser capaz de decodificar mais de uma camada.

[0096] Núcleos de processamento diferentes

podem ter capacidades diferentes. Por exemplo, núcleos de processamento diferentes podem ser capazes de decodificar dados de vídeo que pertencem a perfis, planos ou níveis diferentes. Por exemplo, um primeiro núcleo de processamento do decodificador de vídeo 30 pode ser capaz de processar dados de vídeo que pertencem a um primeiro nível, porém não a um segundo nível. Nesse exemplo, um segundo núcleo de processamento de decodificador de vídeo 30 pode ser capaz de processar dados de vídeo que pertencem ao primeiro nível e ao segundo nível. Em geral, núcleos de processamento capazes de decodificar dados de vídeo que pertencem a perfis, planos e níveis mais elevados são mais complexos e, portanto, mais caros do que núcleos de processamento capazes somente de decodificar dados de vídeo que pertencem a perfis, planos e níveis mais baixos. Por conseguinte, um fabricante de decodificador de vídeo 30 pode escolher incluir no decodificador de vídeo 30 núcleos de processamento tendo as capacidades mais baixas adequadas para tratar uma tarefa desejada.

[0097] Wang, e outros, "MV-HEV/SHVC HLS: On level definition, "Joint Collaborative Team on vídeo coding (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11 17ª reunião: Valência, ES, 27 de março - 4 de abril de 2014, documento no. JCTVC-q0145 (a seguir, "JCTVC-Q10145") descreve vários problemas relacionados às definições de nível em SHVC.

[0098] Primeiramente, se mais de 4 camadas de relação de sinal para ruído (SNR) de resolução de 720 p (ou número equivalente de pixels de luma, por exemplo, mais camadas com camadas SNR e espacial combinadas) forem necessárias, então o Nível 5 ou acima teria de ser usado. Consequentemente, o valor de CtbSizeY será igual a 32 ou 64 (por item e dos limites gerais). Para resolução como 720p

ou mais baixa, essa restrição não permite o uso de tamanhos de bloco de árvore de codificação menores como 16x16 e pode, portanto, resultar em eficiência de codificação abaixo de ótima.

[0099] Em segundo lugar, se um decodificador SHVC for implementado em um modo que consiste em 4 núcleos de decodificadores de HEVC nível 3.1, e é capaz de decodificar 4 camadas SNR de 720p, de acordo com a definição de nível atual, teria de ser dito como se conformando ao Nível 4 ou acima. Consequentemente, o decodificador pode necessitar ser capaz de decodificar quaisquer fluxos de bits de Nível 4. Entretanto, com apenas alterações de sintaxe de nível elevado (isto é, sem alteração nos núcleos de hardware), tal decodificador não seria capaz de decodificar um fluxo de bits de Nível H SHVC com 2 camadas SNR de resolução de 1080p.

[0100] Em terceiro lugar, se um decodificador SHVC não for implementado por reutilizar múltiplos núcleos de decodificador HEVC existentes, porém for implementado em um modo que possa decodificar tanto um fluxo de bits HEVC de camada única de 1080p como um fluxo de bits SHVC de duas camadas de 720p, de acordo com a definição de nível atual seria rotulado como um Nível 3.1. Entretanto, então a expressão da outra capacidade está ausente.

[0101] Em quarto lugar, se um decodificador SHVC for implementado em um modo que consiste em 4 núcleos de decodificadores HEVC 3.1 para ser capaz de decodificar 4 camadas SNR de 720p, de acordo com a definição de nível atual, seria dito como se conformando ao Nível 4 ou acima. Consequentemente, cada imagem de camada de aperfeiçoamento pode ter mais de 3 linhas de azulejo e mais de 3 colunas de azulejo, por exemplo 5 linhas de azulejo e 5 colunas de azulejo, cada azulejo com largura de 256 amostras de luma e

altura de 144 amostras de luma. Entretanto, isso iria além dos limites de Nível 3.1 e consequentemente os núcleos de decodificador para decodificar as camadas de aperfeiçoamento teriam problemas.

[0102] Em quinto lugar, todos os itens na subcláusula A.4.1 da especificação de minuta de HEVC são atualmente especificados para serem aplicados em cada camada. Entretanto, alguns desses itens não são diretamente aplicáveis a cada camada. Para o item d no tamanho de DPB, o elemento de sintaxe SPS não é aplicável para camadas de aperfeiçoamento. Além disso, o DPB no SHVC WD6 é um design sub-DPB compartilhado, desse modo o item d não pode ser diretamente aplicado a cada camada. Para os itens h e i no tamanho CPB, para operações CPB específicas de fluxo de bits, o parâmetro não pode ser aplicado a cada camada.

[0103] Em sexto lugar, restrições específicas de fluxo de bits em tamanho de CPB (por itens h e i na subcláusula A.4.1 da especificação de minuta HEVC) são necessárias. Entretanto, os itens h e i na subcláusula A.4.1 da especificação de minuta HEVC não podem ser diretamente aplicados no nível de fluxo de bits, porque se aplicados diretamente, o mesmo limite de tamanho de CPB para fluxos de bits de camada única também seria o limite para fluxos de bits de multicamadas. Isso não é escalonável para o número de camadas e somente permitiria qualidade baixa de imagem quando há muitas camadas.

[0104] Em sétimo lugar, as restrições por itens b, c, d, g, h, i e j na subcláusula A.4.2 da especificação de minuta HEVC são atualmente especificadas como sendo somente específicas da camada. Entretanto, restrição específica de fluxo de bits por esses itens devem ser especificadas, independente de se suas duplicatas específicas de camada são especificadas.

[0105] JVTVC-Q0145 propôs uma abordagem para definir níveis e capacidades de decodificador bem como outras alterações para resolver os problemas acima. O design em JCTVC-Q0145 é associado pelo menos aos seguintes problemas. Primeiramente, o design é inferior a ótimo para decodificadores SHVC/MV-HEVC que cada é construído com base em múltiplos decodificadores de camada única HEVC sem alterar os núcleos de hardware (por exemplo, sem modificar as capacidades dos núcleos de hardware) onde os múltiplos decodificadores de camada única HEVC são de capacidades diferentes, por exemplo, perfis, planos ou níveis diferentes. Em segundo lugar, os níveis podem ser definidos em termos de restrições. MV-HEVC WD8 e SHVC WD6 definem muitas dessas restrições em termos dos elementos de sintaxe ou variáveis tendo ou não tendo valores específicos ao decodificar um fluxo de bits. Em outras palavras, algumas das restrições de nível (por exemplo, taxa de bit e taxa de amostra) são especificadas como sendo específicas de fluxo de bits. Entretanto, para permitir que um decodificador de camada única HEVC seja parte de um decodificador SHVC/MV-HEVC sem alterar o núcleo de hardware e decodificar um subconjunto das camadas (isto é, uma ou mais camadas, porém nem todas as camadas de um conjunto de camadas de saída), tais restrições de nível podem necessitar ser específicas ao subconjunto das camadas.

[0106] Uma visão geral de várias características de acordo com várias técnicas dessa revelação é fornecida abaixo. Algumas implementações detalhadas de acordo com um ou mais exemplos dessa revelação são fornecidas em seções posteriores dessa revelação. Algumas das características mencionadas abaixo podem ser aplicadas independentemente e algumas das características mencionadas abaixo podem ser aplicadas em

combinação como seria reconhecido por uma pessoa versada na técnica.

[0107] Uma técnica de exemplo dessa revelação provê sinalização de um ou mais esquemas de divisão de camadas em divisões para cada conjunto de camadas de saída, com cada esquema de divisão contendo um conjunto de divisões desconexas cuja união forma o conjunto de camadas de saída e cada divisão contendo uma ou mais camadas. Em outras palavras, cada esquema de divisão de uma camada de saída especifica um modo diferente de dividir as camadas do conjunto de camadas de saída em divisões não sobrepostas. Em alguns exemplos, divisões são associadas a conjuntos de camada de saída ao invés de conjuntos de camada.

[0108] Em alguns exemplos, um esquema de divisão e uma unidade de divisão são definidas como a seguir: um conjunto de camadas de saída pode ter um ou mais esquemas de divisão. Um esquema de divisão de um conjunto de camadas de saída divide as camadas no conjunto de camadas de saída em uma ou mais divisões. Cada camada em um conjunto de camadas de saída pertencerá exatamente a uma divisão em um esquema de divisão.

[0109] Desse modo, em um exemplo, o codificador de vídeo 20 gera um fluxo de bits que compreende uma representação codificada de dados de vídeo. Como parte de gerar o fluxo de bits, o codificador de vídeo 20 inclui, no fluxo de bits, um ou mais elementos de sintaxe indicando um ou mais esquemas de divisão. Para cada esquema de divisão respectivo de um ou mais esquemas de divisão, o esquema de divisão respectivo especifica um conjunto respectivo de divisões desconexas cuja união forma um conjunto de camadas de saída. Nesse exemplo, cada divisão respectiva do conjunto respectivo de divisões desconexas contém uma ou mais camadas, o conjunto de

camadas de saída consiste em todas as camadas da pluralidade de camadas ou um subconjunto adequado da pluralidade de camadas. Além disso, o codificador de vídeo 20 transmite o fluxo de bits.

[0110] Em um exemplo similar, o decodificador de vídeo 30 decodifica um fluxo de bits que compreende uma representação codificada de dados de vídeo. Nesse exemplo, o fluxo de bits compreende uma pluralidade de camadas. Como parte de decodificar o fluxo de bits nesse exemplo, o decodificador de vídeo 30 obtém, do fluxo de bits, um ou mais elementos de sintaxe indicando um ou mais esquemas de divisão. Em alguns exemplos, o decodificador de vídeo 30 decodifica cada das divisões de um esquema de divisão específico usando núcleos de processamento diferentes em uma pluralidade de núcleos de hardware, o esquema de divisão específico sendo um de um ou mais esquemas de divisão.

[0111] Além disso, em alguns exemplos, o codificador de vídeo 20 pode incluir, em um fluxo de bits, elementos de sintaxe indicando uma pluralidade de conjuntos de camada de saída. Para cada conjunto de camadas de saída respectivo da pluralidade de conjuntos de camada de saída, o codificador de vídeo 20 pode incluir, no fluxo de bits, elementos de sintaxe indicando um ou mais esquemas de divisão para o conjunto de camadas de saída respectivo. Em um exemplo similar, o decodificador de vídeo 30 pode obter, a partir do fluxo de bits, elementos de sintaxe indicando uma pluralidade de conjuntos de camadas de saída, a pluralidade de conjuntos de camadas de saída incluindo o primeiro conjunto de camadas de saída. Nesse exemplo, para cada conjunto de camadas de saída respectivos da pluralidade de conjuntos de camadas de saída, o decodificador de vídeo 30 pode obter, a partir do fluxo de

bits, elementos de sintaxe indicando um ou mais esquemas de divisão para o conjunto de camadas de saída respectivo.

[0112] De acordo com alguns exemplos dessa revelação, os termos “esquema de divisão” e “unidade de divisão” têm as seguintes definições:

Esquema de divisão: uma divisão de camadas em um conjunto de camadas de saída em uma ou mais divisões de modo que cada camada no conjunto de camadas de saída seja incluída exatamente em uma divisão do esquema de divisão e cada divisão do esquema de divisão contenha uma ou mais camadas.

Unidade de divisão: um subconjunto de unidade de acesso contendo um conjunto de unidades NAL de acordo com um esquema de divisão.

[0113] Como indicado acima, uma extensão VPS pode especificar informação PTL respectiva para cada conjunto de camadas de saída respectivo. Por exemplo, a extensão VPS pode incluir, para cada conjunto de camadas de saída respectivo, um elemento de sintaxe `profile_level_tier_idx` para o conjunto de camadas de saída respectivo que identifica uma estrutura de sintaxe PTL para o conjunto de camadas de saída respectivo. A estrutura de sintaxe PTL para o conjunto de camadas de saída respectivo especifica a informação PTL para o conjunto de camadas de saída respectivo.

[0114] Como o decodificador de vídeo 30 pode usar a informação PTL de um conjunto de camadas de saída para determinar se o decodificador de vídeo 30 é capaz de decodificar o conjunto de camadas de saída, a informação PTL do conjunto de camadas de saída tipicamente especifica o perfil, plano e nível mais baixo necessário para decodificar o conjunto de camadas de saída como um todo. Entretanto, o perfil, plano e nível de algumas divisões do

conjunto de camadas de saída podem ser mais baixos que o perfil, plano e nível do conjunto de camadas de saída como um todo. Em outras palavras, o perfil, plano e nível do conjunto de camadas de saída pode ser o perfil, plano e nível mais elevado da divisão no esquema de divisão do conjunto de camadas de saída, onde cada das camadas do conjunto de camadas de saída pertence a uma divisão.

[0115] Consequentemente, quando divisões diferentes de um conjunto de camada de saída têm informação PTL diferente, a informação PTL para o conjunto de camadas de saída pode não ser suficiente para expressar a diversidade de informação PTL das divisões do conjunto de camadas de saída. Uma ou mais técnicas dessa revelação podem fornecer a sinalização de um perfil, um plano e um nível para cada divisão. Por exemplo, para cada divisão respectiva de cada esquema de divisão respectivo sinalizado em um fluxo de bits, o codificador de vídeo 20 pode incluir, no fluxo de bits, informação de perfil, plano e nível (PTL) para a divisão respectiva. Similarmente, para cada divisão respectiva de cada esquema de divisão respectiva sinalizado em um fluxo de bits, o decodificador de vídeo 30 pode obter, a partir do fluxo de bits, um elemento de sintaxe respectivo (por exemplo, `profile_level_tier_partition_idx`) que indica informação PTL respectiva para a divisão respectiva.

[0116] Em alguns exemplos, o decodificador de vídeo 30 determina, com base na informação PTL para divisões dos esquemas de divisão, um esquema de divisão de um ou mais esquemas de divisão. Além disso, em alguns exemplos, o esquema de divisão é determinado com base nas condições da rede que é usada para enviar o fluxo de bits para o decodificador. O decodificador de vídeo 30 pode decodificar cada divisão respectiva do esquema de divisão

de determinação usando um núcleo de hardware respectivo. Cada dos núcleos de hardware pode implementar um decodificador de camada única respectivo. Em alguns exemplos, núcleos individuais dos núcleos de hardware decodificam mais de uma camada (por exemplo, implementam um decodificador de multicamadas). Além disso, núcleos de hardware diferentes podem ter capacidades diferentes. Por exemplo, pelo menos um primeiro núcleo de hardware de uma pluralidade de núcleos de hardware do decodificador de vídeo 30 pode suportar um perfil, plano ou nível diferente de um segundo núcleo de hardware da pluralidade de núcleos de hardware do decodificador de vídeo 30. Desse modo, em um exemplo, o decodificador de vídeo 30 pode compreender um primeiro núcleo de hardware e um segundo núcleo de hardware. Nesse exemplo, o primeiro núcleo de hardware pode implementar um decodificador de nível 3 (isto é, um decodificador capaz de decodificar HEVC nível 3 e abaixo) e o segundo núcleo de hardware pode implementar um decodificador de nível 2 (isto é, um decodificador capaz de decodificar HEVC nível 2 e abaixo). Nesse exemplo, se a informação PTL de uma primeira divisão indicar que a primeira divisão pode ser decodificada por um decodificador de nível 3 ou mais elevado e se a informação PTL indicar que uma segunda divisão pode ser decodificada por um decodificador de nível 2 ou mais elevada, o decodificador de vídeo 30 pode usar o primeiro núcleo de hardware para decodificar a primeira divisão e pode usar o segundo núcleo de hardware para decodificar a segunda divisão. Em algumas instâncias, quando um decodificador de vídeo de multicamadas que inclui um ou mais núcleos de hardware para decodificar as divisões, e pelo menos um dos núcleos de decodificador está em conformidade com um de perfis escalonáveis (por exemplo, Escalonável principal ou

Escalonável principal 10) especificado, a capacidade em executar reamostragem ou amostragem ascendente de imagens pode ser incluída. Essa capacidade de reamostragem pode ser implementada em cada núcleo que é capaz de decodificar um perfil escalonável, ou implementado como bloco comum no decodificador de multicamadas.

[0117] Como brevemente descrito acima, o codificador de vídeo 20 pode incluir, em um VPS, elementos de sintaxe identificando conjuntos de camada de saída. Desse modo, conjuntos de camada de saída podem ser sinalizados em um VPS. Em alguns exemplos, conjuntos de camada de saída também podem ser sinalizados por um meio externo. O meio externo pode ser algum mecanismo diferente do fluxo de bits de vídeo. Desse modo, uma camada de saída sinalizada por um meio externo e compreendendo camadas de saída de um fluxo de bits não é necessariamente sinalizado no fluxo de bits.

[0118] De acordo com uma ou mais técnicas dessa revelação, para cada esquema de divisão respectivo de cada conjunto de camadas de saída, uma mensagem SEI ou VUI VPS de um fluxo de bits pode identificar um conjunto de parâmetros de HRD para o esquema de divisão respectivo. Desse modo, em alguns exemplos dessa revelação, os parâmetros de HRD para cada esquema de divisão para cada conjunto de camadas de saída sinalizado no VPS ou por meio externo são sinalizados nas mensagens SEI ou VUI VPS. Como conjuntos de parâmetros HRD são identificados para cada esquema de divisão, o decodificador de vídeo 30 ou outro dispositivo pode ser capaz de usar os parâmetros HRD para o esquema de divisão para testar se o fluxo de bits é um fluxo de bits de conformação quando dividido de acordo com o esquema de divisão.

[0119] Em tal exemplo, o codificador de vídeo

20 gera um fluxo de bits que compreende uma representação codificada dos dados de vídeo. Nesse exemplo, o fluxo de bits compreende uma pluralidade de camadas. Como parte de gerar o fluxo de bits nesse exemplo, o codificador de vídeo 20 inclui, no fluxo de bits, um ou mais elementos de sintaxe indicando um ou mais esquemas de divisão para um conjunto de camadas de saída. Nesse exemplo, o conjunto de camadas de saída consiste em todas as camadas da pluralidade de camadas ou um subconjunto adequado da pluralidade de camadas. Para cada esquema de divisão respectivo de um ou mais esquemas de divisão, o esquema de divisão respectivo especifica uma divisão do conjunto de camadas de saída em uma ou mais divisões, em que cada camada no conjunto de camadas de saída pertence exatamente a uma divisão no esquema de divisão respectivo. Além disso, como parte de gerar o fluxo de bits nesse exemplo, o codificador de vídeo 20 pode, para cada esquema de divisão respectivo, incluir, em uma mensagem SEI no fluxo de bits ou VUI em um VPS no fluxo de bits, um conjunto respectivo de elementos de sintaxe indicando conjuntos de parâmetros HRD para o respectivo esquema de divisão. O codificador de vídeo 20 pode transmitir o fluxo de bits.

[0120] Em um exemplo similar, um dispositivo de processamento de vídeo, como codificador de vídeo 20 ou decodificador de vídeo 30, pode decodificar um fluxo de bits que compreende uma representação codificada dos dados de vídeo. Nesse exemplo, o fluxo de bits compreende uma pluralidade de camadas. Como parte de decodificação do fluxo de bits nesse exemplo, o dispositivo de processamento de vídeo obtém, a partir do fluxo de bits, um ou mais elementos de sintaxe indicando um ou mais esquemas de divisão para um conjunto de camadas de saída. Adicionalmente, para cada esquema de divisão respectivo, o

decodificador de vídeo 30 pode obter, de uma mensagem SEI no fluxo de bits ou informação de capacidade de uso de vídeo (VUI) em um VPS no fluxo de bits, um conjunto respectivo de conjuntos de parâmetros HRD para o esquema de divisão respectivo. Para cada divisão respectiva do esquema de divisão respectivo, o dispositivo de processamento de vídeo testa, com base nos parâmetros HRD para o respectivo esquema de divisão, a respectiva divisão para conformidade com um padrão de codificação de vídeo.

[0121] Pelo menos em alguns documentos de padrões, como SHVC WD6, MV-HEVC WD8, e a especificação de minuta de HEVC, perfis, planos e níveis são definidos em termos de restrições e limites de nível. Um limite de nível pode especificar um limite sobre o valor de uma variável ou elemento de sintaxe. Uma restrição pode especificar uma exigência ou limitação. Por exemplo, JCTV3V-H1003 define um perfil "Estéreo principal" e inclui, entre várias outras restrições, uma restrição exigindo que o fluxo de bits contenha uma camada com `nuh_layer_id` igual a `i` para o qual `ViewScalExtLayerFlag[i]` é igual a 1. Muitos dos limites e restrições de nível especificadas em SHVC WD6, MV-HEVC WD8 e a especificação de minuta de HEVC são aplicáveis a fluxos de bits. Por exemplo, o parágrafo G.11.1.2 de MV-HEVC WD8 descreve o perfil de "Estéreo principal" e especifica com ênfase adicionado, **"Fluxos de bits que se conformam** ao perfil Estéreo Principal obedecerão às seguintes limitações...."

[0122] Entretanto, em alguns exemplos dessa revelação, divisões de fluxo de bits diferentes podem ter informações de PTL diferentes. Para suportar divisões de fluxos de bits diferentes tendo informações de PTL diferentes, as definições dos limites e restrições de nível podem ser expressas em termos de divisões ao invés de

fluxos de bits. Desse modo, de acordo com algumas técnicas de exemplo dessa revelação, todos os limites e restrições de nível, exceto para três na largura de imagem, altura de imagem e tamanho de buffer de imagem sub-decodificada (DPB) que são naturalmente específicos de camada, são especificados como sendo específicos de divisão. Por exemplo, todos os limites e restrições de nível de um perfil, plano ou nível, exceto por limites e restrições em largura de imagem, altura de imagem, e tamanho de sub-DPB, são definidos como sendo específicos de divisão.

[0123] Como observado acima, restrições de tamanho de sub-DPB são naturalmente específicos de camada. Para elaborar, o DPB pode ser virtualmente dividido em sub-DPBs. Imagens de cada camada são armazenadas em um sub-DPB separado e cada sub-DPB contém uma camada. Por exemplo, um conjunto de camada de saída contendo duas camadas, o DPB pode ser dividido em dois sub-DPBs, com o primeiro sub-DPB contendo a primeira camada e o segundo sub-DPB contendo a segunda camada. ao invés de especificar o tamanho de DPB como um total, o tamanho de sub-DPB (aquele exigido para cada camada) é especificado no fluxo de bits. Em tais exemplos, os limites de nível indiretamente põem algumas restrições sobre o tamanho geral de DPB.

[0124] De acordo com uma ou mais técnicas dessa revelação, uma ou mais restrições adicionais são introduzidas. Uma restrição possível, por exemplo, é que para cada conjunto de camadas de saída, haverá somente um esquema de divisão que contém somente uma divisão e aquela divisão contém todas as camadas no conjunto de camadas de saída. Cada esquema de divisão está fornecendo flexibilidade de um esquema de fornecimento, isto é, quais camadas podem ser fornecidas juntas. Quando todas as camadas estão em uma divisão, o único modo que se pode ter

esquemas de fornecimento diferentes é por ter taxas de bit diferentes, por exemplo. Como resultado, essa limitação ajudaria a reduzir alguma sintaxe adicional.

[0125] Em outra restrição de exemplo, a soma dos tamanhos de imagem de uma divisão é restrita a ser menor que as amostras de luma máximas especificadas para o nível da divisão. Essa restrição pode assegurar que se uma divisão com múltiplas camadas é decodificada por um único núcleo, os tamanhos gerais de DPB do núcleo atual não são excedidos. Além disso, essa restrição pode assegurar que o tamanho total de imagens de todas as camadas não excede o tamanho de imagem de uma camada que possa ter sido decodificável pelo decodificador de camada única. Essa restrição pode permitir extensão mais fácil de núcleos de camada única para fins de multicamadas (por exemplo, sem ter de potencialmente adicionar memória adicional).

[0126] Em outra restrição de exemplo, os tripletos sinalizados de perfil, plano e nível associados ao conjunto de camadas de saída também é associado à divisão que contém todas as camadas no conjunto de camadas de saída.

[0127] Além disso, de acordo com uma técnica de exemplo dessa revelação, a capacidade de decodificação de um decodificador é expressa como se conformando a uma lista de tripletos de perfil, plano e nível. Em alguns exemplos, o número de entradas na lista indica o número de decodificadores de camada única usados para construir o decodificador de multicamadas, e cada triplete de perfil, plano e nível indica o perfil, plano e nível de um dos decodificadores de camada única.

[0128] Desse modo, em um exemplo, um dispositivo de processamento de vídeo, como decodificador de vídeo 30 ou outro dispositivo, pode determinar, com base

em uma lista de entradas de triplete, se um decodificador de vídeo de multicamadas (por exemplo, decodificador de vídeo 30) é capaz de decodificar um fluxo de bits que compreende uma representação codificada dos dados de vídeo de multicamadas. O número de entradas de triplete na lista pode ser menor ou igual a um número de decodificadores de camada única no decodificador de vídeo de multicamadas. Cada entrada de triplete respectiva na lista de entradas de triplete indica um perfil, um plano e um nível para um decodificador de camada única respectivo no decodificador de vídeo de multicamadas. Em alguns exemplos, o dispositivo de processamento de vídeo pode determinar que o decodificador de multicamadas seja capaz de decodificar o fluxo de bits por determinar que o fluxo de bits especifica um esquema de divisão onde a informação PTL de cada divisão do esquema de divisão indica que a divisão pode ser decodificada por pelo menos um dos decodificadores de camada única. Com base no decodificador de vídeo de multicamadas sendo capaz de decodificar o fluxo de bits, o decodificador de vídeo de multicamadas pode decodificar o fluxo de bits.

[0129] Em alguns exemplos dessa revelação, um decodificador que se conforma a uma lista de tripletos de perfil, plano e nível é exigido ser capa de decodificar qualquer conjunto de camadas de saída de conformação para o qual haja pelo menos um esquema de divisão que atenda a seguinte condição: para cada divisão do esquema de divisão, um dos decodificadores de camada única do decodificador de multicamadas pode ser atribuído para decodificar a divisão. Quando um decodificador de camada única é atribuído uma divisão para decodificação, o decodificador de camada única pode decodificar a divisão. Em alguns exemplos, cada das divisões pode ser atribuída a um decodificador diferente

dos decodificadores de camada única. Em outros exemplos, duas ou mais divisões podem ser atribuídas a um decodificador único dos decodificadores de camada única. Além disso, em alguns exemplos, o número de tripletos PTL anunciado pelo decodificador e sua associação de cada triplete a um núcleo é somente uma possibilidade. Em tais exemplos, o decodificador pode ter qualquer implementação interna (Por exemplo, apenas um super-núcleo) porém ainda anunciar múltiplos tripletos PTL.

[0130] Desse modo, em alguns exemplos, um dispositivo de processamento de vídeo, como decodificador de vídeo 30 ou outro dispositivo, pode obter, a partir de um fluxo de bits, um ou mais elementos de sintaxe que indicam um ou mais esquemas de divisão. Para determinar se um decodificador de vídeo de multicamadas, como decodificador de vídeo 30, é capaz de decodificar o fluxo de bits, o dispositivo de processamento de vídeo pode determinar, com base em uma condição sendo atendida, que o decodificador de vídeo de multicamadas é capaz de decodificar o fluxo de bits. Nesse exemplo, a condição é atendida com base em haver um esquema de divisão adequado em um ou mais esquemas de divisão. Para cada divisão respectiva do esquema de divisão adequado, um dos decodificadores de camada única do decodificador de vídeo de multicamadas pode ser atribuído para decodificar a divisão respectiva.

[0131] De acordo com outras técnicas de exemplo dessa revelação, testes de conformidade de fluxo de bits são especificados para testar a conformidade de cada divisão, em um modo que imagens decodificadas e outras informações necessárias para todas as camadas de referência diretas ou indiretas das camadas na divisão são assumidas como sendo disponíveis similarmente como a especificação

atual nas recentes especificações SHVC/MV-HEVC para a camada base quando é externamente fornecida.

[0132] Em alguns exemplos, os testes de conformidade para uma divisão de um fluxo de bits são definidos, e a conformidade do fluxo de bits é definida com base na conformidade de todas as divisões de cada esquema de divisão especificado para o fluxo de bits. Em alguns exemplos, quando uma divisão é selecionada para teste, a decodificação das outras divisões no esquema de divisão respectivo é considerada como sendo decodificada por um HRD de acordo com os parâmetros especificados para aquelas divisões.

[0133] A figura 2 é um diagrama de blocos ilustrando uma implementação de exemplo de decodificador de vídeo 30 compreendendo uma pluralidade de decodificadores de camada única, de acordo com uma técnica dessa revelação. No exemplo da figura 2, o decodificador de vídeo 30 compreende decodificadores de camada única 50A-50D (coletivamente, "decodificadores de camada única 50"). Cada dos decodificadores de camada única 50 pode ser implementado por um núcleo de hardware separado respectivo. Decodificadores diferentes de decodificadores de camada única 50 podem ter capacidades diferentes. Por exemplo, decodificadores diferentes de decodificadores de camada única 50 podem ser capazes de decodificar fluxos de bits se conformando a perfis, planos e níveis diferentes. Em alguns exemplos, o decodificador de vídeo 30 pode determinar, com base na informação PTL para divisões de esquemas de divisão, um esquema de divisão específico a partir de um ou mais esquemas de divisão. O decodificador de vídeo 30 pode decodificar cada divisão respectiva do esquema de divisão específico usando um decodificador respectivo de decodificadores de camada única 50.

[0134] A figura 3 é um diagrama de blocos ilustrando um codificador de vídeo de exemplo 20 que pode implementar técnicas dessa revelação. A figura 3 é fornecida para fins de explicação e não deve ser considerada limitadora das técnicas como amplamente exemplificado e descrito nessa revelação. Para fins de explicação, essa revelação descreve codificador de vídeo 20 no contexto de codificação HEVC. Entretanto, as técnicas dessa revelação podem ser aplicáveis a outros padrões ou métodos de codificação.

[0135] No exemplo da figura 3, o codificador de vídeo 20 inclui uma unidade de processamento de predição 200, uma unidade de geração residual 202, uma unidade de processamento de transformada 204, uma unidade de quantização 206, uma unidade de quantização inversa 208, uma unidade de processamento de transformada inversa 210, uma unidade de reconstrução 212, uma unidade de filtro 214, um buffer de imagem decodificada 216 e uma unidade de codificação por entropia 218. A unidade de processamento de predição 200 inclui uma unidade de processamento Inter-predição 220 e uma unidade de processamento intra-predição 226. A unidade de processamento inter-predição 220 inclui uma unidade de estimação de movimento 222 e uma unidade de compensação de movimento 224. Em outros exemplos, o codificador de vídeo 20 pode incluir mais, menos ou diferentes componentes funcionais.

[0136] O codificador de vídeo 20 pode receber dados de vídeo. Memória de dados de vídeo 201 pode armazenar dados de vídeo a serem codificados pelos componentes do codificador de vídeo 20. Os dados de vídeo armazenados na memória de dados de vídeo 201 podem ser obtidos, por exemplo, a partir da fonte de vídeo 18. O buffer de quadro decodificador 216 pode ser uma memória de

quadro de referência que armazena dados de vídeo de referência para uso na codificação de dados de vídeo por codificador de vídeo 20, por exemplo, em modos intra- ou inter-codificação. A memória de dados de vídeo 201 e buffer de quadro decodificado 216 podem ser formados por qualquer de uma variedade de dispositivos de memória, como memória de acesso aleatório dinâmica (DRAM), incluindo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM), ou outros tipos de dispositivos de memória. A memória de dados de vídeo 201 e buffer de quadro decodificado 216 podem ser fornecidos pelo mesmo dispositivo de memória ou dispositivos de memória separados. Em vários exemplos, a memória de dados de vídeo 201 pode ser no chip com outros componentes de codificador de vídeo 20, ou fora de chip em relação àqueles componentes.

[0137] O codificador de vídeo 20 pode codificar cada CTU em uma fatia de uma imagem dos dados de vídeo. Cada das CTUs pode ter blocos de árvore de codificação de luma de tamanho igual (CTBs) e CTBs correspondentes da imagem. Como parte de codificar uma CTU, a unidade de processamento de predição 200 pode executar divisão de quad-tree para dividir as CTBs da CTU em blocos progressivamente menores. O bloco menor pode ser blocos de codificação de CUs. Por exemplo, a unidade de processamento de predição 200 pode dividir um CTB de uma CTU em quatro sub-blocos de tamanho igual, dividir um ou mais dos sub-blocos em quatro sub-sub-blocos de tamanho igual, e etc.

[0138] O codificador de vídeo 20 pode codificar CUs de uma CTU para gerar representações codificadas das CUs (isto é, CUs codificadas). Como parte de codificar uma CU, a unidade de processamento de predição 200 pode dividir os blocos de codificação associados a CU

entre uma ou mais PUs da CU. Desse modo, cada PU pode ser associada a um bloco de predição de luma e blocos de predição de croma correspondentes. O codificador de vídeo 20 e decodificador de vídeo 30 podem suportar PUs tendo vários tamanhos. Como indicado acima, o tamanho de uma CU pode se referir ao tamanho do bloco de codificação de luma da CU e o tamanho de uma PU pode se referir o tamanho de um bloco de predição de luma da PU. Assumindo que o tamanho de uma CU específica é $2N \times 2N$, o codificador de vídeo 20 e o decodificador de vídeo 30 podem suportar tamanhos de PU de $2N \times 2N$ ou $N \times N$, para inter-predição e tamanhos de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$ ou similares para inter predição. O codificador de vídeo 20 e decodificador de vídeo 30 podem também suportar divisão assimétrica para tamanhos de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ e $nR \times 2N$ para inter predição.

[0139] A unidade de processamento inter-predição 220 pode gerar dados preditivos para uma PU por executar inter predição em cada PU de uma CU. Os dados preditivos para a PU podem incluir um ou mais blocos de amostra preditiva da PU e informações de movimento para a PU. A unidade de processamento de inter-predição 220 pode executar operações diferentes para uma CU de uma CU dependendo de se a PU está em uma fatia I, uma fatia P, ou uma fatia B. em uma fatia I, todas as PUs são intra preditas. Consequentemente, se a PU estiver em uma fatia I, a unidade de processamento inter-predição 220 não executa inter predição na PU.

[0140] Se uma PU estiver em uma fatia P, a unidade de estimação de movimento 222 pode buscar as imagens de referência em uma lista de imagens de referência (por exemplo, "RefPicList0") para uma região de referência para a PU. A região de referência para a PU pode ser uma

região, em uma imagem de referência, que contém amostras que mais estreitamente correspondem com os blocos de predição da PU. A unidade de estimação de movimento 222 pode gerar um índice de referência que indica uma posição na RefPicList0 da imagem de referência contendo a região de referência para a PU. Além disso, a unidade de estimação de movimento 222 pode gerar um vetor de movimento que indica um deslocamento espacial entre um bloco de codificação da PU e uma localização de referência associada à região de referência. Por exemplo, o vetor de movimento pode ser um vetor bidimensional que fornece um deslocamento a partir das coordenadas na imagem atual para coordenadas em uma imagem de referência. A unidade de estimação de movimento 222 pode transmitir o índice de referência e o vetor de movimento como as informações de movimento da PU. A unidade de compensação de movimento 224 pode gerar os blocos de amostra preditivos da PU com base em amostras efetivas ou interpoladas na localização de referência indicada pelo vetor de movimento da PU.

[0141] Se uma PU estiver em uma fatia B, a unidade de estimação de movimento 222 pode executar uni-predição ou bi-predição para a PU. Para executar uni-predição para a PU, a unidade de estimação de movimento 222 pode buscar as imagens de referência de RefPicList0 ou uma segunda lista de imagem de referência ("RefPicList'") para uma região de referência para a PU. A unidade de estimação de movimento 222 pode transmitir, como a informação de movimento da PU, um índice de referência que indica uma posição e RefPicList0 ou RefPicList1 da imagem de referência que contém a região de referência, um vetor de movimento que indica um deslocamento espacial entre um bloco de amostra da PU e uma localização de referência associada à região de referência, e um ou mais indicadores

de direção de predição que indicam se a imagem de referência está em RefPicList0 ou RefPicList1. A unidade de compensação de movimento 224 pode gerar os blocos preditivos da PU com base pelo menos em parte em amostras efetivas ou interpoladas no local de referência indicado pelo vetor de movimento da PU.

[0142] Para executar inter predição bidirecional para uma PU, a unidade de estimação de movimento 222 pode buscar as imagens de referência em RefPicList0 para uma região de referência para a PU e também pode buscar as imagens de referência em RefPicList1 para outra região de referência para a PU. A unidade de estimação de movimento 222 pode gerar índices de imagem de referência que indicam posições em RefPicList0 e RefPicList1 das imagens de referência que contêm as regiões de referência. Além disso, a unidade de estimação de movimento 222 pode gerar vetores de movimento que indicam deslocamentos espaciais entre os locais de referência associados às regiões de referência e um bloco de predição da PU. A informação de movimento da PU pode incluir os índices de referência e os vetores de movimento da PU. A unidade de compensação de movimento 224 pode gerar os blocos preditivos da PU com base pelo menos em parte em amostras efetivas ou interpoladas nos locais de referência indicados pelos vetores de movimento da PU.

[0143] A unidade de processamento intra-predição 226 pode gerar dados preditivos para uma PU por executar intra predição na PU. Os dados preditivos para a PU podem incluir blocos preditivos para a PU e vários elementos de sintaxe. A Unidade de processamento intra-predição 226 pode executar intra predição em PUs em fatias I, fatias P e fatias B.

[0144] Para executar intra predição em uma PU,

a unidade de processamento intra-predição 226 pode usar múltiplos modos intra predição para gerar múltiplos conjuntos de dados preditivos para a PU. Ao executar intra predição usando um modo de intra predição específico, a unidade de processamento intra-predição 226 pode gerar blocos preditivos para a PU usando um conjunto específico de amostras a partir de blocos vizinhos. Os blocos vizinhos podem estar acima, acima e à direita, acima e à esquerda, ou à esquerda dos blocos de predição da PU, assumindo uma ordem de codificação esquerda para direita, superior para inferior para PUs, CUs e CTUs. A unidade de processamento intra-predição 226 pode usar vários números de modos intra predição, por exemplo, 33 modos intra predição direcionais. Em alguns exemplos, o número de modos intra predição pode depender do tamanho dos blocos de predição da PU.

[0145] A unidade de processamento de predição 200 pode selecionar os dados preditivos para PUs de uma CU a partir entre os dados preditivos gerados pela unidade de processamento inter-predição 220 para as PUs ou os dados preditivos gerados pela unidade de processamento intra-predição 226 para as PUs. Em alguns exemplos, a unidade de processamento de predição 200 seleciona os dados preditivos para as PUs da CU com base em métrica de distorção/taxa dos conjuntos de dados preditivos. Os blocos de amostra preditivos dos dados preditivos selecionados podem ser mencionados aqui como os blocos preditivos selecionados.

[0146] A unidade de geração residual 202 pode gerar, com base nos blocos de codificação de uma CU e os blocos preditivos selecionados das PUs da CU, blocos residuais da CU. Por exemplo, a unidade de geração residual 202 pode gerar, com base no bloco de codificação de luma, Cb e Cr de uma CU e os blocos preditivos selecionados de luma, Cb e Cr as PUs dos blocos residuais de CU, luma, Cb e

Cr da CU. Em alguns exemplos, a unidade de geração residual 202 pode gerar os blocos residuais da CU de modo que cada amostra nos blocos residuais tenha um valor igual a uma diferença entre uma amostra em um bloco de codificação da CU e uma amostra correspondente em um bloco preditivo selecionado correspondente de uma PU da CU.

[0147] A unidade de processamento de transformada 204 pode executar divisão quad-tree para dividir os blocos residuais de uma CU em blocos de transformada de TUs da CU. Desse modo, uma TU pode ter um bloco de transformada de luma e dois blocos de transformada de croma correspondentes. Os tamanhos e posições dos blocos de transformada de TUs de uma CU podem ou não ser baseados nos tamanhos e posições dos blocos de predição das PUs da CU.

[0148] A unidade de processamento de transformada 204 pode gerar blocos de coeficiente de transformada para cada TU de uma CU por aplicar uma ou mais transformadas aos blocos de transformada da TU. A unidade de processamento de transformada 204 pode aplicar várias transformadas a um bloco de transformada de uma TU. Por exemplo, a unidade de processamento de transformada 204 pode aplicar uma transformada de cosseno discreta (DCT), uma transformada direcional, ou uma transformada conceptualmente similar a um bloco de transformada. Em alguns exemplos, a unidade de processamento de transformada 204 não aplica transformadas em um bloco de transformada. Em tais exemplos, o bloco de transformada pode ser tratado com um bloco de coeficiente de transformada.

[0149] A unidade de quantização 206 pode quantizar os coeficientes de transformada em um bloco de coeficiente. O processo de quantização pode reduzir a profundidade de bit de alguns ou todos os coeficientes de

transformada. Por exemplo, um coeficiente de transformada de bit-n pode ser arredondado para um coeficiente de transformada de bit-m durante quantização, onde n é maior que m. A unidade de quantização 206 pode quantizar um bloco de coeficiente de uma TU de uma CU com base em um valor de parâmetro de quantização (QP) para a CU. O codificador de vídeo 20 pode ajustar o grau de quantização aplicado aos blocos de coeficiente de a uma CU por ajustar o valor QP para a CU. A quantização pode introduzir perda de informação, desse modo coeficientes de transformada quantizadas podem ter precisão mais baixa que os originais.

[0150] A unidade de quantização inversa 208 e a unidade de processamento de transformada inversa 210 podem aplicar transformadas inversa e quantização inversa em um bloco de coeficiente, respectivamente, para reconstruir um bloco residual a partir do bloco de coeficiente. A unidade de reconstrução 212 pode adicionar o bloco residual reconstruído a amostras correspondentes a partir de um ou mais blocos preditivos gerados pela unidade de processamento de predição 200 para produzir um bloco de transformada reconstruído de uma TU. Por reconstruir blocos de transformada para cada TU de uma CU desse modo, o codificador de vídeo 20 pode reconstruir os blocos de codificação da CU.

[0151] A unidade de filtro 214 pode executar uma ou mais operações de desbloqueio para reduzir artefatos de bloqueio nos blocos de codificação de uma CU. O buffer de imagem decodificada 216 pode armazenar os blocos de codificação reconstruídos após a unidade de filtro 214 executar uma ou mais operações de desbloqueio nos blocos de codificação reconstruídos. A unidade de processamento inter-predição 220 pode usar uma imagem de referência que contém os blocos de codificação reconstruídos para executar

inter predição em PUs de outras imagens. Além disso, a unidade de processamento intra-predição 226 pode usar blocos de codificação reconstruídos em buffer de imagem decodificada 216 para executar intra predição nas outras PUs na mesma imagem que a CU.

[0152] A unidade de codificação por entropia 218 pode receber dados a partir de outros componentes funcionais do codificador de vídeo 20. Por exemplo, a unidade de codificação por entropia 218 pode receber blocos de coeficiente a partir da unidade de quantização 206 e pode receber elementos de sintaxe a partir da unidade de processamento de predição 200. A unidade de codificação por entropia 218 pode executar uma ou mais operações de codificação por entropia nos dados para gerar dados codificados por entropia. Por exemplo, a unidade de codificação por entropia 218 pode executar uma operação de codificação de comprimento variável adaptável ao contexto (CAVLC), uma operação CABAC, uma operação de codificação de comprimento variável a variável (V2V), uma operação de codificação aritmética binária adaptável ao contexto baseado em sintaxe (SBAC), uma operação de codificação de Entropia de divisão de intervalo de probabilidade (PIPE), uma operação de codificação Golomb-exponencial, ou outro tipo de operação de codificação por entropia sobre os dados. O codificador de vídeo 20 pode transmitir um fluxo de bits que inclui dados codificados por entropia gerados por unidade de codificação por entropia 218.

[0153] De acordo com alguns exemplos dessa revelação, a unidade de codificação por entropia 218 pode incluir, no fluxo de bits, um ou mais elementos de sintaxe indicando um ou mais esquemas de divisão. Para cada esquema de divisão respectivo de um ou mais esquemas de divisão, o esquema de divisão respectivo especifica um conjunto

respectivo de divisões desconexas cuja união forma um conjunto de camadas de saída, em que cada divisão respectiva do conjunto respectivo de divisões desconexas contém uma ou mais camadas, o conjunto de camadas de saída pode consistir em todas as camadas da pluralidade de camadas ou um subconjunto adequado da pluralidade de camadas.

[0154] A figura 4 é um diagrama de blocos ilustrando um decodificador de vídeo de exemplo 30 que pode implementar técnicas dessa revelação. A figura 4 é fornecida para fins de explicação e não é limitadora das técnicas como amplamente exemplificado e descrito nessa revelação. Para fins de explicação, essa revelação descreve decodificador de vídeo 30 no contexto de codificação de HEVC. Entretanto, as técnicas dessa revelação podem ser aplicáveis a outros padrões ou métodos de codificação.

[0155] No exemplo da figura 4, o decodificador de vídeo 30 inclui uma unidade de decodificação por entropia 250, uma unidade de processamento de predição 252, uma unidade de quantização inversa 254, uma unidade de processamento de transformada inversa 256, uma unidade de reconstrução 258, uma unidade de filtro 260, e um buffer de imagem decodificada 262. A unidade de processamento de predição 252 inclui uma unidade de compensação de movimento 264 e uma unidade de processamento intra-predição 266. Em outros exemplos, o decodificador de vídeo 30 pode incluir mais, menos ou diferentes componentes funcionais.

[0156] Um buffer de imagem codificada (CPB) 251 pode receber e armazenar dados de vídeo codificados (por exemplo, unidades NAL) de um fluxo de bits. Os dados de vídeo armazenados em CPB 151 podem ser obtidos, por exemplo, de mídia legível em computador 16, por exemplo, de uma fonte de vídeo local, como uma câmera, através de

comunicação de rede cabeada ou sem fio de dados de vídeo, ou por acessar mídia de armazenagem de dados físicos. CPB 151 pode formar uma memória de dados de vídeo que armazena dados de vídeo codificados a partir de um fluxo de bits de vídeo codificado. O buffer de imagem decodificada 162 pode ser uma memória de imagem de referência que armazena dados de vídeo de referência para uso em decodificar dados de vídeo por decodificador de vídeo 30, por exemplo, em modos intra- ou inter-codificação. CPB 151 e buffer de imagem decodificada 162 podem ser formados por qualquer de uma variedade de dispositivos de memória, como memória de acesso aleatório dinâmica (DRAM), incluindo DRAM síncrona (SDRAM), RAM magnetorresistivo (MRAM), RAM resistiva (RRAM), ou outros tipos de dispositivos de memória. CPB 151 e buffer de imagem decodificada 162 podem ser fornecidos pelo mesmo dispositivo de memória ou dispositivos de memória separados. Em vários exemplos, CPB 151 pode estar em chip com outros componentes do decodificador de vídeo 30 ou fora de chip em relação àqueles componentes.

[0157] A unidade de decodificação por entropia 250 pode receber unidades NAL a partir de CPB 251 e analisar as unidades NAL para obter elementos de sintaxe a partir do fluxo de bits. A unidade de decodificação por entropia 250 pode decodificar por entropia elementos de sintaxe codificados por entropia na unidade NAL. A unidade de processamento de predição 252, unidade de quantização inversa 254, unidade de processamento de transformada inversa 256, unidade de reconstrução 258, e unidade de filtro 260 podem gerar dados de vídeo decodificados com base nos elementos de sintaxe obtidos a partir do fluxo de bits.

[0158] Por exemplo, a unidade de decodificação por entropia 250 pode obter, a partir do fluxo de bits, um

ou mais elementos de sintaxe indicando um ou mais esquemas de divisão. A unidade de processamento de predição 252, unidade de quantização inversa 254, unidade de processamento de transformada inversa 256, unidade de reconstrução 258, e unidade de filtro 260 podem decodificar as divisões de um esquema de divisão específica. Em alguns exemplos, para cada esquema de divisão respectiva de um ou mais esquemas de divisão, a unidade de decodificação por entropia 250 obtém, a partir de uma mensagem SEI no fluxo de bits ou VUI em um VPS no fluxo de bits, um conjunto respectivo de conjuntos de parâmetros HRD para o esquema de divisão respectivo. O decodificador de vídeo 30 pode testar, com base nos parâmetros HRD para um ou mais esquemas de divisão, cada divisão de cada de um ou mais esquemas de divisão para conformidade com um padrão de codificação de vídeo.

[0159] As unidades NAL do fluxo de bits podem incluir unidades NAL de fatia codificada. Como parte de decodificar o fluxo de bits, a unidade de decodificação por entropia 250 pode decodificar por entropia elementos de sintaxe a partir das unidades NAL de fatia codificada. Cada das fatias codificadas pode incluir um cabeçalho de fatia e dados de fatia. O cabeçalho de fatia pode conter elementos de sintaxe pertinentes a uma fatia.

[0160] Além de obter os elementos de sintaxe a partir do fluxo de bits, o decodificador de vídeo 30 pode executar uma operação de decodificação em uma CU. Por executar a operação de decodificação em uma CU, o decodificador de vídeo 30 pode reconstruir blocos de codificação da CU.

[0161] Como parte de executar uma operação de reconstrução em uma CU, a unidade de quantização inversa 254 pode quantizar inverso, isto é, desquantizar, blocos de

coeficiente associados de TUs da CU. A unidade de quantização inversa 254 pode usar um valor QP para a CU da TU para determinar um grau de quantização e, de modo semelhante, um grau de quantização inversa para unidade de quantização inversa 254 aplicar. Isto é, a razão de compressão, isto é, a razão do número de bits usados para representar a sequência original e a comprimida, pode ser controlada por ajustar o valor do QP usado ao quantizar coeficientes de transformada. A razão de compressão pode depender também do método de codificação por entropia empregado.

[0162] Após a unidade de quantização inversa 254 quantizar inverso um bloco de coeficiente, a unidade de processamento de transformada inversa 256 pode aplicar uma ou mais transformadas inversas ao bloco de coeficiente para gerar um bloco residual para a TU. Por exemplo, a unidade de processamento de transformada inversa 256 pode aplicar um DCT inverso, uma transformada de número inteiro inversa, uma transformada Karhunen-Loeve inversa (KLT), uma transformada rotacional inversa, uma transformada direcional inversa, ou outra transformada inversa ao bloco de coeficiente.

[0163] Se uma PU for codificada usando intra predição, a unidade de processamento intra-predição 266 pode executar intra predição para gerar blocos preditivos para a PU. Por exemplo, a unidade de processamento intra-predição 266 pode executar intra predição para gerar os blocos luma, Cb e Cr preditivos para a PU. A unidade de processamento intra-predição 266 pode usar um modo intra predição para gerar os blocos preditivos para a PU com base nos blocos de predição de PUs espacialmente vizinhas. A unidade de processamento intra-predição 266 pode determinar o modo intra predição para a PU com base em um ou mais

elementos de sintaxe obtidos a partir do fluxo de bits.

[0164] A unidade de processamento de predição 252 pode construir uma primeira lista de imagem de referência (RefPicList0) e uma segunda lista de imagem de referência (RefPicList1) com base em elementos de sintaxe extraídos a partir do fluxo de bits. Além disso, se uma PU for codificada usando inter predição, a unidade de decodificação por entropia 250 pode obter informações de movimento para a PU. A unidade de compensação de movimento 264 pode determinar, com base nas informações de movimento da PU, uma ou mais regiões de referência para a PU. A unidade de compensação de movimento 264 pode gerar, com base em blocos de amostra em um ou mais blocos de referência para a PU, blocos preditivos para a PU. Por exemplo, a unidade de compensação de movimento 264 pode gerar, com base em amostras de um ou mais blocos de referência para a PU, luma, Cb e blocos preditivos Cr para a PU.

[0165] A unidade de reconstrução 258 pode usar os valores residuais a partir dos blocos de transformada de TUs de uma CU e os blocos preditivos das PUs da CU para reconstruir os blocos de codificação da CU. Por exemplo, a unidade de reconstrução 258 pode usar valores residuais de blocos de transformada de luma, Cb e Cr de TUs de uma CU e os blocos de luma, Cb e Cr preditivos das PUs da CU para reconstruir os blocos de codificação de luma, Cb e Cr da CU. Por exemplo, a unidade de reconstrução 258 pode adicionar amostras de blocos de transformada a amostras correspondentes dos blocos preditivos para reconstruir os blocos de codificação da CU.

[0166] A unidade de filtro 260 pode executar uma operação de desbloqueio para reduzir artefatos de bloqueio associados aos blocos de codificação da CU. O

decodificador de vídeo 30 pode armazenar os blocos de codificação da CU no buffer de imagem decodificada 262. O buffer de imagem decodificada 262 pode fornecer imagens de referência para compensação de movimento subsequente, intra predição e apresentação e um dispositivo de exibição, como dispositivo de exibição 32 da figura 1. Por exemplo, o decodificador de vídeo 30 pode executar, com base nos blocos no buffer de imagem decodificada 262, operações de intra predição ou inter predição em PUs de outras CUs. Desse modo, o decodificador de vídeo 30 pode obter, a partir do fluxo de bits, níveis de coeficiente de transformada de um bloco de coeficiente, quantizar inverso os níveis de coeficiente de transformada, aplicar uma transformada aos níveis de coeficiente de transformada para gerar um bloco de transformada, gerar, com base pelo menos em parte no bloco de transformada, um bloco de codificação e transmitir o bloco de codificação para exibição.

[0167] As seguintes seções dessa revelação descrevem alguns detalhes de implementação de algumas das técnicas de exemplo dessa revelação descritas acima. O seguinte texto mostra alterações para implementar uma ou mais técnicas dessa revelação. O texto sublinhado indica texto adicionado. Texto em *itálico* incluído em parênteses é texto deletado. Nomes de elementos de sintaxe são mostrados em **negrito**.

[0168] Como indicado acima, de acordo com algumas técnicas dessa revelação, o codificador de vídeo 20 pode sinalizar um ou mais esquemas de divisão. Cada dos esquemas de divisão divide as camadas em um conjunto de camadas de saída em uma ou mais divisões. Cada camada em um conjunto de camadas de saída pertencerá exatamente a uma divisão em um esquema de divisão.

[0169] A tabela 2 abaixo indica alterações em

sintaxe de extensão de VPS para esquemas de divisão de sinal, definidos em SHVC WD6 e MV-HEVC WD8.

TABELA 2

vps_extension() {	Descritor
avc_base_layer_flag	u(1)
...	u(1)
for(i = 1; i < NumOutputLayerSets; i++) {	
if(i > vps_num_layer_sets_minus1)	
output_layer_set_idx_minus1[i]	u(v)
if(i > vps_num_layer_sets_minus1 defaultTargetOutputLayerIdx == 2)	
for(j = 0; j < NumLayersInIdList[LayerSetIdxForOutputLayerSet[i]]; j++)	
output_layer_flag[i][j]	u(1)
profile_level_tier_idx[i]	u(v)
num_partition_schemes[i]	uc(v)
for(j = 0; j < num_partition_schemes[i]; j++) {	
num_partitions_in_scheme_minus1[i][j]	uc(v)
for(k = 0; k < num_partitions_in_scheme_minus1[i][j]; k++) {	
for(r = 0; r <= vps_max_layers_minus1; r++)	
if(layer_id_included_flag[LayerSetIdxForOutputLayerSet[i]][r])	
layer_included_in_partition_flag[i][j][k][r]	u(1)
profile_level_tier_partition_idx[i][j][k]	u(v)
}	
}	
if(NumOutputLayersInOutputLayerSet[i] == 1 && NumDirectRefLayers[OlsHighestOutputLayerId[i]] > 0)	
alt_output_layer_flag[i]	u(1)
}	
...	
}	

[0170] No exemplo da tabela 2, num_partition_schemes[i] especifica o número de esquemas de divisão especificados para o iº conjunto de camadas de saída. O valor de num_partition_schemes[i] estará na faixa de 0 a 15, inclusive. Além disso, num_partitions_in_scheme_minus1[i][j] mais 1 especifica o número de divisões no jº esquema de divisão do iº conjunto de camadas de saída. O valor de num_partitions_in_scheme_minus1[i][j] estará na faixa de 0 a

NumLayersInIdList[LayerSetIdxForOutputLayerSet[i]] - 1, inclusive.

[0171] Além disso, no exemplo da tabela 2, layer_included_in_partition_flag[i][j][k][r] igual a 1 especifica que a r^a camada especificada no VPS é incluída na k^a divisão do j^o esquema de divisão do i^o conjunto de camadas de saída. Layer_included_in_partition_flag[i][j][k][r] igual a 0 especifica que a r^a camada especifica no VPS não é incluída na k^a divisão do j^o esquema de divisão do i^o conjunto de camadas de saída.

[0172] Desse modo, para cada esquema de divisão respectivo, para cada divisão respectiva do esquema de divisão respectivo, para cada camada respectiva do conjunto de camadas de saída, o decodificador de vídeo 30 pode obter, a partir do fluxo de bits, um elemento de sintaxe (isto é, layer_included_in_partition_flag) indicando se a camada respectiva é incluída na divisão respectiva. Em um exemplo similar, para cada esquema de divisão respectivo de um ou mais esquemas de divisão, para cada divisão respectiva do esquema de divisão respectivo, para cada camada respectiva do conjunto de camadas de saída, o codificador de vídeo 20 pode incluir, em um fluxo de bits, um elemento de sintaxe (isto é, layer_included_in_partition_flag) indicando se a camada respectiva é incluída na divisão respectiva.

[0173] No exemplo da Tabela 2, é uma exigência de conformidade de fluxo de bits que para cada camada com nuh_layer_id igual a currLayerId no i^o conjunto de camadas de saída, existe um e somente um valor de k na faixa de 0 para num_partitions_in_scheme_minus[i][j], inclusive, para cada esquema de divisão com índice j do i^o conjunto de camadas de saída, de modo que

layer_included_in_partition_flag[i][j][k][LauerIdxInVps[currLayerId] é igual a 1.

[0174] Como mencionado acima, de acordo com alguns exemplos dessa revelação, a sinalização de um perfil, um plano e um nível é fornecida para cada divisão. Como mostrado no exemplo da Tabela 2, profile_level_tier_partition_idx[i][j][k] especifica o índice, na lista de estruturas de sintaxe profile_tier_level() no VPS, da estrutura de sintaxe profile_tier_level() que se aplica à k^a divisão do j^o esquema de divisão do i^o conjunto de camadas de saída. O comprimento do elemento de sintaxe profile_level_tier_partition_idx[i] é $\text{Ceil}(\text{Log}_2(\text{vps_num_profile_tier_level_minus1} + 1))$ bits. O valor de profile_level_tier_idx[0][0][0] é inferido como sendo igual a 0. O valor de profile_level_tier_partition_idx[i] estará na faixa de 0 a vps_num_profile_tier_level_minus1, inclusive.

[0175] Em outro exemplo, as alterações mostradas na Tabela 3 podem ser feitas na sintaxe de extensão de VPS para sinalizar esquemas de divisão. As alterações mostradas na tabela 3 são relativas à Tabela 2, acima.

Tabela 3

vps_extension() {	Descriptor
...	
profile_level_tier_idx[i]	u(v)
num_partitioning_schemes_minus1[i]	ue(v)
for(j = 0; j <= num_partitioning_schemes_minus1[i]; j++) {	
num_partitions_in_scheme_minus1[i][j]	ue(v)
for(k = 0; k <=	
$\frac{N_{um}[p]}{Partitions[i]} \ln[\frac{s}{Scheme[i]_{minus1}}][i][j]; k++) \{$	
for(r = 0; r <= vps_max_layers_minus1; r++)	
if(layer_id_included_flag[LayerSetIdxForOutputLayerSet[i]][r])	
layer_included_in_partition_flag[i][j][k][r]	u(1)
profile_level_tier_partition_idx[i][j][k]	u(v)
}	
}	
...	

[0176] No exemplo da Tabela 3, as seguintes alterações podem ser feitas à semântica de elementos de sintaxe específicos da extensão VPS descrita acima com referência à Tabela 2.

Num_partitioning_schemes_minus1[i]plus 1

especifica o número de esquemas de divisão especificados para o i° conjunto de camadas de saída. O valor de Num_partitioning_schemes_minus1[i] estará compreendido na faixa de 0 a 15, inclusive.

Num_partitions_in_scheme_minus1[[1]]2[i][j] plus [[1]]2 especifica o número de divisões no j° esquema de divisão do i° conjunto de camadas de saída. O valor de num_partitions_in_scheme_minus1[[1]]2[i][j] estará na faixa de 0 a

NumLayersInIdList[LayerSetIdxForOutputLayerSet[i]-1, inclusive.

O valor da variável NumPartitionInScheme[i][j] é derivado como a seguir:

NumPartitionsInScheme[i][j] = (j == 0)? 1:

Num_partitions_in_scheme_minus2[i][j] + 2.

[[É uma exigência de conformidade de fluxo de bits que quando j é maior que 0, o valor de num_partitions_in_scheme_minus1[i][j] será maior que 0.]]

Layer_included_in_partition_flag[i][j][k][r]

igual a 1 especifica que a r^a camada especificada no VPS é incluído na k^a divisão do j^o esquema de divisão do i^o conjunto de camadas de saída.

Layer_included_in_partition_flag[i][j][k][r]

igual a 0 especifica que a r^a camada especificada no VPS não é incluída na k^a divisão do j^o esquema de divisão do i^o conjunto de camadas de saída. O valor de

layer_included_in_partition_flag[i][0][0][r] é inferido como sendo igual ao valor de layer_id_included_flag[LayerSetIdxForOutputLayerSet[i]][r] para r na faixa de 0 a vps_max_layers_minus1, inclusive.

Nota 5 - *[[É uma exigência de conformidade de fluxo de bits que fl]* Para cada camada com nuh_layer_id igual a currLayerId no i^o conjunto de camadas de saída, existe um e somente um valor de k na faixa de 0 a *[[n]Num[[p]]Partitions[[i]]In[[_sd]]Scheme[[_minus1]][i][j][[, inclusive]]*, para cada esquema de divisão com índice j do conjunto de camadas de saída, de modo que

layer_included_in_partition_flag[i][j][k][LayerId xInVps[CurrLayerId] é igual a 1.

Profile_level_tier_partition_idx[i][j][k]

especifica o índice, na lista de estruturas de sintaxe de profile_tier_level() no VPS, da estrutura de sintaxe profile_tier_level() que se aplica à k^a divisão do j^o esquema de divisão do i^o conjunto de camadas de saída. O comprimento do elemento de sintaxe Profile_level_tier_partition_idx[i][j][k] é $\text{Ceil}(\text{Log2}(\text{vps_num_profile_tier_level_minus1} + 1))$ bits. O valor de profile_level_tier_partition_idx[0][0][0] é inferido como sendo igual a 0. O valor de profile_level_tier_partition_idx[0][0][0] é definido para ser igual ao profile_level_tier_idx[i] para i na faixa de 1 para NumOutputLayerSets - 1, inclusive. O valor de profile_level_tier_partition_idx[i][j][k] estará na faixa de 0 a vps_num_profile_tier_level_minus1, inclusive.

[0177] Em alguns exemplos, uma limitação é adicionada de modo que não mais de que um esquema de divisão para cada conjunto de camadas de saída contenha somente uma divisão. Além disso, em alguns exemplos, o elemento de sintaxe profile_tier_level_idx[i] não é sinalizado e o elemento de sintaxe profile_tier_partition_idx[i][j][k] é usado para sinalizar as indicações de perfil, plano e nível do esquema de divisão do conjunto de camadas de saída que contém uma divisão incluindo todas as camadas do conjunto de camadas de saída.

[0178] Além disso, de acordo com algumas

técnicas dessa revelação, limites de nível e restrições que definem perfis, planos e níveis, são específicos de divisão. O texto a seguir indica outras alterações em MV-HEVC WD8 para implementar limites de nível específicos de divisão e restrições para perfis, planos e níveis.

G.11 Perfis, planos e níveis

G.11.1 Perfis

G.11.1.2 Perfil Estéreo principal

Fluxos de bits contendo conjuntos de camada de saída com divisões de camadas se conformando ao perfil Estéreo Principal obedecerão às seguintes limitações no sub-fluxo de bits derivado para cada dos conjuntos de camadas de saída, com layerSetIdx sendo o índice do conjunto de camadas para o conjunto de camadas de saída, o sub-fluxo de bits sendo derivado por invocar o processo de extração de sub-fluxo de bits como especificado na subcláusula F.10 com o fluxo de bits, tIdTarget igual a 7, e TargetDecLayerIdList contendo os valores de nuh layer id do conjunto de camadas com o índice layerSetIdx, como entradas, as divisões sendo derivadas por invocar o processo de demultiplexagem para derivar uma divisão de fluxo de bits como especificado na subcláusula C.6 com o sub-fluxo de bits, a lista de camadas na divisão listOfLayersInPartition, e o número de camadas na divisão numLayersInPartition como entradas.

Fluxos de bits correspondendo a sub-fluxos de bits para conjuntos de camadas de saída com divisões de camadas se conformando ao perfil Estéreo Principal obedecerão às seguintes limitações:

- o sub-fluxo de bits resultando do processo de extração de sub-fluxo de bits com qualquer valor de `tIdTarget` e um valor de 0 em `layerIdListTarget` como entradas se conformará ao perfil Principal.
- o fluxo de bits conterá uma camada com `nuh_layer_id` igual a `i` para o qual `ViewScalExtLayerFlag[i]` é igual a 1.
- Quando `ViewScalExtLayerFlag[i]` é igual a 1, `Inter_view_mv_vert_constraint_flag` será igual a 1 na estrutura de sintaxe `sps_multilayer_extension()` do RBSP SPS ativo para a camada com `nuh_layer_id` igual a `i`.
- Quando `ViewScalExtLayerFlag[i]` é igual a 1, `ScalabilityId[LayerIdxInVps[i]][smIdx]` será igual a 0 para qualquer valor `smIdx` de 0 a 15, inclusive, que não seja igual a 1, para qualquer imagem codificada com `nuh_layer_id` igual a `i`.
- Quando `ViewScalExtLayerFlag[i]` é igual a 1, `Num_scaled_ref_layer_offsets` será igual a 0 em cada SPS ativo para a camada com `nuh_layer_id` igual a `i`.
- Quando `ViewScalExtLayerFlag[i]` é igual a 1, Os valores de `pic_width_in_luma_samples` e `pic_height_in_luma_samples` no SPS ativo para a camada com `nuh_layer_id` igual a `i` serão iguais aos valores de `pic_width_i_luma_samples` e `pic_height_in_luma_samples`, respectivamente, nos SPSs ativos para todas as camadas de referência direta daquela camada.
- o fluxo de bits conterá um sub-fluxo de bits consistindo em duas camadas tendo `nuh_layer_id` igual a 0 e `nuhLayerIdA` para a qual `ScalabilityId[LayerIdxInVps[ nuhLayerIdA]][smIdx]`

será igual a 0 para qualquer smIdx de 0 a 15, inclusive que não é igual a 1.

- VPSs terão `avec_base_layer_flag` igual a 0 somente.

- VPSs terão `vps_num_rep_formats_minus1` na faixa de 0 a 15, inclusive.

- SPSs terão `sps_extension_type_flag[i]` igual a 0 somente para `i` igual a 0, e na faixa de 2 a 6, inclusive.

- PPSs terão `PPS_extension_type_flag[i]` igual a 0 somente para `i` na faixa de 1 a 6, inclusive.

No restante dessa subcláusula e subcláusula G.11.3, para um sub-fluxo de bits de um conjunto de camadas de saída, `general profile ide`, `general profile compatibility flag[i]`, `general tier flag` e `general level ide` se referem ao `general profile ide`, `general profile compatibility flag[i]`, `general tier flag` e `general level ide`, respectivamente, na estrutura de sintaxe de `profile tier level()` para o conjunto de camadas de saída. A conformidade de um subfluxo de bits de um conjunto de camadas de saída para o perfil Estéreo Principal é indicada por `general profile ide` sendo igual a 5 ou `general profile compatibility flag[5]` sendo igual a 1.

- NOTA - Quando `general profile compatibility flag[5]` é igual a 1, `general profile compatibility flag[i]` para `i` igual a 1 deve ser também igual a 1.

[0179] Além disso, para implementar limites e restrições de nível específicos de divisão para perfis,

planos e níveis, as seguintes alterações podem ser para o Parágrafo G.11.2 de JCTVC-Q0145, que ele próprio modifica o parágrafo G.11.2 de Chen e outros, "High efficiency video coding (HEVC) scalable extensions Draft 5, Joint Collaborative Team on video coding (JCT-VC) of ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 16ª reunião: San José, EUA, 9 a 17 de janeiro de 2014, documento JCTVC-P1008_v4.

G.11.2 Planos e níveis

G.11.2.1 Limites gerais de plano e nível

Para fins de comparação de capacidades de plano, o plano com `general_tier_flag` igual a 0 é considerado como sendo um plano mais baixo que o plano com `general_tier_flag` igual a 1.

Para fins de comparação de capacidades de nível, um nível específico de um plano específico é considerado como sendo um nível mais baixo que algum outro nível do mesmo plano quando o valor do `general_level_id` do nível específico é menor que aquele do outro nível.

O que segue é especificado para expressão das limitações nessa subcláusula, subcláusula G.11.2.2, e subcláusula G.11.3:

- deixe a unidade de acesso n ser a n^{a} unidade de acesso em ordem de decodificação, com a primeira unidade de acesso sendo unidade de acesso 0 (isto é, a 0^{a} unidade de acesso).
- deixe a variável `fR` ser definida igual a $1 \div 300$.
- deixe a variável `optLsIdx` ser o índice de conjunto de camadas de saída de um conjunto de camadas de saída contido em um fluxo de bits, e o subfluxo de bits do conjunto de camadas de saída ser derivado por invocar o processo de extração

de sub-fluxo de bits como especificado na subcláusula F.10 com o fluxo de bits, TIdTarget igual a 7, e TargetDecLayerIdList contendo os valores nuh_layer_id do conjunto de camadas de saída como entradas.

- deixe a variável partitionIdx ser o índice para qualquer divisão específica do conjunto de camadas de saída com índice optLsIdx, e a divisão é derivada por invocar o processo de demultiplexagem como especificado na subcláusula C.6 com o subfluxo de bits, a lista de camadas na divisão ListOfLayersInPartition, e o número de camadas na divisão numLayersInPartition como entradas.

- Deixe as variáveis numLayersInOptLayerSet e numLayersInSubBitstream serem derivadas como a seguir:

```
numLayersInOptLayerSet =
NumLayersInIdList[LayerSetIdxForOutputLayerSet[optLsIdx]] se (a camada de base no conjunto de camadas de saída for fornecida por meio externo)
numLayersInSubBitstream = numLayersInOptLayerSet - 1 (H-63) ou
```

```
numLayersInSubBitstream = numLayersInOptLayerSet
```

- Deixe a variável partitionSizeInSamplesY ser derivada como a seguir:

```
partitionSizeInSamplesY = 0 (H-64)
para (i=0; i< numLayersInPartition; i+=)
partitionSizeInSamplesY += PicSizeInSamplesY
da ia camada na divisão
```

- deixe a variável ScaleFactor ser igual a $\text{Ceil}(\text{partitionSizeInSamplesY} / \text{MaxLumaPs})$, onde MaxLumaPs é especificado na tabela A-1

para o nível da divisão.

- Deixe a variável $\text{maxPartitionSliceSegs}$ ser definido igual a $\text{scaleFactor} * \text{MaxSliceSegmentsPerPicture}$, onde $\text{MaxSliceSegmentsPerPicture}$ é especificado na Tabela A-1 para o plano e nível da divisão. Cada divisão se conformando a um perfil em um plano e nível especificados obedecerá as seguintes limitações para cada teste de conformidade de fluxo de bits como especificado no Anexo A:

- a) Para cada camada na divisão, PicSizeInSamplesY será menor ou igual a MaxLumaPs , onde MaxLumaPs é especificado na Tabela A-1 para o plano e nível da divisão.
- b) Para cada camada na divisão, o valor de $\text{pic_width_in_luma_samples}$ será menor ou igual a $\text{Sqrt}(\text{MaxLumaPs} * 8)$.
- c) Para cada camada na divisão, o valor de $\text{pic_height_in_luma_samples}$ será menor ou igual a $\text{Sqrt}(\text{MaxLumaPs} * 8)$.
- d) Para cada camada com nuh_layer_id igual a currLayerId na divisão, o valor de $\text{Max_vps_dec_pic_buffering}[\text{optLsIdx}][\text{LayerIdxInVps}[\text{currLayerId}]][\text{HighestTid}]$ será menor ou igual a MaxDpbSize como derivado pela equação A-2, com PicSizeInSamplesY sendo substituído com $\text{partitionSizeinSamplesY}$, para o plano e nível da divisão.
- e) Para o nível 5 e níveis mais elevados, o valor de CtbSizeY para cada camada na divisão será igual a 32 ou 64.

- f) O valor de NumPicTotalCurr para cada imagem na divisão será menor ou igual a 8.
- g) Ao decodificar cada imagem codificada na divisão, o valor de num_tile_columns_minus1 será menor que MaxTileCols e num_tile_rows_minus1 será menor que MaxTileRows, onde MaxTileCols e MaxTileRows são especificados na Tabela A-1 para o plano e nível da divisão.
- h) Para os parâmetros VCL HRD da divisão, CpbSize[i] será menor ou igual a $1000 * \text{MaxCPB}$ para pelo menos um valor de i na faixa de 0 a cpb_ent_minus1[HighestTid], inclusive, onde CpbSize[i] é especificado na subcláusula E.3.3 com base em parâmetros selecionados como especificado na subcláusula C.1 e MaxCPB é especificado na Tabela A-1 para o plano e nível da divisão em unidades de 1000 bits.
- i) Para os parâmetros NAL HRD da divisão, CpbSize[i] será menor ou igual a $1100 * \text{MaxCPB}$ para pelo menos um valor de i na faixa de 0 a cpb_cnt_minus1 [HighestTid], inclusive, onde CpbSize[i] é especificado na subcláusula E.3.3 com base em parâmetros selecionados como especificado na subcláusula C.1 e MaxCPB é especificado na Tabela A-1 para o plano e nível da divisão em unidades de 1100 bits.
- a Tabela A-1 especifica os limites para cada nível de Cada plano.

Um plano e nível ao qual uma divisão se conforme são indicados pelos elementos de

sintaxe `general_tier_flag` e `general_leve_idc` como a seguir:

- `General_tier_flag` igual a 0 indica conformidade com o plano Principal, e `general_tier_flag` igual a 1 indica conformidade ao plano Elevado, de acordo com as limitações de plano especificadas na Tabela A-1. `General_tier_flag` será igual a 0 para níveis abaixo de nível 4 (correspondendo às entradas na Tabela A-1 marcadas com "-").
- `General_leve_idc` será definido igual a um valor de 30 vezes o número de nível especificado na Tabela A-1.

[0180] JCTVC-Q0145 inclui um anexo (JCTVC-Q0145_MvHvcText) que descreve alterações em Tech e outros, "MV-HEVC Draft Text 7", Joint Collaborative Team on 3d Video Coding Extension development of ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 7ª reunião: San José, EUA, 11 a 17 de janeiro de 2014, JCT3V-G1004-v7. O texto a seguir indica alterações em JCTVC-Q0145_MVHeveText para implementar limites e restrições de nível específicos de divisão para perfis, planos e níveis. Para concisão, texto deletado a partir de JCTVC-Q0145_MvHvcText é omitido abaixo.

G.11.2.2 Limites de nível específicos de perfil para os perfis Principal, Principal 10 e Estéreo principal

Cada divisão se conformando ao perfil Estéreo principal em um plano e nível especificados obedecerá às seguintes limitações para cada teste de conformação de fluxo de bits como especificado no Anexo C, onde "unidade de acesso" é usado para indicar o conjunto de imagens contidas na

divisão, juntamente com suas unidades não VCL NAL associadas, de camadas contidas na divisão e que têm o mesmo tempo de saída e o CPB é entendido como sendo o BPB:

- a) O tempo de remoção nominal de unidade de acesso n (com n maior que 0) a partir de CPB, como especificado na subcláusula C.2.3, atenderá a limitação que $AuNominalRemovalTime[n] - AuCpbRemovalTime[n - 1]$ é maior ou igual a $Max(\text{partitionSizeInSamplesY} \div \text{MaxLumaSr}, \text{fR})$, onde partitionSizeInSamplesY é o valor de partitionSizeInSamplesY para unidade de acesso $n - 1$ e MaxLumaSr é o valor especificado na Tabela A-2 que se aplica à unidade de acesso $n - 1$ para o plano e nível da divisão.
- b) A diferença entre tempos de saída consecutivos de imagens a partir do DPB, como especificado na subcláusula C.3.3, atenderá a limitação de que $DpbOutputInterval[n]$ é maior ou igual a $Max(\text{partitionSizeInSamplesY} \div \text{MaxLumaSr}, \text{fR})$, onde partitionSizeInSamplesY é o valor de partitionSizeInSamplesY da unidade de acesso n e MaxLumaSr é o valor especificado na Tabela A-2 para unidade de acesso n para o plano e nível da divisão, com a condição de que a unidade de acesso n seja uma unidade de acesso que tem uma imagem que é transmitida e não é a última de tais unidades de acesso.
- c) O tempo de remoção da unidade de acesso 0 atenderá a limitação de que o número de segmentos de fatia codificada na unidade de acesso 0 é menor ou igual a

$$\text{Min}(\text{Max}(\text{numLayersInPartition},$$

$$\frac{\text{MaxSliceSegmentsPerPicture}}{\text{MaxLumaPs}} * \text{MaxLumaSr} /$$

$$\text{MaxLumaPs} * (\text{AuCpbRemovalTime}[0] -$$

$$\text{AuNominalRemovalTime}[0]) +$$

$$\frac{\text{MaxSliceSegmentsPerpicture} * \text{partitionSizeIn}}{\text{SamplesY}} / \text{MaxLumaPs}),$$

maxSliceSegmentsPerPicture), onde

$$\text{MaxSliceSegmentsPerPicture}, \text{MaxLumaPs} \text{ e}$$

$$\text{MaxLumaSr}$$
 são os valores especificados na
Tabela A-1 e Tabela A-2 para o plano e nível
da divisão.

- d) A diferença entre tempos de remoção CPB consecutivos de unidades de acesso n e $n - 1$ (com n maior que 0) atenderá a limitação de que o número de segmentos de fatia codificada na unidade de acesso n é menor ou igual a
- $$\text{Min}((\text{Max}(\text{numLayersInPartition},$$

$$\frac{\text{MaxSliceSegmentPerPicture}}{\text{MaxLumaSr/MaxLumaPs}} * (\text{AuCpbRemovalTime}[n] -$$

$$\text{AuCpbRemovalTime}[n - 1])),$$

MaxSliceSegmentsPerpicture), onde

$$\text{numLayersInPartition}$$
 é o valor para a unidade de acesso n , e MaxSliceSegmentsPerPicture,
$$\text{maxLumaPs}, \text{ e } \text{MaxLumaSr}$$
 são os valores especificados na Tabela A-1 e Tabela A-2 que se aplicam a unidade de acesso n para o plano e nível da divisão.

- e) Para os parâmetros VCL HRD para a divisão,
$$\text{BitRate}[i]$$
 será menor ou igual a
$$\text{CpbVelFactor} * \text{MaxBR}$$
 para pelo menos um valor de i na faixa de 0 a $\text{cpb_cnt_minus1}[\text{HighestTid}]$, inclusive, onde
$$\text{BitRate}[i]$$
 é especificado na subcláusula E.3.3 com base em parâmetros selecionados como

especificado na subcláusula C.1 e MaxBR é especificado na Tabela A-2 em unidades de CpbBrVelFactor bits/s, onde CpbBrVelFactor é especificado na Tabela A-3 para o plano e nível da divisão.

- f) Para os parâmetros NAL HRD para a divisão, BitRate[i] será menor ou igual a CpbBrNalFactor * MaxBR para pelo menos um valor de i na faixa de 0 a cpb_cnt_minus1[HighestTid], inclusive, onde BitRate[i] é especificado na subcláusula E.3.3 com base em parâmetros selecionados como especificado na subcláusula C.1 e MaxBR é especificado na Tabela A-2 em unidades de CpbBrNalFactor bits/s, onde CpbBrNalFactor é especificado na Tabela A-3 para o plano e nível da divisão.
- g) A soma das variáveis NumBytesInNalunit para unidade de acesso 0 será menor ou igual a $\text{FormatCapabilityFactor} * (\text{Max}(\text{partitionSizeSamplesY}, \text{fR} * \text{MaxLumaSr}) + \text{MaxLumaSr} * (\text{AuCpbRemovalTime}[0] - \text{AuNominalRemovalTime}[0])) \div \text{MinCr}$, onde MaxLumaSr e MinCr são os valores especificados na Tabela A-2 e FormatCapabilityFactor é o valor especificado na tabela A-3 que se aplicam à imagem 0 para o plano e nível da divisão.
- h) A soma das variáveis NumBytesInNalUnit para unidade de acesso n (com n maior que 0) será menor ou igual a $\text{FormatCapabilityFactor} * \text{MaxLumaSr} * (\text{AuCpbRemovalTime}[n] - \text{AuCpbRemovalTime}[n - 1])$.

1]) $\div \text{MinCr}$, onde MaxLumasSr e MinCr são os valores especificados na tabela A-2 e $\text{FormatCapabilityFactor}$ é o valor especificado na Tabela A-3 que se aplicam à imagem n para o plano e nível da divisão.

i) O tempo de remoção da unidade de acesso 0 satisfará a limitação de que o número de azulejos em imagens codificadas na unidade de acesso 0 é menor ou igual a $\text{Min}(\text{Max}(\text{numLayersinPartition}, \text{MaxTileCols} * \text{MaxTileRows} * 120 * (\text{AuCpbRemovalTime}[0] - \text{AuNominalRemovalTime}[0]) + \text{MaxTileCols} * \text{MaxTiuleRows}), \text{MaxTileCols} * \text{MaxTileRows})$, onde MaxTileCols e MaxTileRows são os valores especificados na Tabela A-1 para o plano e nível da divisão.

j) A diferença entre tempos de remoção de CPB consecutivos de unidades de acesso n e $n - 1$ (com n maior que 0) atenderá a limitação de que o número de tiles em imagens codificadas na unidade de acesso n é menor ou igual a $\text{Min}(\text{Max}(\text{numLayersInPartition}, \text{MaxTileCols} * \text{MaxTileRows} * 120 * (\text{AuCpbRemovalRTIME}[n] - \text{AuCpbRemovalTime}[n - 1])), \text{MaxTileCols} * \text{MaxTileRows})$, onde $\text{numLayersinPartition}$ são os valores para a unidade de acesso n , e MaxTileCols e MaxTileRows são os valores especificados na Tabela A-1 que se aplicam à unidade de acesso n para o plano e nível da divisão.

[0181] Além disso, de acordo com alguns exemplos dessa revelação, as seguintes alterações podem ser feitas ao Parágrafo G.11.1.2 Perfil de Estéreo Principal de

MV-HEVC WD8. No texto abaixo, alterações são mostradas em relação à versão de exemplo do Parágrafo G.11.1.2 descrito acima nessa revelação.

G.11.1.2 Perfil de estéreo principal

Para um fluxo de bits contendo conjuntos de camadas de saída com divisões se conformando ao perfil de Estéreo principal, deixe

- olsIdx ser o índice de conjunto de camadas de saída de um conjunto de camadas de saída.

- layerSetIdx ser o índice de conjunto de camadas para o conjunto de camadas de saída.

- o subfluxo de bits do conjunto de camadas de saída ser derivado por invocar o processo de extração de subfluxo de bits conforme especificado na subcláusula F.10 com as seguintes entradas: o fluxo de bits, tIdTarget igual a 6, e layerIdListTarget contendo os valores nuh layer id das camadas que são incluídas no conjunto de camadas com o índice layerSetIdx e são camadas de imagem primária.

- o fluxo de bits de camada de base ser derivado por invocar o processo de extração de subfluxo de bits como especificado na subcláusula F.10 com o fluxo de bits, tIdTarget igual 6, e layerIdListTarget contendo somente um valor nuh layer id que é igual a 0 como entradas.

A [[Fluxos de bits contendo conjuntos de camadas de saída com divisões de camadas se conformando ao perfil Estéreo principal obedecerão às seguintes limitações no subfluxo de bits derivado para cada dos conjuntos de camadas de saída, com layerSetIdx sendo o índice do conjunto de camadas para o conjunto de camadas de saída, o subfluxo

de bits sendo derivado por invocar o processo de extração de subfluxo de bits conforme especificado na subcláusula F.10 com o fluxo de bits, `tIdTarget` igual a 7, e `TargetDecLayerLayerIdList` contendo os valores `nuh_layer_id` do conjunto de camadas com o índice `layerSetIdx` como entradas, as divisões sendo derivadas por invocar o processo de demultiplexagem para derivar uma divisão de fluxo de bits como especificado na subcláusula C.6 com o subfluxo de bits, a lista de camadas na divisão `listOfLayersInPartition`, e o número de camadas na divisão `numLayersInPartition` como entradas `B[]` fluxo de bits `[[s correspondendo a subfluxos de bits para]]` contendo conjuntos de camadas de saída com divisões `[[de camadas]]` se conformando ao perfil Estéreo Principal obedecerão as seguintes limitações:

- todos VPSs ativos terão `vps_base_layer_internal_flag` igual a 1 somente.
- todos VPSs ativos terão `vps_num_rep_formats_minus1` na faixa de 0 a 15, inclusive.
- `[[Fluxos de bits contendo conjuntos de camadas de saída se conformando ao perfil Estéreo Principal obedecerão às seguintes limitações no subfluxo de bits derivado para cada dos conjuntos de camadas de saída, com índice de conjunto de camadas de saída olsIdx e layerSetIdx sendo o conjunto de camadas para o conjunto de camadas de saída, e o subfluxo de bits sendo derivado por invocar o processo de extração de subfluxo de bits como especificado na subcláusula F.10 com as`

seguintes entradas: o fluxo de bits, *tIdTarget* igual a 6, e *layerIdListTarget* contendo os valores *nuh_layer_id* das camadas que são incluídas no conjunto de camadas com o índice *layerSetIdx* e são camadas de imagem primária.]]

[[Fluxos de bits contendo conjuntos de camadas de saída se conformando ao perfil Estéreo principal obedecerão também às seguintes limitações no fluxo de bits de camada base derivado por invocar o processo de extração de subfluxo de bits conforme especificado na subcláusula F.10 com o fluxo de bits, *tIdTarget* igual a 6, e com *layerIdListTarget* contendo somente um valor *nuh_layer_id* que é igual a 0 como entradas.]]

O fluxo de bits de camada base [[derivado de fluxos de bits se conformando ao perfil Estéreo principal]] obedecerá as seguintes limitações:

- o fluxo de bits de camada base obedecerá a todas as limitações do perfil Principal especificado na subcláusula A.3.2.
- todos os SPSs ativos [[de]] para a camada base [[fluxo de bits]] terão *general_profile_idc* igual a 1 ou *general_profile_compatibility_flag[1]* igual a 1.

O [[derivado]] subfluxo de bits de cada [[para um]] conjunto de camadas de saída [[se conformando ao perfil Estéreo principal]] obedecerá às seguintes limitações:

- todos SPSs ativos para o subfluxo de bits terão *chroma_format_idc* igual a 1 apenas.
- *CtbLog2SizeY* derivado de qualquer SPS ativo [[de]] para o subfluxo de bits estará na faixa de 4 a 6, inclusive.

- ScalabilityId[j][smIdx] será igual a 0 para qualquer valor smIdx não igual a 1 e para qualquer valor de j para o subfluxo de bits.

No restante dessa subcláusula e subcláusula G.11.3, para um subfluxo de bits de um conjunto de camadas de saída, general_profile_idc, general_profile_compatibility_flag[i], general_tier_flag, e general_level_idc se referem ao general_profile_idc, general_profile_compatibility_flag[i], general_tier_flag e general_level_idc, respectivamente, na estrutura de sintaxe de profile_tier_level() para a *[[conjunto de camadas de saída]]* divisão.

Conformidade de um subfluxo de bits de uma *[[conjunto de camadas de saída]]* divisão ao perfil Estéreo principal é indicada por general_profile_idc sendo igual a 6[[5]] ou general_profile_compatibility_flag[6[[5]]] sendo igual a 1.

- NOTA - quando general_profile_compatibility_flag[6[[5]]] é igual a 1, general_profile_compatibility_flag[1[[i]]] *[[para i igual a 1]]* deve ser também igual a 1.

[0182] Além disso, em alguns exemplos dessa revelação, as seguintes alterações podem ser feitas no parágrafo G.11.2.1 de MV-HEVC WD8. O texto abaixo indica alterações em uma porção da versão de exemplo do parágrafo G.11.2.1 descrito acima nessa revelação.

G.11.2.1 Limites gerais de plano e nível

A condição (a) é alterada como a seguir:

j) *[[Para cada camada no]]* O valor de divisão

[[,]] PicSizeInSamplesY será menor ou igual a MaxLumaPs, onde MaxLumaPs é especificado na tabela A-1 para o plano e nível da divisão.

[0183] Adicionalmente, em alguns exemplos dessa revelação, as seguintes alterações são feitas no parágrafo G.11.2.2 de MV-HEVC WD8. O texto abaixo indica alterações em uma porção da versão de exemplo do Parágrafo G.11.2.1 descrito acima nessa revelação.

G.11.2.2 Limites de nível específico de perfil para os Perfis Principal, Principal 10 e Estéreo Principal

Cada divisão se conformando ao perfil Estéreo principal em um plano e nível especificados obedecerá as seguintes limitações para [[cada fluxo de bits]] o teste de conformidade conforme especificado no Anexo C, onde “unidade de acesso” é usada para indicar o conjunto de imagens contidas na divisão, juntamente com suas unidades não VCL NAL associadas, de camadas contidas na divisão e que têm o mesmo tempo de saída e o CPB é entendido como sendo o BPB:

[0184] Além disso, como descrito em outro lugar de acordo com uma ou mais técnicas dessa revelação, para cada esquema de divisão respectivo de cada conjunto de camadas de saída, uma mensagem VPS VUI ou SEI de um fluxo de bits pode identificar um conjunto de parâmetros HRD para o esquema de divisão respectivo. As alterações mostradas na Tabela 4 podem ser feitas na estrutura de sintaxe de parâmetros HRD de divisão de fluxo de bits VPS VUI definida no Parágrafo F.7.3.2.1.6 de MV-HEVC WD8 e SHVC WD6 para implementar tais exemplos.

Tabela 4

vps_vui_bsp_hrd_parameters(){	Descritor
vps_num_bsp_hrd_parameters_minus1	ue(v)
for(i = 0; i <= vps_num_bsp_hrd_parameters_minus1; i++) {	
if(i > 0)	
bsp_cprms_present_flag[i]	u(1)
hrd_parameters(bsp_cprms_present_flag[i], vps_max_sub_layers_minus1)	
}	

for(h = 1; h <= <u>NumOutputLayerSets</u> [[vps_num_layer_sets_minus1]]; h++) {	
[[num_bitstream_partitions [h]]]	[[ue(v)]]
[[for(i = 0; i < num_bitstream_partitions[h]; i++)]]	
[[for(j = 0; j < NumLayersInIdList[h]; j++)]]	
[[layer_in_bsp_flag [h][i][j]]]	[[u(1)]]
for(i = 1; i <= num_partitioning_schemes_minus1[h]; i++) {	
[[if(num_bitstream_partitions[h] > 0) {]]	
num_bsp_sched_combinations_minus1 [h][i]	ue(v)
for([[i]] <u>i</u> = 0; [[i]] <u>i</u> <= num_bsp_sched_combinations_minus1[h][i]; [[i]] <u>i</u> ++)	
for([[j]] <u>k</u> = 0; [[j]] <u>k</u> < [[n]] <u>Num</u> [[p]] <u>Partitions</u> [[i]] <u>In</u> [[s]] <u>Scheme</u> [[minus1 num_bitstream_partitions]][h][i]; [[j]] <u>k</u> ++) {	
bsp_comb_hrd_idx [h][i][j][k]	ue(v)
bsp_comb_sched_idx [h][i][j][k]	ue(v)
}	
}	
}	
}	

[0185] De acordo com o exemplo da tabela 4, as seguintes alterações podem ser feitas na semântica dos parâmetros HRD de divisão de fluxo de bits VPS VUI como

definido nos parágrafos F.7.4.3.1.6 de MV-HEVC WD8 e SHVC WD6.

[[num_bitstream_partitions[h] especifica o número de divisões de fluxo de bits para os quais parâmetros HRD são especificados para o conjunto de camadas com índice h.

Layer_in_bsp_flag[h][i][j] especifica que a camada com índice j é uma parte da divisão de fluxo de bits com índice i no conjunto de camadas com índice h.]]

Num_bsp_sched_combinations_minus1[h][i] plus 1 especifica o número de combinações de programas de fornecimento e hrd_parameters() especificados para divisões de fluxo de bits para o iº esquema de divisão do hº conjunto de camadas de saída [[com índice h]].

A variável SchedCombCnt[h][i] é definida igual a num_bsp_sched_combinations_minus[h][i] +1.

Bsp_comb_hrd_idx[h][i][j][k] especifica o índice de hrd_parameters() na estrutura de sintaxe vps_vui_bsp_hrd_parameters() usada na [[i]]jª combinação de um programa de fornecimento e hrd_parameters() especificado para a [[fluxo de bits]]kª divisão [[com índice j e]] do iº esquema de divisão para o hº conjunto de camadas de saída [[com índice h]].

Bsp_comb_sched_idx[h][i][j][k] especifica o índice de um programa de fornecimento na estrutura de sintaxe hrd_parameters() com o índice bsp_comb_hrd_idx[h][i][j][k] que é usado na [[i]]jª combinação de um programa de fornecimento e hrd_parameters() especificado para o [[fluxo de bits]]-kª divisão [[com índice j]]

do iº esquema de divisão *[[e]]* para o hº conjunto de camadas de saída *[[com índice h]]*.

[0186] Além disso, de acordo com uma ou mais técnicas dessa revelação, as alterações mostradas na tabela 5, abaixo, podem ser feitas na sintaxe de mensagem SEI de parâmetros HRD de divisão de fluxo de bits como definido no parágrafo F.14.1.5 de MV-HEVC WD8 e SHVC WD6.

Tabela 5

<code>bsp_hrd(payloadSize) {</code>	Descritor
<code> sei_num_bsp_hrd_parameters_minus1</code>	<code>uc(v)</code>
<code> for(i = 0; i <= sei_num_bsp_hrd_parameters_minus1; i++) {</code>	
<code> if(i > 0)</code>	
<code> sei_bsp_cprms_present_flag[i]</code>	<code>u(1)</code>
<code> hrd_parameters(sei_bsp_cprms_present_flag[i],</code> <code> nesting_max_temporal_id_plus1[0] - 1)</code>	

}	
for(h = 0; h <= nesting_num_ops_minus1; h++) {	
olsIdx = nesting_op_idx[h]	
<u>num_sei_partitioning_schemes_minus1[olsIdx]</u>	<u>ue(v)</u>
for(i = 1; i <= num_sei_partitioning_schemes_minus1[olsIdx]; i++) {	
<u>_num_sei_bitstream_partitions_minus1[1][2][olsIdx][i]</u>	<u>ue(v)</u>
__for([[i]] j = 0; [[i]] j <=	
[[n]]Num[[s]]Sei[[b]]Bitstream[[p]]Partitions[[minus1]][[olsIdx][i];	
j++)	
__for([[j]] k = 0; [[j]] k <= vps_max_layers_minus1; [[j]] k++)	
__if(	
layer_id_included_flag[LayerSetIdxForOutputLayerSet[olsIdx]][[[layer_id	
_in_nuh[j]] k []])	
<u>_sei_layer_in_bsp_flag[olsIdx][i][j][k]</u>	<u>u(1)</u>
<u>_sei_num_bsp_sched_combinations_minus1[olsIdx][i]</u>	<u>ue(v)</u>
__for([[i]] j = 0; [[i]] j <=	
sei_num_bsp_sched_combinations_minus1[olsIdx][i]; [[i]] j++)	
__for([[j]] k = 0; [[j]] k <=	
[[n]]Num[[s]]Sei[[b]]Bitstream[[p]]Partitions[[minus1]][[olsIdx][i];	
[[j]] k++) {	
<u>_sei_bsp_comb_hrd_idx[olsIdx][i][j][k]</u>	<u>ue(v)</u>
<u>_sei_bsp_comb_sched_idx[olsIdx][i][j][k]</u>	<u>ue(v)</u>
__}	
__}	
}	
}	

[0187] Além disso, no exemplo da tabela 5, a divisão de fluxo de bits que encaixa a mensagem SEI fornece um mecanismo para associar mensagens SEI com uma divisão de fluxo de bits de um conjunto de camadas de saída. No exemplo da tabela 5, as seguintes alterações na semântica de mensagem SEI de parâmetros HRD de divisão de fluxo de

bits podem ser feitas ao parágrafo F.14.2.5 de MV-HEVC WD8 e SHVC WD6.

Num_sei_partitioning_schemes_minus1[olsIdx]plus 1 especifica o número de esquemas de divisão especificados para o conjunto de camadas de saída com índice olsIdx. O valor de num_sei_partitioning_schemes_minus1[olsIdx] estará na faixa de 0 a 15, inclusive.

Num_sei_bitstream_partitions_minus2[[1]][olsIdx][i]plus[[1]]2 especifica o número de divisões de fluxo de bits para as quais parâmetros HRD são especificados para o i° esquema de divisão do conjunto de camadas de saída com índice olsIdx. O valor de num_sei_bitstream_partitions_minus1[olsIdx][i] estará na faixa de 0 a NumLayersInIdList[LayerSetIdxForOutputLayerSet[olsIdx]]-1, inclusive.

O valor de num_sei_bitstream_partitions_minus1[olsIdx][0] é inferido como sendo igual a 0.

O valor da variável NumPartitionInScheme[i][j] é derivado como a seguir:

NumSeiBitstreamPartitions[olsIdx][i] = (i == 0) ? 1:

Num_sei_bitstream_partitions_minus2[i][j] + 2.

Sei_layer_in_bps_flag[olsIdx][i][j][k] especifica que a camada com índice [[j]]k é uma parte de divisão de fluxo de bits com índice [[i]]j no esquema de divisão com índice i do conjunto de camadas com índice olsIdx. O valor de layer_included_in_partition_flag[olsIdx][0][0][k] é inferido como sendo igual ao valor de

layer id included flag[LayerSetIdxForOutputLayerSet[olsIdx][k] para k na faixa de 0 a vps_max_layers_minus1, inclusive.

É uma exigência de conformidade de fluxo de bits que a divisão de fluxo de bits com índice $[[j]]$ não incluirá camadas de referência direta ou indireta de quaisquer camadas na divisão de fluxo de bits com índice $[[i]]$ m para quaisquer valores de $[[i]]$ m e $[[j]]$ n na faixa de 0 a $N[[n]]um[[s]]Sei[[b]]Bitstream[[p]]Partitions[[minus1]][olsIdx][i][, inclusive]$, de modo que $[[i]]$ m seja menor que $[[j]]$ n.

Sei_num_bsp_sched_combinations_minus1[olsIdx][i]p lus 1 especifica o número de combinações de programas de fornecimento e hrd_parameters() especificados para divisões de fluxo de bits para o esquema de divisão com índice i do conjunto de camadas de saída com índice olsIdx.

A variável SchedCombCnt[olsIdx][i] é definida igual a

Sei_num_bsp_sched_combinations_minus1[olsIdx][i] + 1.

Sei_bsp_comb_hrd_idx[olsIdx][i][j][k] especifica o índice de hrd_parameters() nessa mensagem SEI usada na $[[i]]j^a$ combinação de um programa de fornecimento e hrd_parameters() especificado para a divisão de fluxo de bits com índice $[[j]]k$ do esquema de divisão com índice i $[[e]]$ para o conjunto de camadas de saída com índice olsIdx.

Sei_bsp_comb_sched_idx[olsIdx][i][j][k] especifica o índice de um programa de fornecimento na estrutura de sintaxe hrd_parameters() com o índice

sei_bsp_comb_hrd_idx[olsIdx][i][j][k] que é usado na $[[i]]j^a$ combinação de um programa de fornecimento e hrd_parameters() especificado para a divisão de fluxo de bits com índice $[[j]]k$ do esquema de divisão com índice i $[[e]]$ para o conjunto de camadas de saída com índice olsIdx.

[0188] O Anexo C em MV-HEVC WD8 e SHVC WD6 descreve o HRD. Mais particularmente, o parágrafo C.1 de MV-HEVC WD8 e SHVC WD6 indica que múltiplos testes podem ser necessários para verificar a conformidade de um fluxo de bits, que é mencionado como o fluxo de bits em teste. Além disso, o parágrafo C.1 de MV-HEVC WD8 e SHVC WD6 indica uma série de etapas para aplicar para verificar a conformidade de um fluxo de bits.

[0189] Em alguns exemplos dessa revelação, a série de etapas descritas no parágrafo C.1 de MV-HEVC WD8 e SHVC WD6 é modificada para suportar o esquema de divisão descrito em outro lugar nessa revelação. Por exemplo, as seguintes modificações podem ser feitas no parágrafo C.1 de MV-HEVC WD8 e SHVC WD6.

Para cada teste, as seguintes etapas se aplicam na ordem listada:

1. Um ponto de operação em teste, indicado como TargetOp, é selecionado por selecionar um valor para TargetOutputLayerSetIdx identificando um conjunto de camadas de saída alvo e selecionando um valor TemporalId mais elevado alvo HighestTid. O valor de TargetOutputLayerSetIdx estará compreendido na faixa de 0 a NumOutputLayerSets - 1, inclusive. O valor de HighestTid estará na faixa de 0 a MaxSubLayersInLayerSetMinus1[LayerSetIdxForOutputLayerSet[Target

OutputLayerSetIdx]], inclusive. As variáveis TargetDecLayerSetIdx, TargetOptLayerIdList e TargetDecLayerIdList são então derivadas como especificado pela equação 8-1. O ponto de operação em teste tem OptLayerIdList igual a TargetOptLayerIdList, OpLayerIdList igual a TargetDecLayerIdList e OpTid igual a HighestTid.

2. O processo de extração de subfluxo de bits como especificado na cláusula 10 é invocado com o fluxo de bits em teste, HighestTid, e TargetDecLayerIdList como entradas, e a saída é atribuída a BitstreamToDecode. Um esquema de divisão é selecionado a partir da lista de esquemas de divisão sinalizados no VPS ativo para o conjunto de camadas de saída selecionado. Após isso, para o esquema de divisão selecionado, uma divisão é selecionada a partir da lista de divisões sinalizadas no VPS para o conjunto de camadas de saída selecionadas.
3. Se a divisão selecionada contiver todas as camadas do conjunto de camadas de saída selecionadas, a operação CPB específica de fluxo de bits é selecionada. De outro modo, a operação CPB específica de divisão de fluxo de bits é selecionada. As etapas subsequentes se aplicam ao subfluxo de bits da divisão selecionada, mencionada como a divisão de fluxo de bits em teste. [[Quando a estrutura de sintaxe vps_vui_bsp_hrd_parameters() está presente no VPS ativo, um esquema de divisão, indicado como TargetPartitioningScheme, é

selecionado por selecionar um valor para `TargetPartitioningSchemeIdx` que está na faixa de 1 a `num_partitioning_schemes_minus1[TargetOutputLayerSetIdx]`, inclusive. Quando `num_bitstream_partitions_minus1[TargetDecOutputLayerSetIdx][TargetPartitioningSchemeIdx]` é maior que 0 onde `TargetPartitioningSchemeIdx` é o índice do `TargetPartitioningScheme` selecionado que está compreendido na faixa de 1 a `num_partitioning_schemes_minus1[TargetOutputLayerSetIdx]` ou uma mensagem SEI de parâmetros HRD de divisão de fluxo de bits está presente e a mensagem SEI contém elemento de sintaxe `num_sei_bitstream_partitions_in_scheme` `bitstream_partitions_minus1[TargetDecOutputLayerSetIdx][TargetPartitioningSchemeIdx]` maior que 0, a operação CPB específica de fluxo de bits ou a operação CPB específica de divisão de fluxo de bits é selecionada para um teste de conformidade, e ambas as operações CPB serão testadas para verificar a conformidade de um fluxo de bits. Quando a operação CPB específica de fluxo de bits é testada, as etapas subsequentes se aplicam para o fluxo de bits em teste. Quando a operação CPB específica de divisão de fluxo de bits é testada, as etapas subsequentes se aplicam a cada divisão de fluxo de bits do fluxo de bits em teste, mencionada como a divisão de fluxo de bits em teste. Quando a operação CPB específica de divisão de fluxo de bits é testada e a entrada para o HRD é um fluxo de

bits, as divisões de fluxo de bits são]] derivadas com o processo de demultiplexagem para derivar uma divisão de fluxo de bits na subcláusula C.6. Quando a operação CPB específica de divisão de fluxo de bits é selecionada, as divisões de fluxo de bits de outras divisões do mesmo esquema de divisão do conjunto de camada de saída selecionada são assumidas como sendo decodificadas por decodificar capacidades diferentes de HRD, porém disponíveis para o HRD como programado de acordo com os parâmetros HRD daquelas divisões.

4. A estrutura de sintaxe `hrd_parameters()` e a estrutura de sintaxe `sub_layer_hrd_parameters()` aplicáveis a `TargetOp` são selecionadas como a seguir:

Se a operação CPB específica de fluxo de bits for testada, o seguinte se aplica:

- se `TargetDecLayerIdList` contiver todos os valores `nuh_layer_id` presentes no fluxo de bits em teste, a estrutura de sintaxe `hrd_parameters()` no SPS ativo para a camada base (ou fornecida através de um meio externo não especificado nesse Relatório descritivo) é selecionada.
- de outro modo, a estrutura de sintaxe `hrd_parameters()` no VPS ativo (ou fornecido através de algum meio não especificado nesse Relatório descritivo) que se aplica a `TargetOp` é selecionado.
- de outro modo, a estrutura de sintaxe `hrd_parameters()` é selecionada como a seguir:
- qualquer uma das estruturas de sintaxe

hrd_parameters() nas seguintes condições pode ser selecionada, se ambas as seguintes condições forem verdadeiras:

- a estrutura de sintaxe vps_vui_bsp_hrd_parameters() está presente no VPS ativo (ou está disponível através de algum meio externo não especificado nesse Relatório descritivo) e contém uma estrutura de sintaxe hrd_parameters() que se aplica a TargetOp, para TargetPartitioningScheme e à divisão de fluxo de bits em teste.
- uma mensagem SEI de parâmetros HRD de divisão de fluxo de bits que é incluída em uma mensagem SEI de encaixe escalonável que se aplica a TargetOp e contém uma estrutura de sintaxe hrd_parameters() que se aplica a TargetOp, para TargetPartitioningScheme e à divisão de fluxo de bits em teste está presente (ou é disponível através de algum meio externo não especificado nesse Relatório descritivo).
- de outro modo, se a estrutura de sintaxe vps_vui_bsp_hrd_parameters() estiver presente no VPS ativo (ou estiver disponível através de algum meio externo não especificado nesse Relatório descritivo) e contiver uma estrutura de sintaxe hrd_parameters() que se aplica a TargetOp, a TargetPartitioningScheme e a divisão de fluxo de bits em teste, aquela estrutura de sintaxe hrd_parameters() é selecionada.
- de outro modo, uma estrutura de sintaxe hrd_parameters() que se aplica à divisão de fluxo de bits em teste na mensagem SEI de parâmetros HRD de divisão de fluxo de bits que é incluída em

uma mensagem SEI de encaixe escalonável que se aplica a TargetOp e a TargetPartitioningScheme estará presente (ou estará disponível através de algum meio externo não especificado nesse Relatório descritivo) e é selecionada.

Na estrutura de sintaxe de hrd_parameters() selecionada, se BitstreamToDecode for um fluxo de bits Tipi 1, a estrutura de sintaxe sub_layer_hrd_parameters(HighestTid) que imediatamente segue a condição "se (vcl_hrd_parameters_present_flag)" for selecionada e a variável NalHrdModeFlag for definida igual a 0; de outro modo (BitstreamToDecode é um fluxo de bits tipo II), a estrutura de sintaxe sub_layer_hrd_parameters(HighestTid) que imediatamente segue a condição "se (vcl_hrd_parameters_present_flag)" (nesse caso a variável NalHrdModeFlag é definida igual a 0) ou a condição "se (nal_hrd_parameters_present_flag)" (nesse caso a variável NalHrdModeFlag é definida igual a 1) é selecionada. Quando BitstreamToDecode for um fluxo de bits do tipo II e NalHrdModeFlag for igual a 0, todas as unidades não VCL NAL exceto as unidades NAL de dados de enchimento, e todos os elementos de sintaxe leading_zero_8bits, zero_byte, start_code_prefix_one_3bytes, e trailing_zero_8bits que formam um fluxo de bits a partir do fluxo de unidade NAL (como especificado no anexo B), quando presente, são descartados do BitstreamToDecode, e o fluxo de bits restante é atribuído a BitstreamToDecode.

5. Uma unidade de acesso associada a uma mensagem SEI de período de armazenagem (presente em BitstreamToDecode ou disponível através de meio externo não especificado nesse Relatório descritivo) aplicável a TargetOp é selecionada como o ponto de inicialização HRD e mencionada como unidade de acesso 0. Uma mensagem SEI de período de armazenagem aplicável é disponível através de meio externo não especificado nesse Relatório descritivo ou é selecionado da unidade de acesso 0 como a seguir:

- se a operação CPB específica de fluxo de bits for testada, o seguinte se aplica:
- se TargetDecLayerIdList contiver todos os valores nuh_layer_id presentes no fluxo de bits em teste, uma mensagem SEI de período de armazenagem não encaixada é selecionada.
- de outro modo, uma mensagem SEI de período de armazenagem incluída na mensagem SEI de encaixe escalonável com bitstream_subset_flag igual a 1 e aplicável a TargetOp é selecionada.
- de outro modo, uma mensagem SEI de período de armazenagem incluída na mensagem SEI de encaixe de divisão de fluxo de bits aplicável à divisão de fluxo de bits em teste é selecionada.

A variável MultiLayerCpbOperationFlag é derivada como a seguir:

- se a mensagem SEI de período de armazenagem selecionada for não encaixada ou for incluída em uma mensagem SEI de encaixe escalonável que se aplica somente ao subfluxo de bits que

contém somente a camada base, MultiLayerCpbOperationFlag é definido igual a 0.

- de outro modo, MultiLayerCpbOperationFlag é definido igual a 1.

6. Para cada unidade de acesso em BitstreamToDecode iniciando a partir da unidade de acesso 0, a mensagem SEI de período de armazenagem (presente em BitstreamToDecode ou disponível através de meio externo não especificado nesse relatório descritivo) que é associado à unidade de acesso e se aplica a TargetOp é selecionada, a mensagem SEI de temporização de imagem (presente em BitstreamToDecode ou disponível através de meio externo não especificado nesse relatório descritivo) que é associado à unidade de acesso e se aplica a TargetOp é selecionada, e quando SubPicHrdFlag é igual a 1 e sub_pic_cpb_params_in_pic_timing_sei_flag é igual a 0, as mensagens SEI de informação de unidade de decodificação (presentes em BitstreamToDecode ou disponível através de meio externo não especificado nesse Relatório descritivo) que são associados a unidades de decodificação na unidade de acesso e se aplicam a TargetOp são selecionadas como a seguir:

- se a operação CPB específica de fluxo de bits for testada, o seguinte se aplica:

- se TargetDecLayerIdList contiver todos os valores nuh_layer_id presente no fluxo de bits em teste, mensagens SEI de informação de

unidade de decodificação, temporização de imagem e período de armazenagem não encaixado, são selecionadas.

- de outro modo, mensagens SEI de informação de período de armazenagem, temporização de imagem e unidade de decodificação incluídas na mensagem SEI de encaixe escalonável com `bitstream_subset_flag` igual a 1 e aplicável a `TargetOp` são selecionadas.

- de outro modo, mensagens SEI de informação de período de armazenagem, temporização de imagem e unidade de decodificação incluídas na mensagem SEI de encaixe de divisão de fluxo de bits e aplicáveis à divisão de fluxo de bits em teste são selecionadas.

7. um valor de `SchedSelIdx` é selecionado como a seguir:

- se a operação CPB específica de fluxo de bits for testada, `SchedSelIdx` selecionado estará na faixa de 0 a `cpb_cnt_minus1[HighestTid]`, inclusive, onde `cpb_cnt_minus1[HighestTid]` é encontrado na estrutura de sintaxe `sub_layer_hrd_parameters(HighestTid)` como selecionado acima.

- de outro modo (a operação CPB específica de divisão de fluxo de bits é testada), um `SchedSelCombIdx` é selecionado para o fluxo de bits em teste e usado para cada divisão de fluxo de bits em teste. O seguinte se aplica:

- se a estrutura de sintaxe `vps_vui_bsp_hrd_parameters()` estiver presente no VPS ativo (ou tornado disponível através de

meio externo não especificado nesse relatório descritivo) e contiver a estrutura de sintaxe `hrd_parameters()` selecionada que se aplica a `TargetOp` e a divisão de fluxo de bits em teste, o `SchedSelCombIdx` selecionado estará na faixa de 0 a `num_bsp_sched_combinations_minus1` `TargetOutput` `[[Dec]]LayerSetIdx` `[TargetPartitioningSchemeIdx]`, inclusive, e o `SchedSelIdx` selecionado será igual a `bsp_comb_sched_idx` `TargetOutput` `[[Dec]]LayerSetIdx` `[TargetPartitioningSchemeIdx]` `[SchedSelCombIdx]` `[j]` onde `j` é o índice da divisão de fluxo de bits em teste, onde `TargetPartitioningSchemeIdx` é o índice do `TargetPartitioningScheme` selecionado que está na faixa de 1 a `num_partitioning_schemes_minus1` `TargetOutputLayerSetIdx`.

- de outro modo, o `SchedSelCombIdx` selecionado estará na faixa de 0 a `sei_num_bsp_sched_combinations_minus1` `TargetOuptut` `[[Dec]]LayerSetIdx` `[TargetPartitioningSchemeIdx]`, inclusive, e o `SchedSelIdx` selecionado será igual a `sei_bsp_comb_sched_idx` `TargetOutput` `[[Dec]]LayerSetIdx` `[TargetPartitioningSchemeIdx]` `[SchedSelCombIdx]` `[j]` da mensagem SEI de parâmetros HRD de divisão de fluxo de bits aplicável a `TargetOp` onde `j` é o índice da divisão de fluxo de bits em teste.

O HRD opera como a seguir:

-O HRD é inicializado na unidade de decodificação 0, com o CPB, cada sub-DPB do DPB, e cada BPB sendo definido como sendo vazio (a plenitude de sub-DPB para cada sub-DPB é definida igual a 0).

NOTA 6 - após inicialização, o HRD não é inicializado novamente por mensagens SEI de período de armazenagem subsequentes.

- para a operação CPB específica de fluxo de bits, dados associados a unidades de decodificação que fluem para dentro do CPB de acordo com um programa de chegada especificado são fornecidos pelo HSS. Para a operação de CPB específica de divisão de fluxo de bits, dados associados a unidades de decodificação que fluem para dentro do BPB de acordo com um programa de chegada especificado são fornecidos por um HBPS.

- quando a operação CPB específica de divisão de fluxo de bits é usada, cada divisão de fluxo de bits com índice j é processado como especificado na clausula C.2 com o HSS substituído pelo HPBS e com SchedSelIdx igual a

```

bsp_comb_sched_idx[
TargetOutput[[Dec]]LayerSetIdx]
[TargetPartitioningSchemeIdx][SchedSelCombIdx]
[j], se a estrutura de sintaxe
vps_vui_bsp_hrd_parameters() estiver presente
no VPS ativo ou estiver disponível através de
algum meio externo não especificado nesse
relatório descritivo, ou igual a
sei_bsp_com_sched_idx[TargetOutput[[Dec]]
LayerSetIdx][TargetPartitioningSchemeIdx]
```

[SchedSelCombIdx][j] da mensagem SEI de parâmetros HRD de divisão de fluxo de bits aplicável a TargetOp, de outro modo.

- os dados associados a cada unidade de decodificação são removidos e decodificados instantaneamente pelo processo de decodificação instantâneo no tempo de remoção de CPB da unidade de decodificação.
- cada imagem decodificada é colocada no DPB.
- uma imagem decodificada é removida do DPB quando não mais se torna necessária para referência inter predição e não mais necessária para transmissão.

[0190] Como mostrado acima, quando um HRD testa cada divisão de cada esquema de divisão, com base em uma divisão selecionada de um esquema de divisão específico de um ou mais esquemas de divisão não contendo todas as camadas do conjunto de camadas de saída, o HRD pode aplicar, uma operação de buffer de imagem codificada específica de divisão (CPB). Por exemplo, o HRD pode obter, a partir de uma estrutura de sintaxe de parâmetros de programação de fluxo de bits (BSP) no VUI, um ou mais conjuntos de parâmetros de HRD. Em um exemplo similar, o HRD pode obter de uma estrutura de sintaxe BSP na mensagem SEI, um ou mais conjuntos de parâmetros HRD. Para cada esquema de divisão respectivo de um ou mais esquemas de divisão do conjunto de camadas de saída, para cada combinação respectiva em um conjunto de uma ou mais combinações de programa para o esquema de divisão respectivo, para cada divisão respectiva do esquema de divisão respectiva, o HRD pode obter, a partir da estrutura de sintaxe BSP, um elemento de sintaxe respetivo (por exemplo, bsp_comb_hrd_idx) especificando um conjunto de

parâmetros HRD para a divisão respectiva na estrutura de sintaxe BSP e um elemento de sintaxe respectivo especificando um programa de fornecimento (por exemplo, sei_bsp_comb_sched_idx).

[0191] Em exemplos correspondentes, o codificador de vídeo 20 inclui, em uma estrutura de sintaxe BSP no VUI, um ou mais conjuntos de parâmetros HRD. Para cada combinação respectiva em um conjunto de uma ou mais combinações de programa para um esquema de divisão respectivo, para cada divisão respectiva do esquema de divisão específico, o codificador de vídeo 20 pode incluir, na estrutura de sintaxe BSP, um elemento de sintaxe respectivo especificando um conjunto de parâmetros HRD para a divisão respectiva na estrutura de sintaxe BSP e um elemento de sintaxe respectivo especificando um programa de fornecimento. Em outro exemplo, o codificador de vídeo 20 inclui, em uma estrutura de sintaxe BSP na mensagem SEI, um ou mais conjuntos de parâmetros HRD. Para cada combinação respectiva em um conjunto de uma ou mais combinações de programa para um esquema de divisão respectivo, para cada divisão respectiva do esquema de divisão respectivo, o codificador de vídeo 20 inclui, na estrutura de sintaxe BSP, um elemento de sintaxe respectivo especificando um conjunto de parâmetros HRD para a divisão respectiva na estrutura de sintaxe BSP e um elemento de sintaxe respectivo especificando um programa de fornecimento.

[0192] Além disso, em alguns exemplos, o parágrafo C.4 de MV-HEVC WD8 e SHVC WD6 é modificado como a seguir:

A seguinte sentença é removida:

O fluxo de bits é testado pelo HRD para conformidade como especificado na subcláusula C.1.

A seguinte sentença é adicionada:

O subfluxo de bits de cada divisão de cada esquema de divisão é testado pelo HRD em relação à conformidade como especificado na subcláusula C.1.

[0193] Como indicado acima, um dispositivo de processamento de vídeo pode ser capaz de determinar se um decodificador de vídeo de multicamadas é capaz de decodificar um fluxo de bits com base em uma lista de tripletos de perfil, plano e nível especificados para o decodificador de multicamadas. O seguinte texto é um exemplo de como isso pode ser implementado.

G.11.3 Capacidades de decodificadores

Decodificadores se conformando a um conjunto de tripletos de perfil, plano e nível indicados por (P_i, T_i, L_i) para i na faixa de 0 a $q-1$, inclusive, onde q é um número inteiro positivo, serão capazes de decodificar todos os subfluxos de bits de conjuntos de camadas de saída com pelo menos um esquema de divisão com p divisões para as quais todas as seguintes condições se aplicam:

- deixe a variável Partition[j] para j na faixa de 0 a $p-1$, inclusive, onde p é menor ou igual a q , indicar a j^a divisão no esquema de divisão.
- a união de camadas que estão contidas em divisões da Partition[0] a Partition[p-1], inclusive, representa todas as camadas no conjunto de camadas de saída.
- deixe a variável PartitionAssignment[i] para i na faixa de 0 a $p-1$, inclusive, ser usada para indicar a atribuição de decodificação de tripletos perfil, plano e nível $(P_{PartitionAssignment[i]}, T_{PartitionAssignment[i]},$

$L_{PartitionAssignment[i]}$ para $Partition[j]$. O valor de $PartitionAssignment[j]$ é derivado como a seguir:

- deixe $UnassignedProfileTierLevelTriples$ ser um conjunto de tripletos perfil, plano e nível com $UnassignedProfileTierLevelTriples[i]$ ser igual (P_i, T_i, L_i) para i na faixa de 0 a $q-1$, inclusive.
- o seguinte se aplica:

Para ($j = 0; j < p; j++$)

$PartitionAssignment[j] = -1$.

Se _____ (houver _____ um $UnassignedProfileTierLevelTriples[x]$ não vazio de modo que general profile idc da $Partition[j]$ seja igual _____ a _____ P_x , _____ ou general profile compatibility flag [P_x]

Ser igual a 1 para $Partition[j]$ e

General tier flag de $Partition[j]$ representar um plano mais baixo ou igual a T_x , e

General level idc de $Partition[j]$ representar um nível

Representa um nível mais baixo ou igual a L_x) {

_____ $PartitionAssignment[j] = x$

_____ $UnassignedProfileTierLevelTriples[x]$ vazio

}

- não haverá entrada em $PartitionAssignment[j]$ que seja igual a -1 j na faixa de 0 a $p-1$, inclusive.

[0194] A figura 5A é um fluxograma ilustrando uma operação de exemplo de codificador de vídeo 20, de acordo com uma técnica dessa revelação. No exemplo da figura 5A, o codificador de vídeo 20 gera um fluxo de bits que compreende uma representação codificada dos dados de vídeo (300). O fluxo de bits compreende uma pluralidade de

camadas. Como parte de gerar o fluxo de bits, o codificador de vídeo 20 inclui, no fluxo de bits, um ou mais elementos de sintaxe indicando um ou mais esquemas de divisão (302). Para cada esquema de divisão respectivo de um ou mais esquemas de divisão, o esquema de divisão respectivo especifica um conjunto respectivo de divisões desconexas cuja união forma um conjunto de camadas de saída, em que cada divisão respectiva do conjunto respectivo de divisões desconexas contém uma ou mais camadas, o conjunto de camadas de saída consistindo em todas as camadas da pluralidade de camadas ou um subconjunto adequado da pluralidade de camadas.

[0195] Em alguns exemplos, como parte de gerar o fluxo de bits, para cada divisão respectiva de cada esquema de divisão respectivo de um ou mais esquemas de divisão, o codificador de vídeo 20 pode indicar, no fluxo de bits, um elemento de sintaxe respectivo indicando um perfil, um plano e um nível para a divisão respectiva.

[0196] Além disso, em alguns exemplos, o codificador de vídeo 20 pode incluir, no fluxo de bits, elementos de sintaxe indicando uma pluralidade de conjuntos de camadas de saída, a pluralidade de conjuntos de camadas de saída incluindo o primeiro conjunto de camadas de saída. Para cada conjunto de camadas de saída respectivo da pluralidade de conjuntos de camada de saída, o codificador de vídeo 20 pode incluir, no fluxo de bits, elementos de sintaxe indicando um ou mais esquemas de divisão para o conjunto de camadas de saída respectivo.

[0197] Em alguns exemplos, como parte de sinalizar um ou mais elementos de sintaxe indicando um ou mais esquemas de divisão, para cada esquema de divisão respectivo de um ou mais esquemas de divisão, para cada divisão respectiva do esquema de divisão respectiva, para

cada camada respectiva do conjunto de camadas de saída, o codificador de vídeo 20 pode incluir, no fluxo de bits, um elemento de sintaxe indicando se a camada respectiva é incluída na divisão respectiva.

[0198] Além disso, no exemplo da figura 5A, o codificador de vídeo 20 transmite o fluxo de bits (304). Por exemplo, o codificador de vídeo 20 pode transmitir o fluxo de bits para uma interface de saída, como interface de saída 22. Em alguns exemplos, o codificador de vídeo 20 pode transmitir o fluxo de bits para uma ou mais mídias de armazenagem legível em computador para armazenagem.

[0199] A figura 5B é um fluxograma ilustrando uma operação de exemplo do decodificador de vídeo 30, de acordo com uma técnica dessa revelação. No exemplo da figura 5B, o decodificador de vídeo 30 decodifica um fluxo de bits que compreende uma representação codificada dos dados de vídeo, o fluxo de bits compreendendo uma pluralidade de camadas (320). Como parte de decodificar o fluxo de bits, o decodificador de vídeo 30 obtém a partir do fluxo de bits, um ou mais elementos de sintaxe indicando um ou mais esquemas de divisão (322). Para cada esquema de divisão respectivo de um ou mais esquemas de divisão, o esquema de divisão respectivo especifica um conjunto respectivo de divisões desconexas cuja união forma um conjunto de camadas de saída. Cada divisão respectiva do conjunto respectivo de divisões desconexas contém uma ou mais das camadas. O conjunto de camadas de saída pode consistir em todas as camadas da pluralidade de camadas ou um subconjunto adequado da pluralidade de camadas. Em alguns exemplos, como parte de obter um ou mais elementos de sintaxe, para cada esquema de divisão respectivo de um ou mais esquemas de divisão, para cada divisão respectiva do esquema de divisão respectivo, para cada camada

respectiva do conjunto de camadas de saída, o decodificador de vídeo 30 pode obter, a partir do fluxo de bits, um elemento de sintaxe indicando se a camada respectiva é incluída na divisão respectiva.

[0200] Além disso, como parte de decodificar o fluxo de bits, o decodificador de vídeo 30 decodifica cada das divisões de um esquema de divisão específico usando núcleos de processamento diferentes em uma pluralidade de núcleos de hardware, o esquema de divisão específico sendo um de um ou mais esquemas de divisão (324). Em alguns exemplos, como parte de decodificar o fluxo de bits, o decodificador de vídeo 30 pode, para cada divisão respectiva de cada esquema de divisão respectivo de um ou mais esquemas de divisão, obter, a partir do fluxo de bits, um elemento de sintaxe respectivo que indica informação respectiva de perfil, plano e nível (PTL) para a divisão respectiva.

[0201] Em alguns exemplos, o decodificador de vídeo 30 pode determinar, com base em informação PTL para divisões dos esquemas de divisão, o esquema de divisão específico a partir de um ou mais esquemas de divisão. Além disso, em tais exemplos, o decodificador de vídeo 30 pode decodificar cada divisão respectiva do esquema de divisão específica usando um núcleo respectivo dos núcleos de hardware, em que cada núcleo respectivo dos núcleos de hardware implementa um decodificador de cada única respectivo. Em alguns exemplos, pelo menos um primeiro núcleo de hardware da pluralidade de núcleos de hardware suporte um perfil, plano ou nível diferente de um segundo núcleo de hardware da pluralidade de núcleos de hardware. Em outros exemplos, pelo menos um primeiro núcleo de hardware da pluralidade de núcleos de hardware e um segundo núcleo de hardware diferente da pluralidade de núcleos de

hardware suporta o mesmo conjunto de perfis, planos e níveis.

[0202] Em alguns exemplos, quando o decodificador de vídeo 30 decodifica o fluxo de bits, o decodificador de vídeo 30 obtém, a partir do fluxo de bits, elementos de sintaxe indicando uma pluralidade de conjuntos de camadas de saída, a pluralidade de conjuntos de camadas de saída incluindo o primeiro conjunto de camadas de saída. Em tais exemplos, para cada conjunto de camadas de saída respectivo da pluralidade de conjuntos de camadas de saída, o decodificador de vídeo 30 pode obter, a partir do fluxo de bits, elementos de sintaxe indicando um ou mais esquemas de divisão para o respectivo conjunto de camadas de saída.

[0203] A figura 6A é um fluxograma ilustrando uma operação de exemplo do codificador de vídeo 20, de acordo com uma técnica dessa revelação. No exemplo da figura 6A, o codificador de vídeo 20 gera um fluxo de bits que compreende uma representação codificada dos dados de vídeo (350). O fluxo de bits compreende uma pluralidade de camadas. Como parte de gerar o fluxo de bits, o codificador de vídeo 20 inclui, no fluxo de bits, um ou mais elementos de sintaxe indicando um ou mais esquemas de divisão para um conjunto de camadas de saída (352). No exemplo da figura 6A, o conjunto de camadas de saída consiste em todas as camadas da pluralidade de camadas ou um subconjunto adequado da pluralidade de camadas. Além disso, para cada esquema de divisão respectivo de um ou mais esquemas de divisão, o esquema de divisão respectivo especifica uma divisão do conjunto de camadas de saída em uma ou mais divisões, em que cada camada no conjunto de camadas de saída pertence exatamente a uma divisão no esquema de divisão respectivo. Adicionalmente, como parte de gerar o fluxo de bits, para cada esquema de divisão respectivo de

um ou mais esquemas de divisão, o codificador de vídeo 20 inclui, em uma mensagem SEI no fluxo de bits ou VUI em um VPS no fluxo de bits, um conjunto respectivo de elementos de sintaxe (por exemplo, elementos de sintaxe bsp_comb_hrd_idx) indicando conjuntos de parâmetro HRD para o esquema de divisão respectivo (354). O codificador de vídeo 20 transmite o fluxo de bits (356). Por exemplo, o codificador de vídeo 20 pode transmitir o fluxo de bits para uma interface de saída, como interface de saída 22. Em alguns exemplos, o codificador de vídeo 20 pode transmitir o fluxo de bits para uma ou mais mídia de armazenagem de dados legível em computador para armazenagem.

[0204] Em alguns exemplos, como parte de incluir o conjunto respectivo de elementos de sintaxe indicando os conjuntos de parâmetros HRD para cada de um ou mais esquemas de divisão compreende, o codificador de vídeo 20 pode incluir, em uma estrutura de sintaxe de parâmetros de programação de fluxo de bits (BSP) no VUI, um ou mais conjuntos de parâmetros HRD. Além disso, em tais exemplos, para cada combinação respetiva em um conjunto de uma ou mais combinações de programa para o esquema de divisão respectivo, para cada divisão respectiva do esquema de divisão respectivo, o codificador de vídeo 20 pode incluir, na estrutura de sintaxe BSP, um elemento de sintaxe respectivo especificando o conjunto de parâmetros HRD para a divisão respetiva na estrutura de sintaxe BSP e um elemento de sintaxe respectivo especificando um programa de fornecimento.

[0205] Em alguns exemplos, como parte de incluir o conjunto respectivo de elementos de sintaxe indicando os conjuntos de parâmetros HRD para cada de um ou mais esquemas de divisão compreende, o codificador de vídeo 20 pode incluir, em uma estrutura de sintaxe de parâmetros

de programação de fluxo de bits (BSP) na mensagem SEI, um ou mais conjuntos de parâmetros HRD. Além disso, em tais exemplos, para cada combinação respectiva em um conjunto de uma ou mais combinações de programa para o esquema de divisão respectivo, para cada divisão respectiva do esquema de divisão respectivo, o codificador de vídeo 20 pode incluir, na estrutura de sintaxe BSP, um elemento de sintaxe respectivo especificando um conjunto de parâmetros HRD para a divisão respectiva na estrutura de sintaxe BSP e um elemento de sintaxe respectivo especificando um programa de fornecimento.

[0206] A figura 6B é um fluxograma ilustrando uma operação de exemplo de decodificador de vídeo 30, de acordo com uma técnica dessa revelação. No exemplo da figura 6B, o decodificador de vídeo 30 obtém, a partir de um fluxo de bits que compreende uma representação codificada de dados de vídeo, um ou mais elementos de sintaxe indicando um ou mais esquemas de divisão para um conjunto de camadas de saída (370). No exemplo da figura 6B, o conjunto de camadas de saída consiste em todas as camadas da pluralidade de camadas ou um subconjunto adequado da pluralidade de camadas. Para cada esquema de divisão respectivo de um ou mais esquemas de divisão, o esquema de divisão respectivo especifica uma divisão do conjunto de camadas de saída em uma ou mais divisões, em que cada camada no conjunto de camadas de saída pertence exatamente a uma divisão no esquema de divisão respectivo.

[0207] Além disso, no exemplo da figura 6B, para cada esquema de divisão respectivo de um ou mais esquemas de divisão, o decodificador de vídeo 30 obtém, a partir de uma mensagem SEI no fluxo de bits ou VUI em um VPS no fluxo de bits, um conjunto respetivo de elementos de sintaxe (por exemplo, elementos de sintaxe

bsp_comb_hrd_idx) indicando conjuntos de parâmetros HRD para o esquema de divisão respectivo (372).

[0208] Em alguns exemplos, como parte de obter um conjunto respectivo de elementos de sintaxe indicando conjuntos de parâmetros HRD para o esquema de divisão respectivo, o decodificador de vídeo 30 obtém, a partir de uma estrutura de sintaxe de parâmetros de programação de fluxo de bits (BSP) na VUI, um ou mais conjuntos de parâmetros HRD. Em tais exemplos, para cada combinação respectiva em um conjunto de uma ou mais combinações de programa para o respectivo esquema de divisão, para cada divisão respectiva do esquema de divisão respectivo, o decodificador de vídeo 30 pode obter, a partir da estrutura de sintaxe BSP, um elemento de sintaxe respectivo especificando um conjunto de parâmetros HRD para a divisão respectiva na estrutura de sintaxe BSP e um elemento de sintaxe respectivo especificando um programa de fornecimento.

[0209] Em alguns exemplos, como parte de obter um conjunto respectivo de elementos de sintaxe indicando conjuntos de parâmetros HRD para o esquema de divisão respectivo, o decodificador de vídeo 30 obtém, a partir de uma estrutura de sintaxe de parâmetros de programação de fluxo de bits (BSP) na mensagem SEI, um ou mais conjuntos de parâmetros HRD. Em tais exemplos, para cada combinação respectiva em um conjunto de uma ou mais combinações de programa para o esquema de divisão respectivo, para cada divisão respectiva do esquema de divisão respectivo, o decodificador de vídeo 30 pode obter, a partir da estrutura de sintaxe BSP, um elemento de sintaxe respectivo especificando um conjunto de parâmetros HRTD para a divisão respectiva na estrutura de sintaxe BSP e um elemento de sintaxe respectivo especificando um programa de

fornecimento.

[0210] Para cada divisão respectiva do esquema de divisão respectivo, o decodificador de vídeo 30 testa, com base nos parâmetros HRD para o esquema de divisão respectivo, a divisão respectiva para conformidade (374). Em alguns exemplos, como parte de teste de subfluxos de bits, o decodificador de vídeo 30 aplica, com base em uma divisão selecionada de um esquema de divisão específico de um ou mais esquemas de divisão não contendo todas as camadas do conjunto de camadas de saída, uma operação CPB específica de divisão. Em alguns exemplos, o teste de conformidade de uma divisão específica assume que as imagens decodificadas e outras informações necessárias a partir das camadas de referência direta e indireta das camadas na divisão, e que não estão contidas na divisão específica, são disponíveis no momento de decodificar as imagens das camadas na divisão específica.

[0211] A figura 7 é um fluxograma ilustrando uma operação de exemplo de decodificador de vídeo 30, de acordo com uma técnica dessa revelação. No exemplo da figura 7, o decodificador de vídeo 30 determina, com base em uma lista de entradas de triplete, se o decodificador de vídeo 30 (isto é, um decodificador de vídeo de multicamadas) é capaz de decodificar um fluxo de bits que compreende uma representação codificada de dados de vídeo de multicamadas (400). Em alguns exemplos, o número de entradas de triplete na lista é menor ou igual a um número de decodificadores de camada única nos dados de vídeo de multicamadas. Em alguns exemplos, um núcleo de hardware único de decodificador de vídeo 30 pode decodificar múltiplas camadas. Cada entrada de triplete respectiva na lista de entradas de triplete indica um perfil, um plano e um nível para um decodificador de camada única respectiva

no decodificador de vídeo de multicamadas. Com base no decodificador de vídeo de multicamadas sendo capaz de decodificar o fluxo de bits, o decodificador de vídeo 30 decodifica o fluxo de bits (402).

[0212] Em alguns exemplos, o decodificador de vídeo 30 obtém, a partir do fluxo de bits, um ou mais elementos de sintaxe que indicam um ou mais esquemas de divisão. Para cada esquema de divisão respectivo de um ou mais esquemas de divisão, o esquema de divisão respectivo especifica um conjunto respectivo de divisões desconexas cuja união forma um conjunto de camadas de saída. Cada divisão respectiva do respectivo conjunto de divisões desconexas contém uma ou mais das camadas, o conjunto de camadas de saída consistindo em todas as camadas da pluralidade de camadas ou um subconjunto adequado da pluralidade de camadas. Como parte de determinar se o decodificador de vídeo de multicamadas é capaz de decodificar o fluxo de bits, o decodificador de vídeo 30 pode determinar, com base em uma condição sendo atendida, que o decodificador de vídeo de multicamadas é capaz de decodificar o fluxo de bits. Em tais exemplos, a condição é atendida com base em haver um esquema de divisão adequado em um ou mais esquemas de divisão. Além disso, em tais exemplos, para cada divisão respectiva do esquema de divisão adequado, um dos decodificadores de camada única pode ser atribuído para decodificar a divisão respectiva. Em alguns exemplos, como parte de determinar que o decodificador de vídeo de multicamadas é capaz de decodificar o fluxo de bits, para cada divisão respectiva do esquema de divisão adequado. Com base nas seguintes condições sendo atendidas, o decodificador de vídeo de multicamadas pode determinar que a divisão respectiva pode ser atribuída a um dos decodificadores de camada única: um

nível associado ao decodificador de camada única é maior ou igual a um nível sinalizado para a divisão respectiva, um plano associado ao decodificador de camada única é maior ou igual a um plano sinalizado para a divisão respectiva, e um perfil associado à divisão respectiva é tal que o decodificador de camada única suporta decodificação do perfil.

[0213] 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 em ou transmitidas sobre como uma ou mais instruções ou código em uma mídia legível em computador e executadas por uma unidade de processamento baseado em hardware. Mídia legível em computador pode incluir tanto mídia tangível como de armazenagem em computador, ou mídia de comunicação incluindo qualquer mídia que facilite a transferência de um programa de computador a partir de um lugar para outro, por exemplo, de acordo com um protocolo de comunicação. Desse modo, mídia legível em computador pode corresponder em geral a (1) mídia de armazenagem legível em computador que seja não transitória ou (2) uma mídia de comunicação como uma onda portadora ou sinal. Mídia de armazenagem de dados pode ser qualquer mídia disponível que pode ser acessada por um ou mais computadores, ou um ou mais processadores para recuperar instruções, código e/ou estruturas de dados para implementação das técnicas descritas nessa revelação. Um produto de programa de computador pode incluir uma mídia legível em computador.

[0214] Como exemplo, e não limitação, tal mídia legível em computador pode compreender RAM, ROM, EEPROM, CD-ROM ou outra armazenagem de disco ótica, armazenagem de disco magnética ou outros dispositivos de

armazenagem magnética, memória flash, ou qualquer outra mídia que possa ser usada para armazenar código de programa desejado na forma de instruções ou estruturas de dados e que possam ser acessadas por um computador. Além disso, qualquer conexão é adequadamente denominada uma mídia legível em computador. Por exemplo, se instruções forem transmitidas de um website, servidor ou outra fonte remota usando um cabo coaxial, cabo de fibra ótica, par torcido, linha de assinante digital (DSL), ou tecnologias sem fio como infravermelho, rádio e micro-onda, então o cabo coaxial, cabo de fibra ótica, par torcido, DSL ou tecnologias sem fio como infravermelho, rádio e micro-onda são incluídas na definição de mídia. Deve ser entendido, entretanto, que mídia de armazenagem legível em computador e mídia de armazenagem de dados não incluem conexões, ondas portadoras, sinais ou outra mídia transite, porém são ao invés dirigidos a mídia de armazenagem tangível não transite. Disk e disco, como usados aqui, incluem compact disc (CD), disco laser, disco ótico, digital versatile disc (DVD), disco flexível e disco blu-ray, onde disks normalmente reproduzem dados magneticamente, enquanto discs reproduzem dados óticamente com lasers. Combinações do acima também devem ser incluídas no escopo de mídia legível em computador.

[0215] Instruções podem ser executadas por um ou mais processadores, como um ou mais processadores de sinais digitais (DSPs), microprocessadores de propósito geral, circuitos integrados de aplicação específica (ASICs), disposições de lógica programáveis em campo (FPGAs), ou outro conjunto de circuitos de lógica discreto ou integrado equivalente. Por conseguinte, o termo "processador", como usado aqui pode se referir a qualquer uma da estrutura acima ou qualquer outra estrutura adequada

para implementação das técnicas descritas aqui. Além disso, em alguns aspectos, a funcionalidade descrita aqui pode ser fornecida em módulos de hardware e/ou software dedicados configurados para codificação e decodificação ou incorporados em um codec combinado. Também, as técnicas podem ser totalmente implementadas em um ou mais circuitos ou elementos de lógica.

[0216] As técnicas dessa revelação podem ser implementadas em uma ampla variedade de dispositivos ou aparelhos, incluindo um aparelho telefônico sem fio, um circuito integrado (IC) ou um conjunto de ICs (por exemplo, um conjunto de chips). Vários componentes, módulos ou unidades são descritos nessa revelação para enfatizar aspectos funcionais de dispositivos configurados para executar as técnicas reveladas, porém não necessariamente exigem realização por unidades de hardware diferentes. Ao invés, como descrito acima, várias 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 combinação com software e/ou firmware adequado.

[0217] Vários exemplos foram descritos. Esses e outros exemplos estão compreendidos no escopo das reivindicações a seguir.

REIVINDICAÇÕES

1. Método para decodificar dados de vídeo de várias camadas, o método sendo **caraterizado** pelo fato de compreender:

decodificar um fluxo de bits que compreende uma representação codificada dos dados de vídeo de várias camadas, em que decodificar o fluxo de bits compreende obter, do fluxo de bits, elementos de sintaxe que indicam uma série de esquemas de particionamento diferentes, em que:

os dados de vídeo de várias camadas compreendem uma série de camadas, unidades de Camada de Acesso à Rede (NAL) do fluxo de bits que têm identificadores de camadas diferentes pertencentes a camadas diferentes da série de camadas, e

para cada respectivo esquema de particionamento da série de esquemas de particionamento, o respectivo esquema de particionamento especifica um respectivo conjunto de partições dissociadas cuja união forma um conjunto de camadas de saída, em que cada respectiva partição do respectivo conjunto de partições dissociadas contém uma ou mais das camadas;

determinar se um decodificador de vídeo de várias camadas é capaz de decodificar o conjunto de camadas de saída, em que o decodificador de vídeo de várias camadas se conforma a uma lista de entradas de triplete, cada respectiva entrada de triplete da lista de entradas de triplete indicando um perfil, uma fileira e um nível para um respectivo decodificador de camada única do decodificador de vídeo de várias camadas, e decodificadores que se conformam à lista de entradas de triplete são obrigados a serem capazes de decodificar o conjunto de camadas de saída se a série de esquemas de particionamento

incluir um esquema de particionamento adequado no qual, para cada respectiva partição do esquema de particionamento, o perfil, fileira e nível de informação sinalizada para referida respectiva partição indica que um decodificador de camada única do decodificador de vídeo de várias camadas pode ser atribuído exclusivamente para decodificar a respectiva partição; e

com base na capacidade do decodificador de vídeo de várias camadas ser capaz de decodificar o fluxo de bits, decodificar, pelo decodificador de vídeo de várias camadas, o fluxo de bits.

2. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que o número de entradas de triplete na lista é menor ou igual ao número de decodificadores de camada única no decodificador de vídeo de várias camadas.

3. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que determinar que o decodificador de vídeo de várias camadas é capaz de decodificar o fluxo de bits compreende:

para cada respectiva partição do esquema de particionamento adequado:

com base na satisfação das condições seguintes, determinar que a respectiva partição pode ser atribuída a um decodificador de camada única do decodificador de vídeo de várias camadas:

- um nível associado ao decodificador de camada única é mais elevado ou igual ao nível sinalizado para a respectiva partição,

- uma fileira associada ao decodificador de camada única é maior ou igual à fileira sinalizada para a respectiva partição, e

- um perfil associado à respectiva partição é tal que o decodificador de camada única suporte a decodificação do perfil.

4. Dispositivo para decodificar dados de vídeo de várias camadas, o dispositivo sendo **caracterizado** pelo fato de compreender:

meios para decodificar um fluxo de bits que compreende uma representação codificada dos dados de vídeo de várias camadas, os meios para decodificar o fluxo de bits compreendendo meios para obter, do fluxo de bits, elementos de sintaxe que indicam uma série de esquemas de particionamento diferentes, em que:

os dados de vídeo de várias camadas compreendem uma série de camadas, unidades de Camada de Acesso à Rede (NAL) do fluxo de bits que têm identificadores de camadas diferentes pertencentes a camadas diferentes da série de camadas, e

para cada respectivo esquema de particionamento da série de esquemas de particionamento, o respectivo esquema de particionamento especifica um respectivo conjunto de partições dissociadas cuja união forma um conjunto de camadas de saída, em que cada respectiva partição do respectivo conjunto de partições dissociadas contém uma ou mais das camadas;

meios para determinar, com base em uma lista de entradas de triplete, se um decodificador de vídeo de várias camadas é capaz de decodificar o conjunto de camadas de saída, em que o decodificador de vídeo de várias camadas se conforma a uma lista de entradas de triplete, um fluxo de bits que compreende uma representação codificada dos dados de vídeo de várias camadas, em que cada respectiva entrada de triplete da lista de entradas de triplete indica um perfil, uma fileira e um nível para um respectivo

decodificador de camada única do decodificador de vídeo de várias camadas e dos decodificadores que se conformam à lista de entradas de triplete se exige que sejam capazes de decodificar o conjunto de camadas de saída se a pluralidade de esquemas de particionamento incluir um esquema de particionamento adequado no qual, para cada respectiva partição do esquema de particionamento, o perfil, fileira e nível de informação sinalizada para referida respectiva partição indica que um decodificador de camada única do decodificador de vídeo de várias camadas pode ser atribuído exclusivamente para decodificar a respectiva partição; e

meios para decodificar, com base na capacidade do decodificador de vídeo de várias camadas ser capaz de decodificar o fluxo de bits, o fluxo de bits.

5. Dispositivo, de acordo com a reivindicação 4, **caracterizado** pelo fato de que o número de entradas de triplete na lista é menor ou igual ao número de decodificadores de camada única no decodificador de vídeo de várias camadas.

6. Dispositivo, de acordo com a reivindicação 4, **caracterizado** pelo fato de que os meios para determinar se o decodificador de vídeo de várias camadas é capaz de decodificar o fluxo de bits compreende:

para cada respectiva partição do esquema de particionamento adequado:

meios para determinar, com base na satisfação das condições seguintes, que a respectiva partição pode ser atribuída a um decodificador de camada única do decodificador de vídeo de várias camadas:

- um nível associado ao decodificador de camada única é mais elevado ou igual ao nível sinalizado para a respectiva partição,

- uma fileira associada ao decodificador de camada única é maior ou igual à fileira sinalizada para a respectiva partição, e

- um perfil associado à respectiva partição é tal que o decodificador de camada única suporta a decodificação do perfil.

7. Memória legível por computador caracterizada pelo fato de que possui instruções nela armazenadas que, quando executadas, fazem com que o codificador de vídeo de várias camadas realize o método conforme definido em qualquer uma das reivindicações 1 a 3.

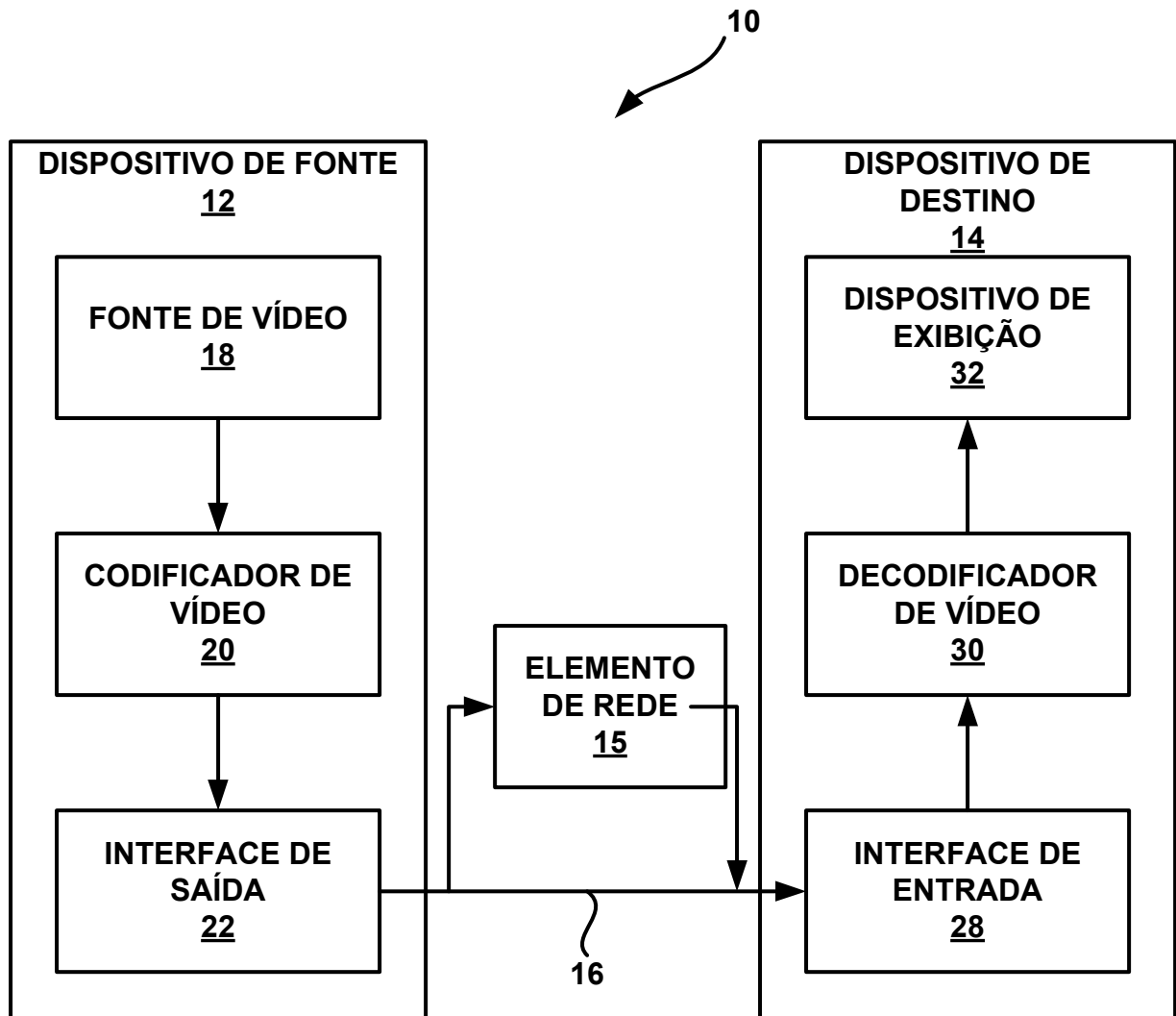


FIG. 1



FIG. 2

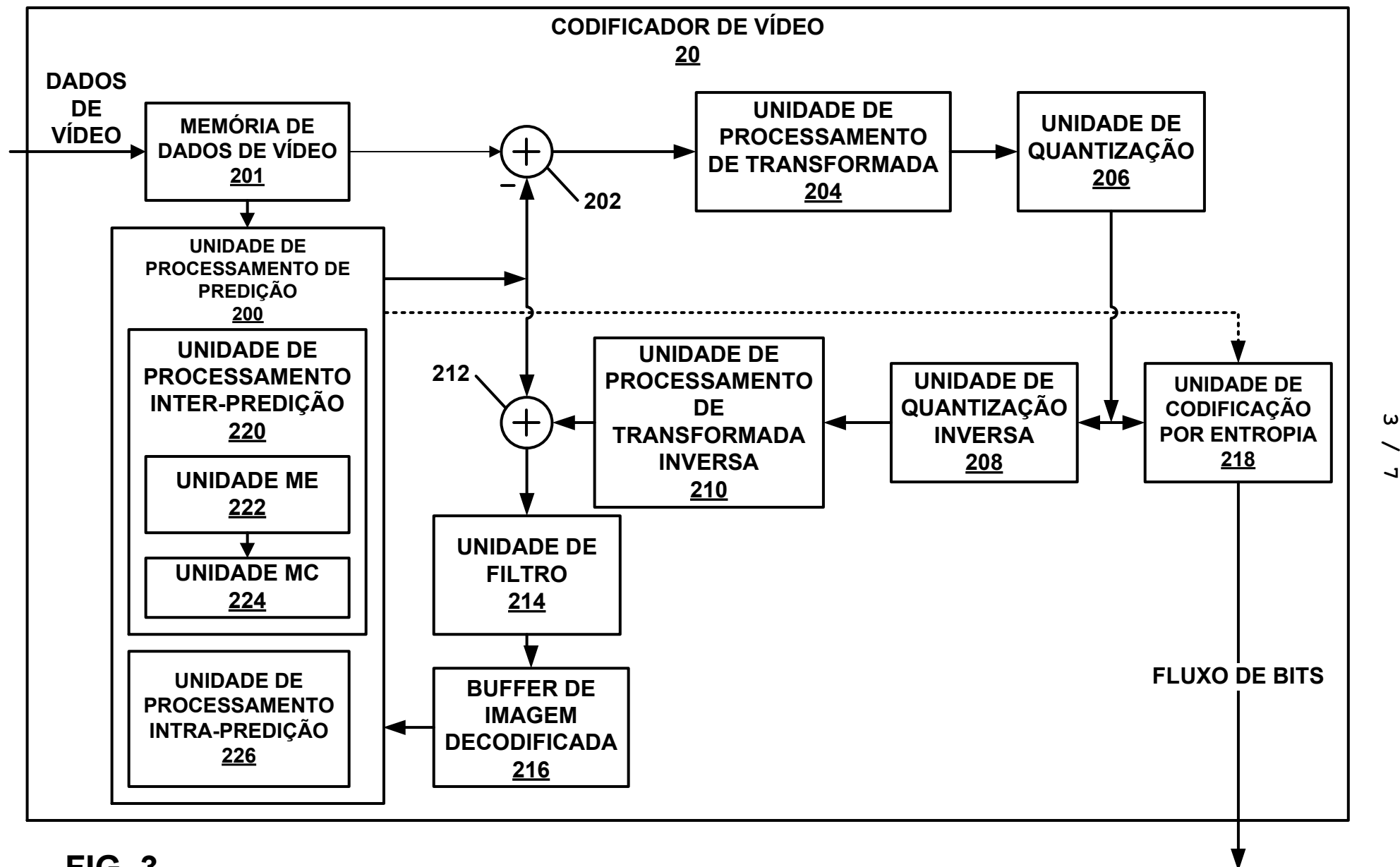


FIG. 3

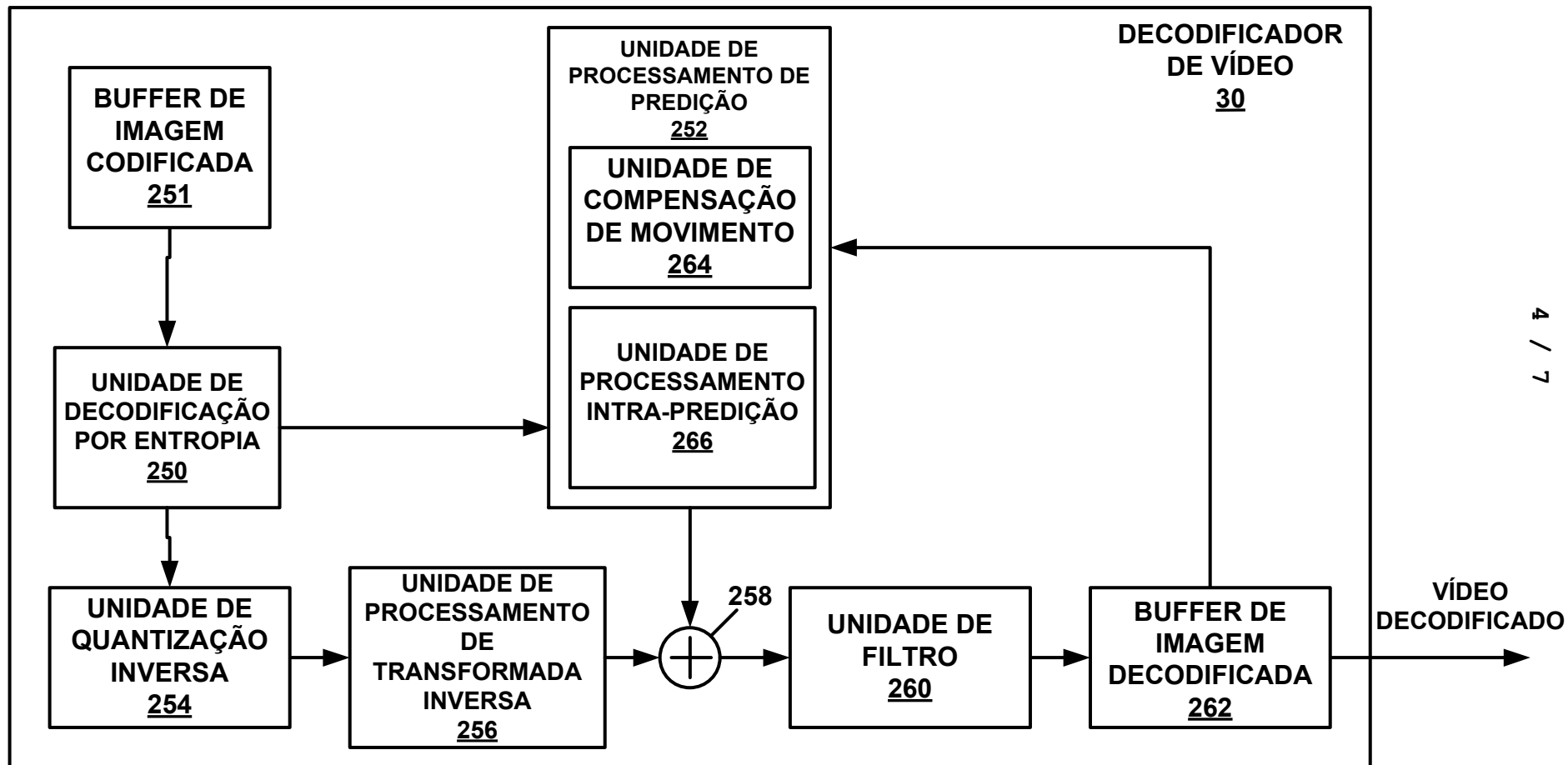


FIG. 4

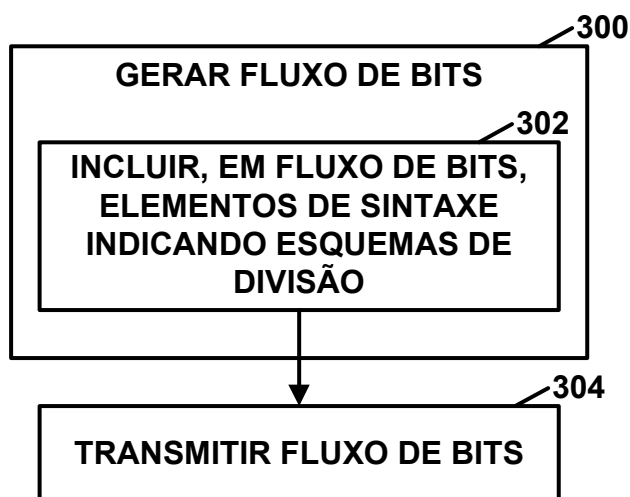


FIG. 5A



FIG. 5B

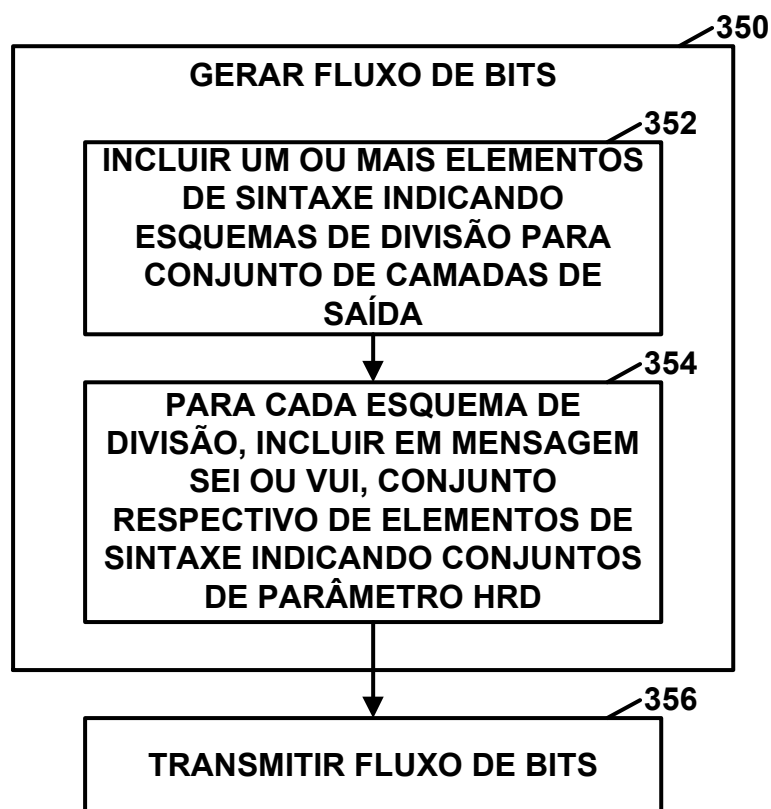


FIG. 6A

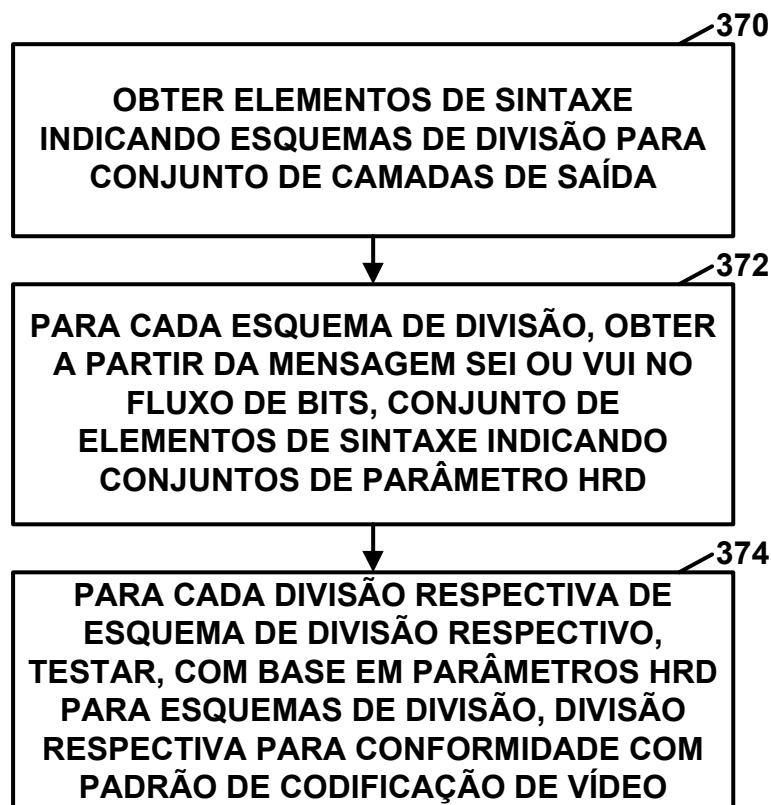


FIG. 6B

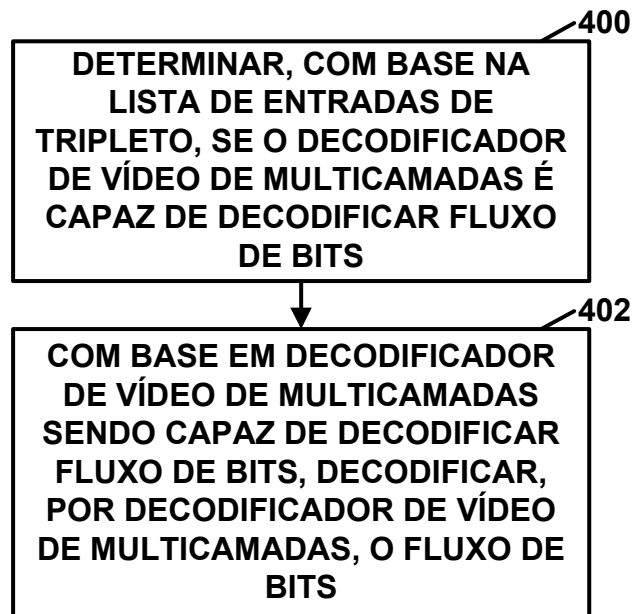


FIG. 7