



República Federativa do Brasil

Ministério do Desenvolvimento, Indústria,  
Comércio e Serviços

Instituto Nacional da Propriedade Industrial



**(11) BR 112016008337-7 B1**

**(22) Data do Depósito:** 15/10/2014

**(45) Data de Concessão:** 02/01/2024

**(54) Título:** APARELHO CONFIGURADO PARA CODIFICAR INFORMAÇÕES DE VÍDEO, MÉTODO PARA CODIFICAR INFORMAÇÕES DE VÍDEO E MEMÓRIA LEGÍVEL POR COMPUTADOR

**(51) Int.Cl.:** H04N 19/70; H04N 19/597; H04N 19/30.

**(30) Prioridade Unionista:** 14/10/2014 US 14/514,153; 15/10/2013 US 61/891,264.

**(73) Titular(es):** QUALCOMM INCORPORATED.

**(72) Inventor(es):** YE-KUI WANG; VADIM SEREGIN; ADARSH KRISHNAN RAMASUBRAMONIAN.

**(86) Pedido PCT:** PCT US2014060660 de 15/10/2014

**(87) Publicação PCT:** WO 2015/057818 de 23/04/2015

**(85) Data do Início da Fase Nacional:** 14/04/2016

**(57) Resumo:** DISPOSITIVO E MÉTODO PARA CODIFICAÇÃO ESCALONÁVEL DE INFORMAÇÕES DE VÍDEO. Trata-se de um aparelho configurado para codificar informações de vídeo que inclui uma unidade de memória e um processador em comunicação com a unidade de memória. A unidade de memória é configurada para armazenar informações de vídeo associadas a uma camada de referência (RL) e uma camada de aperfeiçoamento (EL), sendo que a RL tem uma figuração de RL em uma primeira unidade de acesso, e a EL tem uma primeira figuração de EL na primeira unidade de acesso, em que a primeira figuração de EL é associada a um primeiro conjunto de parâmetros; O processador é configurado para determinar se a primeira figuração de EL é uma figuração de ponto de acesso intra-aleatório (IRAP), determinar se a primeira unidade de acesso segue imediatamente um ponto de emenda em que as primeiras informações de vídeo são unidas às segundas informações de vídeo que incluem a primeira figuração de EL, e realizar, com base na determinação de se a primeira figuração de EL é uma figuração de ponto de acesso intra-aleatório (IRAP) e se a primeira unidade de acesso segue imediatamente um ponto de emenda, um dentre (1) abster-(...).

**"APARELHO CONFIGURADO PARA CODIFICAR INFORMAÇÕES DE VÍDEO,  
MÉTODO PARA CODIFICAR INFORMAÇÕES DE VÍDEO E MEMÓRIA  
LEGÍVEL POR COMPUTADOR"**

**CAMPO DA TÉCNICA**

[0001] A presente descrição refere-se ao campo de compactação e codificação (coding) de vídeo, particularmente, à codificação de vídeo escalonável (SVC), codificação de vídeo de múltiplas visualizações (MVC) ou codificação de vídeo em 3D (3DV).

**FUNDAMENTOS**

[0002] As capacidades de vídeo digital podem ser incorporadas em uma grande variedade de dispositivos, incluindo televisões digitais, sistemas de difusão direta digital, sistema de difusão sem fio, assistentes digitais pessoais (PDAs), computadores do tipo laptop ou desktop, câmeras digitais, dispositivos de gravação digital, reprodutores de mídia digital, dispositivos de jogos eletrônicos, consoles de jogos eletrônicos, telefones de rádio via satélite ou celular, dispositivos de teleconferência por vídeo e similares. Os dispositivos de vídeo digital implantam técnicas de compactação de vídeo como as descritas nos padrões definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificação de Vídeo Avançada (AVC), o padrão de Codificação de Vídeo de Alta Eficiência (HEVC) atualmente sob desenvolvimento e extensões de tais padrões. Os dispositivos de vídeo podem transmitir, receber, codificar (encode), decodificar e/ou armazenar informações de vídeo digital de maneira mais eficiente implantando-se tais técnicas de codificação de vídeo.

[0003] As técnicas de compactação de vídeo realizam predição espacial (intraimagem) e/ou predição temporal (interimagem) para reduzir ou remover a

redundância inerente em sequências de vídeo. Para a codificação de vídeo com base em bloco, uma fatia de vídeo (por exemplo, um quadro de vídeo ou uma porção de um quadro de vídeo) pode ser particionado em blocos de vídeo, que também podem ser denominados de blocos em árvore, unidades de codificação (CUs) e/ou nós de codificação. Os blocos de vídeo em uma fatia intracodificada (I) de uma imagem são codificados com o uso de predição espacial em relação às amostras de referência em blocos próximos na mesma imagem. Os blocos de vídeo em uma fatia intercodificada (P ou B) de uma imagem podem usar predição espacial em relação a amostras de referência em blocos próximos na mesma imagem ou predição temporal em relação às amostras de referência em outras imagens de referência. As imagens podem ser denominadas como quadros e as imagens de referência podem ser denominadas como quadros de referência.

[0004] A predição espacial ou temporal resulta em um bloco preditivo para um bloco a ser codificado. Os dados residuais representam diferenças de pixel entre o bloco original a ser codificado e o bloco preditivo. Um bloco intercodificado é codificado de acordo com um vetor de movimento que aponta para um bloco de amostras de referência que forma o bloco preditivo, sendo que os dados residuais indicam a diferença entre o bloco codificado e o bloco preditivo. Um bloco intracodificado é codificado de acordo com um modo de intracodificação e os dados residuais. Para compactação adicional, os dados residuais podem ser transformados do domínio de pixel para um domínio de transformada, resultando em coeficientes residuais que podem, em seguida, ser quantizados. Os coeficientes de transformada quantizados dispostos inicialmente em uma matriz bidimensional podem ser varridos a fim de produzir um vetor monodimensional de coeficiente de transformada e a

codificação por entropia pode ser aplicada para conseguir ainda mais compactação.

#### BREVE DESCRIÇÃO

[0005] A codificação de vídeo escalonável (SVC) se refere à codificação de vídeo na qual uma camada de base (BL), algumas vezes referida como uma camada de referência (RL) e uma ou mais camadas de aperfeiçoamento escalonável (ELs) são usadas. Uma SVC, a camada de base pode transportar os dados de vídeo com um nível de base de qualidade. À medida que uma ou mais camadas de aperfeiçoamento podem transportar os dados de vídeo adicionais para suportar, por exemplo, maiores níveis de sinal/ruído (SNR), temporais e/ou espaciais. As camadas de aperfeiçoamento podem ser definidas em relação a uma camada anteriormente codificada. Por exemplo, uma camada de fundo pode servir como uma BL, enquanto uma camada de topo pode servir como uma EL. As camadas intermediárias podem servir como ELs ou RLs ou ambas. Por exemplo, uma camada média (por exemplo, uma camada que não é nem a camada mais inferior nem a camada mais superior) pode ser uma EL para as camadas abaixo da camada média, tais como a camada de base ou quaisquer camadas de realce intervenientes e ao mesmo tempo servir como uma RL para uma ou mais camadas de realce acima da camada média. De modo similar, na extensão de Múltiplas Visualizações ou 3D do padrão HEVC, pode haver múltiplas vistas e as informações de uma vista podem ser utilizadas para codificar (por exemplo, codificar ou decodificar) as informações de outra vista (por exemplo, estimativa de movimento, predição de vetor de movimento e/ou outras redundâncias).

[0006] Em SVC, os parâmetros usados pelo codificador (encoder) ou pelo decodificador são agrupados em conjuntos de parâmetros com base no nível de codificação

(por exemplo, nível de vídeo, nível de sequência, nível de imagem, nível de fatia, etc.) em que os mesmos podem ser utilizados. Por exemplo, os parâmetros que podem ser utilizados por uma ou mais sequências de vídeo codificado no fluxo de bits podem estar incluídos em um conjunto de parâmetros de vídeo (VPS) e os parâmetros que são utilizados por uma ou mais imagens em uma sequência de vídeo codificado podem estar incluídos em um conjunto de parâmetros de sequência (SPS). De modo similar, os parâmetros que são utilizados por uma ou mais fatias em uma imagem podem estar incluídos em um conjunto de parâmetros de imagem (PPS) e os outros parâmetros que são específicos para uma única fatia podem estar incluídos em um cabeçalho de fatia. De modo similar, a indicação de qual o(s) conjunto(s) de parâmetros uma camada particular está usando a um determinado tempo pode ser fornecida em vários níveis de codificação. Por exemplo, se o cabeçalho de fatia de uma fatia na camada particular se referir a um PPS, o PPS é ativado para a fatia ou a imagem que contém a fatia. De forma semelhante, se o PPS se referir a um SPS, o SPS é ativado para a imagem ou a sequência de vídeo codificado que contém a imagem, e se o SPS se referir a um VPS, o VPS é ativado para a sequência de vídeo codificado ou a camada de vídeo que contém a sequência de vídeo codificado.

[0007] Tipicamente, um conjunto de parâmetros permanece ativo (por exemplo, está atualmente sendo usado para decodificar um segmento particular do fluxo de bits) para o nível de codificação inteiro em que o conjunto de parâmetros é utilizado. Por exemplo, um SPS ativado pode permanecer ativo pela sequência inteira que ativou inicialmente o SPS, e um PPS ativado pode permanecer ativo pela imagem inteira que ativou inicialmente o PPS. Entretanto, alguns esquemas de codificação existentes, um

conjunto de parâmetros pode ser ativado no meio do nível de codificação em que outro conjunto de parâmetros está atualmente ativo. Em tal caso, alguns dos parâmetros (por exemplo, a resolução de imagem) que deveriam permanecer constantes pela duração do processo de codificação do nível de codificação particular (por exemplo, sequência, imagem, fatia, etc.) podem ser alterados pela ativação do novo conjunto de parâmetros e pode resultar em resultados indesejados.

[0008] Portanto, um esquema de codificação que gerencia a ativação dos conjuntos de parâmetros de forma mais eficaz e aprimora, assim, a precisão de codificação e resiliência a erro é desejado.

[0009] Cada um dentre os sistemas, métodos e dispositivos desta descrição tem diversos aspectos inovadores, nenhum dos quais é unicamente responsável pelos atributos desejáveis revelados no presente pedido.

[0010] Em um aspecto, um aparelho configurado para codificar (coding) (por exemplo, codificar (encoding) ou decodificar) as informações de vídeo incluem uma unidade de memória e um processador em comunicação com a unidade de memória. A unidade de memória é configurada para armazenar informações de vídeo associadas a uma camada de referência (RL) e uma camada de aperfeiçoamento (EL), sendo que a RL tem uma imagem RL em uma primeira unidade de acesso, e a EL tem uma primeira imagem EL na primeira unidade de acesso, em que a primeira imagem EL é associada a um primeiro conjunto de parâmetros; O processador é configurado para determinar se a primeira imagem EL é uma imagem de ponto de acesso intra-aleatório (IRAP), determinar se a primeira unidade de acesso segue imediatamente um ponto de emenda em que as primeiras informações de vídeo são unidas às segundas informações de vídeo que incluem a primeira imagem

EL, e realizar, com base na determinação de se a primeira imagem EL é uma imagem de ponto de acesso intra-aleatório (IRAP) e se a primeira unidade de acesso segue imediatamente um ponto de emenda, um dentre (1) abster-se de associar a primeira imagem EL a um segundo conjunto de parâmetros que é diferente do primeiro conjunto de parâmetros, ou (2) associar a primeira imagem EL a um segundo conjunto de parâmetros que é diferente do primeiro conjunto de parâmetros.

[0011] Em outro aspecto, um método para codificar informações de vídeo compreende determinar se uma primeira imagem de camada de aperfeiçoamento (EL) em uma primeira unidade de acesso é uma imagem de ponto de acesso intra-aleatório (IRAP), sendo que a primeira imagem EL está associada a um primeiro conjunto de parâmetros, determinar se a primeira unidade de acesso segue imediatamente um ponto de emenda em que as primeiras informações de vídeo são unidas às segundas informações de vídeo que incluem a primeira imagem EL, e realizar, com base na determinação de se a primeira imagem EL é uma imagem de ponto de acesso intra-aleatório (IRAP) e se a primeira unidade de acesso segue imediatamente um ponto de emenda, um dentre (1) abster-se de associar a primeira imagem EL a um segundo conjunto de parâmetros que é diferente do primeiro conjunto de parâmetros, ou (2) associar a primeira imagem EL a um segundo conjunto de parâmetros que é diferente do primeiro conjunto de parâmetros.

[0012] Em um outro aspecto, um meio legível por computador não transitório compreende o código que, quando executado, faz com que um aparelho realize um processo. O processo inclui armazenar informações de vídeo associadas a uma camada de referência (RL) e uma camada de aperfeiçoamento (EL), sendo que a RL tem uma imagem RL em

uma primeira unidade de acesso, e em que a EL tem uma primeira imagem EL na primeira unidade de acesso, em que a primeira imagem EL é associada a um primeiro conjunto de parâmetros, determinar se a primeira imagem EL é uma imagem de ponto de acesso intra-aleatório (IRAP), determinar se a primeira unidade de acesso segue imediatamente um ponto de emenda em que as primeiras informações de vídeo são unidas às segundas informações de vídeo que incluem a primeira imagem EL, e realizar, com base na determinação de se a primeira imagem EL é uma imagem de ponto de acesso intra-aleatório (IRAP) e se a primeira unidade de acesso segue imediatamente um ponto de emenda, um dentre (1) abster-se de associar a primeira imagem EL a um segundo conjunto de parâmetros que é diferente do primeiro conjunto de parâmetros, ou (2) associar a primeira imagem EL a um segundo conjunto de parâmetros que é diferente do primeiro conjunto de parâmetros.

[0013] Em outro aspecto, um dispositivo de codificação vídeo configurado para codificar informações de vídeo compreende meios para armazenar informações de vídeo associadas a uma camada de referência (RL) e uma camada de aperfeiçoamento (EL), sendo que a RL tem uma imagem RL em uma primeira unidade de acesso, e em que a EL tem uma primeira imagem EL na primeira unidade de acesso, em que a primeira imagem EL é associada a um primeiro conjunto de parâmetros, meios para determinar se a primeira imagem EL é uma imagem de ponto de acesso intra-aleatório (IRAP), meios para determinar se a primeira unidade de acesso segue imediatamente um ponto de emenda em que as primeiras informações de vídeo são unidas às segundas informações de vídeo que incluem a primeira imagem EL, e meios para realizar, com base na determinação de se a primeira imagem EL é uma imagem de ponto de acesso intra-aleatório (IRAP) e

se a primeira unidade de acesso segue imediatamente um ponto de emenda, um dentre (1) abster-se de associar a primeira imagem EL a um segundo conjunto de parâmetros que é diferente do primeiro conjunto de parâmetros, ou (2) associar a primeira imagem EL a um segundo conjunto de parâmetros que é diferente do primeiro conjunto de parâmetros.

#### BREVE DESCRIÇÃO DOS DESENHOS

[0014] A Figura 1A é um diagrama de blocos que ilustra um sistema de decodificação e codificação de vídeo exemplificativo que pode utilizar técnicas de acordo com os aspectos descritos nesta descrição.

[0015] A Figura 1B é um diagrama de blocos que ilustra outro sistema de decodificação e codificação de vídeo exemplificativo que pode realizar as técnicas de acordo com os aspectos descritos nesta descrição.

[0016] A Figura 2A é um diagrama de blocos que ilustra um exemplo de um codificador (encoder) de vídeo que pode implantar técnicas de acordo com os aspectos descritos nesta descrição.

[0017] A Figura 2B é um diagrama de blocos que ilustra um exemplo de um codificador de vídeo que pode implantar técnicas de acordo com os aspectos descritos nesta descrição.

[0018] A Figura 3A é um diagrama de blocos que ilustra um exemplo de um decodificador de vídeo que pode implantar técnicas de acordo com os aspectos descritos nesta descrição.

[0019] A Figura 3B é um diagrama de blocos que ilustra um exemplo de um decodificador de vídeo que pode implantar técnicas de acordo com os aspectos descritos nesta descrição.

[0020] A Figura 4 é um diagrama de blocos que

ilustra uma configuração exemplificativa de imagens em diferentes camadas, de acordo com uma modalidade da presente descrição.

[0021] A Figura 5 ilustra um fluxograma que ilustra um método de codificação de informações de vídeo, de acordo com uma modalidade da presente descrição.

#### DESCRIÇÃO DETALHADA

[0022] Certas modalidades descritas no presente documento se referem a predição intercamada para codificação de vídeo escalonável no contexto de codecs de vídeo avançados, tal como HEVC (Codificação de Vídeo de Alta Eficiência). Mais especificamente, a presente descrição se refere a sistemas e métodos para desempenho aprimorado de predição de intercamada na extensão de codificação de vídeo escalonável (SVC) de HEVC.

[0023] Na descrição abaixo, as técnicas de H.264/AVC relacionadas a certas modalidades são descritas; o padrão de HEVC e as técnicas relacionadas também são discutidas. Enquanto certas modalidades são descritas no presente documento no contexto dos padrões de HEVC e/ou H.264, um indivíduo que tem habilidade comum na técnica pode verificar que os sistemas e métodos revelados no presente documento podem ser aplicáveis a qualquer padrão de codificação de vídeo adequado. Por exemplo, as modalidades reveladas no presente documento podem ser aplicáveis a um ou mais dos padrões a seguir: ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 ou ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (também chamado de ISO/IEC MPEG-4 AVC), que incluem suas extensões de codificação de vídeo escalonável (SVC) e de codificação de Vídeo de Múltiplas Vistas (MVC).

[0024] HEVC, em geral, segue a estrutura de padrões de codificação de vídeo anteriores em muitos

aspectos. A unidade de predição em HEVC é diferente daquela em certos padrões de codificação de vídeo anteriores (por exemplo, macrobloco). De fato, o conceito de macrobloco não existe em HEVC conforme entendido em certos padrões de codificação de vídeo anteriores. O macrobloco é substituído por uma estrutura hierárquica com base em um esquema de árvore quaternária, que pode fornecer alta flexibilidade, entre outros benefícios possíveis. Por exemplo, dentro do esquema de HEVC, três tipos de blocos, Unidade de Codificação (CU), Unidade de Predição (PU) e Unidade de Transformada (TU), são definidos. A CU pode se referir à divisão de unidade básica de região. A CU pode ser considerada análoga ao conceito de macrobloco, mas HEVC não restringe o tamanho máximo de CUs e pode permitir a divisão recursiva em quatro CUs com tamanho igual para aprimorar a adaptabilidade de conteúdo. A PU pode ser considerada a unidade básica de inter/intrapredição e uma única PU pode conter múltiplas partições de formato arbitrário para codificar de modo eficaz os padrões de imagem irregular. A TU pode ser considerada a unidade básica de transformada. A TU pode ser definida de modo independente da PU; no entanto, o tamanho de uma TU pode ser limitado ao tamanho da CU a qual a TU pertence. Essa separação da estrutura de bloco em três diferentes conceitos pode permitir que cada unidade seja otimizada de acordo com a função respectiva da unidade, que pode resultar em eficiência de codificação aprimorada.

[0025] Apenas para propósitos ilustrativos, certas modalidades reveladas no presente documento são descritas com exemplos que incluem apenas duas camadas (por exemplo, uma camada inferior tal como a camada de base e uma camada superior tal como a camada de aperfeiçoamento). Deve-se compreender que tais exemplos podem ser aplicáveis

às configurações que incluem múltiplas camadas de aperfeiçoamento e/ou de base. Além disso, para facilitar a explicação, a descrição a seguir inclui os termos "quadros" ou "blocos" com referência a certas modalidades. No entanto, esses termos não se destinam a ser limitativos. Por exemplo, as técnicas descritas abaixo podem ser usadas com quaisquer unidades de vídeo adequadas, tais como blocos (por exemplo, CU, PU, TU, macroblocos, etc.), fatias, quadros, etc.

#### PADRÕES DE CODIFICAÇÃO DE VÍDEO

[0026] Uma imagem digital, tal como uma imagem de vídeo, uma imagem de TV, uma imagem estática ou uma imagem gerada por um gravador de vídeo ou um computador, pode consistir em pixels ou amostras dispostas em linhas verticais e horizontais. O número de pixels em uma única imagem está tipicamente na casa das dezenas de milhares. Cada pixel contém tipicamente informações de luminância e crominância. Sem compactação, a quantidade absoluta de informações a ser transmitida de um codificador de imagem para um decodificador de imagem tornaria impossível a transmissão de imagem em tempo real. Para reduzir a quantidade de informações a serem transmitidas, diversos métodos de compactação diferentes, tais como padrões de JPEG, MPEG e H.263, foram desenvolvidos.

[0027] Os padrões de codificação de vídeo incluem ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 ou ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (também chamado de ISO/IEC MPEG-4 AVC), incluindo suas extensões de Codificação de Vídeo Escalonável (SVC) e de Codificação de Vídeo de Múltiplas Vistas (MVC).

[0028] Além disso, um novo padrão de codificação de vídeo, a saber, a Codificação de Vídeo de

Alta Eficácia (HEVC) está sendo desenvolvido pela Equipe de Colaboração Conjunta em Codificação de Vídeo (JCT-VC) do Grupo de Especialistas de Codificação de Vídeo de ITU-T (VCEG) e do Grupo de Especialistas de Filme Cinematográfico de ISO/IEC (MPEG). A citação completa para o Rascunho 10 do HEVC (HEVC Draft 10) é o documento JCTVC-L1003, Bross et al, "High Efficiency Video Coding (HEVC) Text Specification Draft 10" Equipe Colaborativa Conjunta em codificação de Vídeo (JCT-VC) de ITU-T SGI 6 WP3 e ISO/IEC JTC1/SC29/WG11, 12º Encontro: Geneva, Suíça, 14 de janeiro de 2013 a 23 de janeiro de 2013. A extensão de múltiplas visualizações para HEVC, a saber, MV-HEVC e a extensão escalonável para HEVC, a saber, SHVC, também estão sendo desenvolvidas pela JCT-3V (ITU-T/ISO/IEC Equipe Colaborativa Conjunta em codificação de Vídeo em Desenvolvimento de Extensão em 3D) e JCT-VC, respectivamente.

[0029] Vários aspectos dos sistemas, aparelhos e métodos inovadores são descritos mais completamente a partir deste ponto no presente documento com referência aos desenhos anexos. Esta descrição pode, entretanto, ser incorporada de muitas formas diferentes e não deve ser interpretada como limitada a qualquer estrutura específica ou função apresentada ao longo de toda esta descrição. Em vez disso, esses aspectos são fornecidos de modo que essa descrição seja minuciosa e completa, e transmita plenamente o escopo da descrição para aqueles versados na técnica. Com base nos ensinamentos no presente documento, uma pessoa versada na técnica deve apreciar que o escopo da descrição está destinado a cobrir qualquer aspecto dos sistemas, aparelhos e métodos inovadores revelados no presente documento, sejam os mesmos implantados independentemente de, ou combinados com qualquer outro aspecto da descrição. Por exemplo, um aparelho pode ser implantado ou um método

pode ser praticado através do uso de qualquer quantidade dos aspectos apresentados no presente documento. Além disso, o escopo da presente descrição se destina a abranger tal aparelho ou método que é praticado com o uso de outra estrutura, funcionalidade ou estrutura e funcionalidade em adição a ou além dos vários aspectos da presente descrição apresentados no presente documento. Deve ser entendido que qualquer aspecto revelado no presente documento pode ser incorporado por um ou mais elementos de uma reivindicação.

[0030] Embora aspectos particulares sejam descritos no presente documento, muitas variações e permutações desses aspectos se encaixam no escopo da descrição. Embora alguns benefícios e vantagens dos aspectos preferenciais sejam mencionados, o escopo da descrição não está destinado a ser limitado a benefícios, usos ou objetivos particulares. Ao invés disso, os aspectos da descrição estão destinados a serem amplamente aplicáveis a diferentes tecnologias sem fio, configurações de sistema, redes e protocolos de transmissão, alguns dos quais são ilustrados a título de exemplo nas Figuras e na descrição a seguir dos aspectos preferenciais. A descrição detalhada e desenhos são meramente ilustrativos da descrição ao invés de limitantes, sendo que o escopo da descrição é definido pelas reivindicações anexas e equivalentes das mesmas.

[0031] OS desenhos anexos ilustram exemplos. Os elementos indicados por números de referência nos desenhos anexos correspondem a elementos indicados por números de referência similares na descrição a seguir. Nesta descrição, os elementos que têm nomes que começam com palavras ordinais (por exemplo, "primeiro", "segundo", "terceiro" e assim por diante) não necessariamente implicam que os elementos têm uma ordem particular. Ao invés disso, tais palavras ordinais são meramente usadas para se referir

a diferentes elementos de um tipo igual ou similar.

#### SISTEMA DE CODIFICAÇÃO DE VÍDEO

[0032] A Figura 1A é um diagrama de blocos que ilustra um sistema de codificação de vídeo exemplificativo 10 que pode utilizar as técnicas de acordo com os aspectos descritos nesta descrição. Conforme usado descrito no presente documento, o termo "codificador de vídeo" se refere genericamente tanto a codificadores de vídeo quanto a decodificadores de vídeo. Nesta descrição, os termos "codificação de vídeo" ou "codificação" podem se referir genericamente à codificação de vídeo e à decodificação de vídeo. Além de codificadores de vídeo e decodificadores de vídeo, os aspectos descritos no presente pedido podem ser estendidos a outros dispositivos relacionados, como transcodificadores (por exemplo, dispositivos que podem decodificar um fluxo de bits e re-codificar outro fluxo de bits) e dispositivos intermediários (por exemplo, dispositivos que podem modificar, transformar e/ou, de outro modo, manipular um fluxo de bits).

[0033] Conforme mostrado na Figura 1A, o sistema de codificação de vídeo 10 inclui um módulo de fonte 12 que gera dados de vídeo codificados a serem decodificados em um momento posterior por um módulo de destino 14. No exemplo da Figura 1A, o módulo de fonte 12 e o módulo de destino 14 estão em dispositivos separados - especificamente, o módulo de fonte 12 é parte de um dispositivo de fonte e o módulo de destino 14 é parte de um dispositivo de destino. No entanto, observa-se que os módulos de destino e de fonte 12, 14 podem estar em ou ser parte do mesmo dispositivo, conforme mostrado no exemplo da Figura 1B

[0034] Referindo-se novamente à Figura 1A, o módulo de fonte 12 e o módulo de destino 14 podem

compreender qualquer um dentre uma ampla faixa de dispositivos, que incluem computadores do tipo desktop, computadores do tipo notebook (por exemplo, computador do tipo laptop), computadores do tipo tablet, decodificadores de sinais, fones de telefone, tais como, os então chamados telefones "inteligentes", então chamados "smart" pads, televisões, câmeras, dispositivos de exibição, reprodutores de mídia digital, consoles de jogos eletrônicos, dispositivo de transmissão contínua de vídeo ou similares. Em alguns casos, o módulo de fonte 12 e o módulo de destino 14 podem ser equipados para comunicação sem fio.

[0035] O módulo de destino 14 pode receber os dados de vídeo codificados a serem decodificados através de uma ligação 16. A ligação 16 pode compreender qualquer tipo de meio ou dispositivo que tenha capacidade de mover os dados de vídeo codificados do módulo de fonte 12 para o módulo de destino 14. No exemplo da Figura 1A, o enlace 16 pode compreender um meio de comunicação para permitir que o módulo de fonte 12 transmita os dados de vídeo codificados diretamente para o módulo de destino 14 em tempo real. Os dados de vídeo codificados podem ser modulados de acordo com um padrão de comunicação como um protocolo de comunicação sem fio e transmitidos para o dispositivo de destino 14. O meio de comunicação pode compreender qualquer meio de comunicação sem fio ou com fio, como um espectro de radiofrequência (RF) ou uma ou mais linhas de transmissão físicas. A mídia de comunicação pode formar parte de uma rede com base em pacote, tal como, uma rede de área local, uma rede de longa distância ou uma rede global, tal como, a Internet. O meio de comunicação pode incluir roteadores, comutadores, estações-base ou qualquer outro equipamento que possa ser útil para facilitar a comunicação a partir do dispositivo de fonte 12 para o dispositivo de destino 14.

[0036] Alternativamente, dados codificados podem ser emitidos a partir de uma interface de saída 22 para um dispositivo de armazenamento opcional 31. De modo similar, os dados codificados podem ser acessados a partir do dispositivo de armazenamento 31 pela interface de entrada 28. O dispositivo de armazenamento 31 pode incluir qualquer um dentre uma variedade de meios de armazenamento de dados acessados localmente ou distribuídos tais como um disco rígido, memória rápida, memória volátil ou não volátil ou quaisquer outros meios de armazenamento digital adequados para armazenar dados de vídeo codificados. Em um exemplo adicional, o dispositivo de armazenamento 31 pode corresponder a um servidor de arquivos ou outro dispositivo de armazenamento intermediário que pode armazenar o vídeo codificado gerado pelo dispositivo de fonte 12. O módulo de destino 14 pode acessar dados de vídeo armazenados a partir do dispositivo de armazenamento 31 através de transmissão contínua ou transferência por download. O servidor de arquivos pode ser qualquer tipo de servidor com capacidade para armazenar dados de vídeo codificados e transmitir esses dados de vídeo codificados para o módulo de destino 14. Os servidores de arquivo exemplificativos incluem um servidor web (por exemplo, para um site da web), um servidor FTP, dispositivos de armazenamento anexado à rede (NAS) ou uma unidade de disco local. O módulo de destino 14 pode acessar os dados de vídeo codificados através de qualquer conexão de dados padrão, que inclui uma conexão com a Internet. Isso pode incluir um canal sem fio (por exemplo, uma conexão Wi-Fi), uma conexão com fio (por exemplo, DSL, modem de cabo, etc.) ou uma combinação de ambos que seja adequada para acessar dados de vídeo codificados armazenados em um servidor de arquivos. A transmissão de dados de vídeo codificados do dispositivo de

armazenamento 31 pode ser uma transmissão contínua, uma transmissão de transferência por download ou uma combinação das mesmas.

[0037] As técnicas desta descrição não estão necessariamente limitadas a aplicações ou definições sem fio. As técnicas podem ser aplicadas à codificação de vídeo em suporte a qualquer variedade de aplicações multimídia, tais como difusões de televisão pelo ar, transmissões de televisão a cabo, transmissões de televisão via satélite, transmissão contínua de vídeo, por exemplo, através da Internet (por exemplo, transmissão contínua adaptativa dinâmica através de HTTP (DASH), etc.), codificação de vídeo digital para armazenamento em um meio de armazenamento de dados, decodificação de vídeo digital armazenado em um meio de armazenamento de dados ou outros aplicativos. Em alguns exemplos, o sistema de codificação de vídeo 10 pode ser configurado para suportar transmissão de vídeo unidirecional ou bidirecional para suportar aplicações tais como transmissão contínua de vídeo, reprodução de vídeo, difusão de vídeo e/ou telefonia por vídeo.

[0038] No exemplo da Figura 1A, o módulo de fonte 12 inclui uma fonte de vídeo 18, um codificador de vídeo 20 e uma interface de saída 22. Em alguns casos, a interface de saída 22 pode incluir um modulador/demodulador (modem) e/ou um transmissor. No módulo de fonte 12, a fonte de vídeo 18 pode incluir uma fonte tal como um dispositivo de captura de vídeo, por exemplo, uma câmera de vídeo, um arquivo morto de vídeo que contém vídeo anteriormente capturado, uma interface de alimentação de vídeo para receber vídeo a partir de um provedor de conteúdo de vídeo e/ou um sistema de gráficos de computador para gerar dados de gráficos de computador como o vídeo de origem ou uma

combinação de tais fontes. Como um exemplo, se a fonte de vídeo 18 é uma câmera de vídeo, o módulo de fonte 12 e o módulo de destino 14 podem formar os chamados telefones com câmera fotográfica ou telefones com câmera de vídeo, conforme ilustrado no exemplo da Figura 1B. Entretanto, as técnicas descritas nesta descrição podem ser aplicáveis à codificação de vídeo em geral e podem ser aplicadas a aplicações com fio e/ou sem fio.

[0039] O vídeo gerado por computador, pré-capturado ou capturado pode ser codificado pelo codificador de vídeo 20. Os dados de vídeo codificados podem ser transmitidas diretamente para o módulo de destino 14 através da interface de saída 22 do módulo de fonte 12. Os dados de vídeo codificados também podem (ou alternativamente) ser armazenados no dispositivo de armazenamento 31 para acesso posterior pelo módulo de destino 14 ou outros dispositivos para decodificação e/ou reprodução. O codificador de vídeo 20 ilustrado na Figura 1A e 1B pode compreender o codificador de vídeo 20 ilustrado na Figura 2A, o codificador de vídeo 23 ilustrado na Figura 2B ou qualquer outro codificador de vídeo descrito no presente documento.

[0040] No exemplo da Figura 1A, o módulo de destino 14 inclui uma interface de entrada 28, um decodificador de vídeo 30 e um dispositivo de exibição 32. Em alguns casos, a interface de entrada 28 pode incluir um receptor e/ou um modem. A interface de entrada 28 do módulo de destino 14 pode receber os dados de vídeo codificados através do enlace 16. Os dados de vídeo codificados comunicados através da ligação 16 ou fornecidos no dispositivo de armazenamento 31 podem incluir uma variedade de elementos de sintaxe gerados pelo codificador de vídeo 20 para uso por um decodificador de vídeo, tal como o

decodificador de vídeo 30, na decodificação dos dados de vídeo. Tais elementos de sintaxe podem ser incluídos com os dados de vídeo codificados transmitidos em um meio de comunicação, armazenados em um meio de armazenamento ou armazenado em um servidor de arquivos. O decodificador de vídeo 30 ilustrado na Figura 1A e 1B pode compreender o decodificador de vídeo 30 ilustrado na Figura 3A, o decodificador de vídeo 33 ilustrado na Figura 3B ou qualquer outro decodificador de vídeo descrito no presente documento.

[0041] O dispositivo de exibição 32 pode ser integrado ou externo ao módulo de destino 14. Em alguns exemplos, o módulo de destino 14 pode incluir um dispositivo de exibição integrado e também pode ser configurado para fazer interface com um dispositivo de exibição externa. Em outros exemplos, o módulo de destino 14 pode ser um dispositivo de exibição. Em geral, o dispositivo de exibição 32 exibe os dados de vídeo decodificados para um usuário e pode compreender qualquer um dentre uma variedade de dispositivos de exibição tal como um visor de cristal líquido (LCD), um visor de plasma, um visor de diodo emissor de luz orgânico (OLED) ou um outro tipo de dispositivo de exibição.

[0042] Nos aspectos relacionados, a Figura 1B mostra um sistema de decodificação e codificação de vídeo exemplificativo 10' em que os módulos de destino e de fonte 12, 14 estão em ou são parte de um dispositivo ou dispositivo de usuário 11. O dispositivo 11 pode ser um fone de telefone, tal como um telefone "inteligente" ou similares. O dispositivo 11 pode incluir um módulo de processador/controlador opcional 13 em comunicação operativa com os módulos de fonte e de destino 12, 14. O sistema 10' da Figura 1B pode incluir adicionalmente uma

unidade de processamento de vídeo 21 entre o codificador de vídeo 20 e a interface de saída 22. Em algumas implantações, a unidade de processamento de vídeo 21 é uma unidade separada, conforme ilustrado na Figura 1B; no entanto, em outras implantações, a unidade de processamento de vídeo 21 pode ser implantada como uma porção do codificador de vídeo 20 e/ou do módulo de processador/controlador 13. O sistema 10' também pode incluir um rastreador opcional 29, que pode rastrear um objeto de interesse em uma sequência de vídeo. O objeto de interesse a ser rastreado pode ser segmentado por uma técnica descrita em conexão com um ou mais aspectos da presente descrição. Nos aspectos relacionados, o rastreamento pode ser realizado pelo dispositivo de exibição 32, sozinho ou em conjunto com o rastreador 29. O sistema 10' da Figura 1B e os componentes do mesmo, são, de outro modo, similares ao sistema 10 da Figura 1A e componentes do mesmo.

[0043] O codificador de vídeo 20 e o decodificador de vídeo 30 podem operar de acordo com um padrão de compactação de vídeo, tal como, um padrão de Codificação de Vídeo de Alta Eficiência (HEVC) e pode se conformar a um Modelo de Teste HEVC (HM). Alternativamente, o codificador de vídeo 20 e o decodificador de vídeo 30 podem operar de acordo com outros padrões de proprietário ou de indústria, como o padrão ITU-T H.264, referido de modo alternativo como MPEG-4, Parte 10, Codificação de Vídeo Avançada (AVC), ou extensões de tais padrões. Entretanto, as técnicas desta descrição não são limitadas a qualquer padrão de codificação particular. Outros exemplos de padrões de codificação de vídeo incluem MPEG-2 e ITU-T H.263.

[0044] Embora não mostrados nos exemplos da

Figura 1, em alguns aspectos, o codificador de vídeo 20 e o decodificador de vídeo 30 podem ser, cada um, integrados a um codificador e decodificador de áudio e podem incluir unidades MUX-DEMUX apropriadas, ou outro hardware e software, para lidar com a codificação tanto de áudio quanto de vídeo em uma corrente de dados comum ou correntes de dados separadas. Caso aplicável, as unidades MUX-DEMUX podem se conformar ao protocolo multiplexador ITU H.223, ou outros protocolos como protocolo de datagrama de usuário (UDP).

[0045] O codificador de vídeo 20 e o decodificador de vídeo 30, cada um, podem ser implantados como qualquer um dentre uma variedade de conjunto de circuitos de codificador adequado, como um ou mais microprocessadores, processadores de sinal digital (DSPs), circuitos integrados de aplicação específica (ASICs), matrizes de portal programáveis por campo (FPGAs), lógica discreta, software, hardware, firmware ou quaisquer combinações dos mesmos. Quando as técnicas são implantadas parcialmente em software, um dispositivo pode armazenar instruções para o software em um meio legível por computador não transitório adequado e executar as instruções em hardware com o uso de um ou mais processadores para realizar as técnicas desta descrição. Cada um dentre o codificador de vídeo 20 e o decodificador de vídeo 30 podem estar incluídos em um ou mais codificadores ou decodificadores, qualquer um dos quais pode ser integrado como parte de um codificador/decodificador (CODEC) combinado em um respectivo dispositivo.

#### PROCESSO DE CODIFICAÇÃO DE VÍDEO

[0046] Conforme mencionado brevemente acima, o codificador de vídeo 20 codifica os dados de vídeo. Os

dados de vídeo podem compreender uma ou mais imagens. Cada uma das imagens é uma imagem estática que forma parte de um vídeo. Em alguns casos, uma imagem pode ser denominada como um “quadro” de vídeo. Quando o codificador de vídeo 20 codifica os dados de vídeo, o codificador de vídeo 20 pode gerar um fluxo de bits. O fluxo de bits pode incluir uma sequência de bits que forma uma representação codificada dos dados de vídeo. O fluxo de bits pode incluir imagens codificadas e dados associados. Uma imagem codificada é uma representação codificada de uma imagem.

[0047] Para gerar o fluxo de bits, o codificador de vídeo 20 pode realizar operações de codificação em cada imagem nos dados de vídeo. Quando o codificador de vídeo 20 realiza as operações de codificação nas imagens, o codificador de vídeo 20 pode gerar uma série de imagens codificadas e dados associados. Os dados associados podem incluir conjuntos de parâmetros de vídeo (VPS), conjuntos de parâmetros de sequência, conjuntos de parâmetros de imagem, conjuntos de parâmetros de adaptação e outras estruturas de sintaxe. Um conjunto de parâmetros de sequência (SPS) pode conter parâmetros aplicáveis a zero ou mais sequências de imagens. Um parâmetro de imagem definido (PPS) pode conter parâmetros aplicáveis a zero ou mais imagens. Um parâmetro de adaptação definido (APS) pode conter parâmetros aplicáveis a zero ou mais imagens. Os parâmetros em um APS podem ser parâmetros que têm maior probabilidade de mudar do que parâmetros em um PPS.

[0048] Para gerar uma imagem codificada, o codificador de vídeo 20 pode particionar uma imagem em blocos de vídeo igualmente dimensionados. Um bloco de vídeo pode ser um arranjo bidimensional de amostras. Cada um dos blocos de vídeo está associado a um bloco em árvore. Em algumas instâncias, um bloco em árvore pode ser referido

como uma unidade de codificação maior (LCU). Os blocos em árvore de HEVC podem ser amplamente analógicos aos macroblocos de padrões anteriores, tais como, H.264/AVC. No entanto, um bloco em árvore não se limita necessariamente a um tamanho particular e pode incluir uma ou mais unidades de codificação (CUs). O codificador de vídeo 20 pode usar partição em árvore quaternária para particionar os blocos de vídeo de blocos em árvore em blocos de vídeo associados às CUs, por isso o nome "blocos em árvore".

[0049] Em alguns exemplos, o codificador de vídeo 20 pode particionar uma imagem em uma pluralidade de fatias. Cada uma das fatias pode incluir um número inteiro de CUs. Em alguns casos, uma fatia compreende um número inteiro de blocos em árvore. Em outros casos, uma fronteira de uma fatia pode estar dentro de um bloco em árvore.

[0050] Como parte da realização de uma operação de codificação em uma imagem, o codificador de vídeo 20 pode realizar operações de codificação em cada fatia da imagem. Quando o codificador de vídeo 20 realiza uma operação de codificação em uma fatia, o codificador de vídeo 20 pode gerar dados codificados associados com a fatia. Os dados codificados associados à fatia podem ser denominados como "fatia codificada".

[0051] Para gerar uma fatia codificada, o codificador de vídeo 20 pode realizar operações de codificação em cada bloco em árvore em uma fatia. Quando o codificador de vídeo 20 realiza uma operação de codificação em um bloco em árvore, o codificador de vídeo 20 pode gerar um bloco em árvore codificado. O bloco em árvore codificado pode compreender dados que representam uma versão codificada do bloco em árvore.

[0052] Quando o codificador de vídeo 20 gera uma fatia codificada, o codificador de vídeo 20 pode

realizar operações de codificação (por exemplo, codificar) nos blocos em árvore na fatia de acordo com uma ordem de varredura. Por exemplo, o codificador de vídeo 20 pode codificar os blocos em árvore da fatia em uma ordem que procede da esquerda para a direita e atravessa uma fileira mais alta de blocos em árvore na fatia, depois da esquerda para a direita e atravessa uma próxima fileira mais baixa de blocos em árvore e assim por diante até que o codificador de vídeo 20 tenha codificado cada um dos blocos em árvore na fatia.

[0053] Como resultado da codificação dos blocos em árvore de acordo com a ordem de varredura por rastreio, os blocos em árvore acima e à esquerda de um determinado bloco em árvore pode ter sido codificado, mas os blocos em árvore abaixo e à direita do determinado bloco em árvore ainda não foram codificados. Consequentemente, o codificador de vídeo 20 pode ter capacidade para acessar informações geradas pela codificação de blocos em árvore acima e à esquerda do determinado bloco em árvore quando é feita a codificação do determinado bloco em árvore. Contudo, o codificador de vídeo 20 pode não ter capacidade de acessar informações geradas pela codificação de blocos em árvore abaixo e à direita do determinado bloco em árvore quando é feita a codificação do determinado bloco em árvore.

[0054] Para gerar um bloco em árvore codificado, o codificador de vídeo 20 pode recursivamente realizar o particionamento em árvore quaternária no bloco de vídeo do bloco em árvore para dividir o bloco de vídeo em blocos de vídeo progressivamente menores. Cada um dos blocos de vídeo menores pode ser associado a uma CU diferente. Por exemplo, o codificador de vídeo 20 pode particionar o bloco de vídeo de um bloco em árvore em

quatro sub-blocos de igual tamanho, particionar um ou mais dos sub-blocos em quatro sub-blocos de igual tamanho e assim por diante. Uma CU particionada pode ser uma CU cujo bloco de vídeo é particionado em blocos de vídeo associados com outras CUs. Uma CU não particionada pode ser uma CU cujo bloco de vídeo não é particionado em blocos de vídeo associados com outras CUs.

[0055] Um ou mais elementos de sintaxe no fluxo de bits podem indicar um número máximo de vezes que o codificador de vídeo 20 pode particionar o bloco de vídeo de um bloco em árvore. Um bloco de vídeo de uma CU pode ter formato quadrado. O tamanho do bloco de vídeo de uma CU (por exemplo, o tamanho da CU) pode variar de 8x8 pixels até o tamanho de um bloco de vídeo de um bloco em árvore (por exemplo, o tamanho do bloco em árvore) com um máximo de 64x64 pixels ou mais.

[0056] O codificador de vídeo 20 pode realizar operações de codificação em (por exemplo, codificar) cada CU de um bloco em árvore de acordo com uma ordem de varredura-z. Em outras palavras, o codificador de vídeo 20 pode codificar uma CU de topo à esquerda, uma CU de topo à direita, uma CU de fundo à esquerda e depois uma CU de fundo à direita CU, nessa ordem. Quando o codificador de vídeo 20 realiza uma operação de codificação em uma CU particionada, o codificador de vídeo 20 pode codificar CUs associadas a sub-blocos do bloco de vídeo da CU particionada de acordo com a ordem de varredura-z. Em outras palavras, o codificador de vídeo 20 pode codificar uma CU associada a um sub-bloco de topo à esquerda, uma CU associada a um sub-bloco de topo à direita, uma CU associada a um sub-bloco de fundo à esquerda e depois uma CU associada a um sub-bloco de fundo à direita, nessa ordem.

[0057] Como resultado da codificação das CUs de um bloco em árvore de acordo com uma ordem de varredura-z, as CUs acima, acima e à esquerda, acima e à direita, esquerda e abaixo e à esquerda de uma dada CU podem ter sido codificadas. As CUs abaixo e à direita da dada CU ainda não foram codificadas. Consequentemente, o codificador de vídeo 20 pode ter capacidade para acessar as informações geradas pela codificação de algumas CUs próximas da determinada CU ao codificar a determinada CU. No entanto, o codificador de vídeo 20 pode não ter capacidade para acessar as informações geradas pela codificação de outras CUs próximas da determinada CU ao codificar a determinada CU.

[0058] Quando o codificador de vídeo 20 codifica uma CU não particionada, o codificador de vídeo 20 pode gerar uma ou mais unidades de predição (PUs) para a CU. Cada uma das PUs da CU pode estar associada a um bloco de vídeo diferente dentro do bloco de vídeo da CU. O codificador de vídeo 20 pode gerar um bloco de vídeo preditivo para cada PU da CU. O bloco de vídeo preditivo de uma PU pode ser um bloco de amostras. O codificador de vídeo 20 pode usar intrapredição ou interpredição para gerar os blocos de vídeo preditos para uma PU.

[0059] Se o codificador de vídeo 20 usar intrapredição para gerar os blocos preditos de uma PU, o codificador de vídeo 20 pode gerar os blocos preditos da PU com base nas amostras decodificadas da imagem associada à PU. Se o codificador de vídeo 20 usar intrapredição para gerar blocos de vídeo preditos das PUs de uma CU, a CU é uma CU intraprevista. Quando o codificador de vídeo 20 usa uma interpredição para gerar os blocos preditos de uma PU, o codificador de vídeo 20 pode gerar o bloco previsto da PU com base nas amostras decodificadas de uma ou mais imagens

diferentes da imagem associada à PU. Se o codificador de vídeo 20 usar interpredação para gerar os blocos de vídeo preditos das PUs de uma CU, a CU é uma CU interprevista.

[0060] Além disso, quando o codificador de vídeo 20 usa interpredação para gerar um bloco de vídeo previsto para uma PU, o codificador de vídeo 20 pode gerar informações de movimento para a PU. As informações de movimento para uma PU podem indicar um ou mais blocos de referência da PU. Cada bloco de referência da PU pode ser um bloco de vídeo dentro de uma imagem de referência. A imagem de referência pode ser uma imagem diferente da imagem associada com a PU. Em alguns casos, um bloco de referência de uma PU também pode ser chamado de "amostra de referência" da PU. O codificador de vídeo 20 pode gerar o bloco de vídeo previsto para a PU com base nos blocos de referência da PU.

[0061] Após o codificador de vídeo 20 gerar blocos de vídeo preditos para uma ou mais PUs de uma CU, o codificador de vídeo 20 pode gerar dados residuais para a CU com base nos blocos de vídeo preditos para as PUs da CU. Os dados residuais para a CU podem indicar diferenças entre as amostras nos blocos de vídeo preditos para as PUs da CU e o bloco de vídeo original da CU.

[0062] Além disso, como parte da realização de uma operação de codificação em uma CU não particionada, o codificador de vídeo 20 pode realizar particionamento em árvore quaternária recursivo nos dados residuais da CU para particionar os dados residuais da CU em um ou mais blocos de dados residuais (por exemplo, blocos de vídeo residuais) associados às unidades de transformada (TUs) da CU. Cada TU de uma CU pode estar associada a um bloco de vídeo residual diferente.

[0063] O codificador de vídeo 20 pode aplicar

uma ou mais transformadas a blocos de vídeo residuais associados às TUs para gerar blocos de coeficiente de transformada (por exemplo, blocos de coeficientes de transformada) associados às TUs. Conceitualmente, um bloco de coeficiente de transformada pode ser uma matriz bidimensional (2D) de coeficientes de transformada.

[0064] Após gerar um bloco de coeficiente de transformada, o codificador de vídeo 20 pode realizar um processo de quantização no bloco de coeficiente de transformada. Em geral, a quantização se refere a um processo no qual os coeficientes de transformada são quantizados para possivelmente reduzir a quantidade de dados usados para representar os coeficientes de transformada, o que fornece uma compactação adicional. O processo de quantização pode reduzir a profundidade de bit associada ao mesmo coeficiente de transformada ou a todos os coeficientes de transformada. Por exemplo, um coeficiente de transformada de bit  $w$  pode ser arredondado para baixo para um coeficiente de transformada de bit  $m$  durante a quantização, em que  $n$  é maior do que  $m$ .

[0065] O codificador de vídeo 20 pode associar cada CU a um valor de parâmetro de quantização (QP). O valor de QP associado a uma CU pode determinar como o codificador de vídeo 20 quantiza os blocos de coeficiente de transformada associados à CU. O codificador de vídeo 20 pode ajustar o grau de quantização aplicado aos blocos de coeficiente de transformada associados a uma CU através do ajuste do valor de QP associado a CU.

[0066] Após o codificador de vídeo 20 quantizar um bloco de coeficiente de transformada, o codificador de vídeo 20 pode gerar conjuntos de elementos de sintaxe que representam os coeficientes de transformada no bloco de coeficiente de transformada quantizado. O

codificador de vídeo 20 pode aplicar operações de codificação por entropia, tais como operações de Codificação Aritmética Binária Adaptativa ao Contexto (CABAC) a alguns desses elementos de sintaxe. Outras técnicas de codificação por entropia tais como codificação de comprimento variável adaptativa ao conteúdo (CAVLC), codificação por entropia com particionamento de probabilidade de intervalo (PIPE) ou outra codificação aritmética binária também poderiam ser usadas.

[0067] O fluxo de bits gerado pelo codificador de vídeo 20 pode incluir uma série de unidades de Camada de Abstração de Rede (NAL). Cada uma das unidades de NAL pode ser uma estrutura de sintaxe que contém uma indicação de um tipo de dados na unidade de NAL e bits que contêm os dados. Por exemplo, uma unidade de NAL pode conter dados que representam um parâmetro de vídeo definido, um parâmetro de sequência definido, um parâmetro de imagem definido, uma fatia codificada, informações de aperfeiçoamento suplementares (SEI), um delimitador de unidade de acesso, dados de preenchedor ou outro tipo de dados. Os dados em uma unidade de NAL podem incluir várias estruturas de sintaxe.

[0068] O decodificador de vídeo 30 pode receber o fluxo de bits gerado pelo codificador de vídeo 20. O fluxo de bits pode incluir uma representação codificada dos dados de vídeo codificados pelo codificador de vídeo 20. Quando o decodificador de vídeo 30 recebe o fluxo de bits, o decodificador de vídeo 30 pode realizar uma operação de análise no fluxo de bits. Quando o decodificador de vídeo 30 realiza a operação de análise, o decodificador de vídeo 30 pode extrair elementos de sintaxe do fluxo de bits. O decodificador de vídeo 30 pode reconstruir as imagens dos dados de vídeo com base nos

elementos de sintaxe extraídos do fluxo de bits. O processo para reconstruir os dados de vídeo com base nos elementos de sintaxe pode ser genericamente recíproco ao processo realizado pelo codificador de vídeo 20 para gerar os elementos de sintaxe.

[0069] Após o decodificador de vídeo 30 extrair os elementos de sintaxe associados a uma CU, o decodificador de vídeo 30 pode gerar blocos de vídeo preditos para as PUs da CU com base nos elementos de sintaxe. Além disso, o decodificador de vídeo 30 pode quantizar de modo inverso os coeficientes de blocos associados às TUs da CU atual. O decodificador de vídeo 30 pode realizar transformadas inversas nos blocos de coeficiente de transformada para reconstruir blocos de vídeo residuais associados às TUs da CU. Após gerar os blocos de vídeo preditos e reconstruir os blocos de vídeo residuais, o decodificador de vídeo 30 pode reconstruir o bloco de vídeo da CU com base nos blocos de vídeo preditos e nos blocos de vídeo residuais. Desse modo, o decodificador de vídeo 30 pode reconstruir os blocos de vídeo de CUs com base nos elementos de sintaxe no fluxo de bits.

#### CODIFICADOR DE VÍDEO

[0070] A Figura 2A é um diagrama de blocos que ilustra um exemplo de um codificador de vídeo que pode implantar técnicas de acordo com os aspectos descritos nesta descrição. O codificador de vídeo 20 pode ser configurado para processar uma única camada de um quadro de vídeo, tal como para HEVC. Adicionalmente, o codificador de vídeo 20 pode ser configurado para realizar qualquer ou todas as técnicas desta descrição. Como um exemplo, a unidade de processamento de predição 100 pode ser configurada para realizar qualquer ou todas as técnicas

descritas nesta descrição. Em uma outra modalidade, o codificador de vídeo 20 inclui uma unidade prevista de intercamada opcional 128 que é configurada para realizar qualquer ou todas as técnicas descritas nesta descrição. Em outras modalidades, a predição de intercamada pode ser realizada pela unidade de processamento de predição 100 (por exemplo, unidade de interpredição 121 e/ou unidade de intrapredição 126), no caso em que a unidade prevista intercamada 128 pode ser omitida. Entretanto, aspectos dessa descrição não são tão limitados. Em alguns exemplos, as técnicas descritas nesta descrição podem ser compartilhadas entre os vários componentes de codificador de vídeo 20. Em alguns exemplos, adicional ou alternativamente, um processador (não mostrado) pode ser configurado para realizar qualquer uma ou todas as técnicas descritas nesta descrição.

[0071] Para propósitos explicativos, esta descrição descreve o codificador de vídeo 20 no contexto de codificação de HEVC. No entanto, as técnicas dessa descrição podem ser aplicáveis a outros padrões ou métodos de codificação. O exemplo representado na Figura 2A é para um codec de camada única. No entanto, conforme será descrito adicionalmente em relação à Figura 2B, algum ou todo o codificador de vídeo 20 pode ser duplicado para processar um codec de camada múltipla.

[0072] O codificador de vídeo 20 pode realizar intra e intercodificação de blocos de vídeo dentro de fatias de vídeo. A intracodificação depende da predição espacial para reduzir ou remover a redundância espacial no vídeo dentro de uma determinada imagem de quadro de vídeo. A intercodificação tem base na predição temporal para reduzir ou remover a redundância temporal no vídeo dentro de quadros ou imagens adjacentes de uma sequência de vídeo.

O intramodo (modo I) pode se referir a qualquer um dentre vários modos de codificação com base em espaço. Os intermodos, tal como predição monodirecional (modo P) ou predição bidirecional (modo B), podem se referir a qualquer um dentre diversos modos de codificação à base temporal.

[0073] No exemplo da Figura 2A, o codificador de vídeo 20 inclui uma pluralidade de componentes funcionais. Os componentes funcionais de codificador de vídeo 20 inclui uma unidade de processamento de predição 100, uma unidade de geração residual 102, uma unidade de processamento de transformada 104, uma unidade de quantização 106, uma unidade de quantização inversa 108, uma unidade de transformada inversa 110, uma unidade de reconstrução 112, uma unidade de filtro 113, um buffer de imagem decodificada 114 e uma unidade de codificação por entropia 116. A unidade de processamento de predição 100 inclui uma unidade de interpredição 121, uma unidade de estimativa de movimento 122, uma unidade de compensação de movimento 124, uma unidade de intrapredição 126 e uma unidade prevista intercamada 128. Em outros exemplos, o codificador de vídeo 20 pode incluir mais, poucos ou diferentes componentes funcionais. Além disso, a unidade de estimativa de movimento 122 e a unidade de compensação de movimento 124 podem ser altamente integradas, mas são representadas no exemplo da Figura 2A separadamente com propósitos explicativos.

[0074] O codificador de vídeo 20 pode receber os dados de vídeo. O codificador de vídeo 20 pode receber os dados de vídeo de várias fontes. Por exemplo, o codificador de vídeo 20 pode receber os dados de vídeo a partir da fonte de vídeo 18 (por exemplo, mostrada na Figura 1A ou 1B) ou uma outra fonte. Os dados de vídeo podem representar uma série de imagens. Para codificar os

dados de vídeo, o codificador de vídeo 20 pode realizar uma operação de codificação em cada uma das imagens. Como parte da realização da operação de codificação em uma imagem, o codificador de vídeo 20 pode realizar operações de codificação em cada fatia da imagem. Como parte da realização de uma operação de codificação em uma fatia, o codificador de vídeo 20 pode realizar operações de codificação em blocos em árvore na fatia.

[0075] Como parte da realização de uma operação de codificação em um bloco em árvore, a unidade de processamento de predição 100 pode realizar particionamento de árvore quaternária no bloco de vídeo do bloco em árvore para dividir o bloco de vídeo em blocos de vídeo progressivamente menores. Cada um dos blocos de vídeo menores pode ser associado a uma CU diferente. Por exemplo, a unidade de processamento de predição 100 pode particionar um bloco de vídeo de um bloco em árvore em quatro sub-blocos igualmente dimensionados, separar um ou mais dos sub-blocos em quatro sub-blocos igualmente dimensionados e assim por diante.

[0076] Os tamanhos dos blocos de vídeo associados à CUs podem variar de amostras de 8x8 até o tamanho do bloco em árvore com um máximo de amostras de 64x64 ou maior. Nesta descrição, "NxN" e "N por N" podem ser usados de modo intercambiável para se referir às dimensões de amostra de um bloco de vídeo no que diz respeito às dimensões horizontal e vertical, por exemplo, amostras de 16x16 ou amostras de 16 por 16. Em geral, um bloco de vídeo de 16x16 tem dezesseis amostras em uma direção vertical ( $y = 16$ ) e dezesseis amostras em uma direção horizontal ( $x = 16$ ). De modo semelhante, um bloco NxN geralmente tem N amostras em uma direção vertical e N amostras em uma direção horizontal, em que N representa um

valor de número inteiro não negativo.

[0077] Além disso, como parte da realização da operação de codificação em um bloco em árvore, a unidade de processamento de predição 100 pode gerar uma estrutura de dados em árvore quaternária hierárquica para o bloco em árvore. Por exemplo, um bloco em árvore pode corresponder a um nó raiz da estrutura de dados em árvore quaternária. Se a unidade de processamento de predição 100 particionar o bloco de vídeo do bloco em árvore em quatro sub-blocos, o nó raiz tem quatro nós filhos na estrutura de dados em árvore quaternária. Cada um dos nós filhos corresponde a uma CU associada a um dos sub-blocos. Se a unidade de processamento de predição 100 particionar um dos sub-blocos em quatro sub-sub-blocos, o nó correspondente à CU associada ao sub-bloco pode ter quatro nós filhos, sendo que cada um dos quais corresponde a uma CU associada a um dos sub-sub-blocos.

[0078] Cada nó da estrutura de dados em árvore quaternária pode conter dados de sintaxe (por exemplo, elementos de sintaxe) para o bloco em árvore ou CU correspondente. Por exemplo, um nó na árvore quaternária pode incluir um sinalizador de divisão que indica se o bloco de vídeo da CU correspondente ao nó é particionado (por exemplo, dividido) em quatro sub-blocos. Os elementos de sintaxe para uma CU podem ser definidos recursivamente e podem depender de se o bloco de vídeo da CU está dividido em sub-blocos. Uma CU cujo bloco de vídeo não é particionado pode corresponder a um nó folha na estrutura de dados em árvore quaternária. Um bloco em árvore codificado pode incluir dados com base na estrutura de dados em árvore quaternária para um bloco em árvore correspondente.

[0079] O codificador de vídeo 20 pode realizar

operações de codificação em cada CU não particionada de um bloco em árvore. Quando o codificador de vídeo 20 realiza uma operação de codificação em uma CU não particionada, o codificador de vídeo 20 gera dados que representam uma representação codificada da CU não particionada.

[0080] Como parte da realização de uma operação de codificação em uma CU, a unidade de processamento de predição 100 pode particionar o bloco de vídeo da CU dentre uma ou mais PUs da CU. O codificador de vídeo 20 e o decodificador de vídeo 30 podem suportar vários tamanhos de PU. Presumindo-se que o tamanho de uma CU particular é  $2Nx2N$ , o codificador de vídeo 20 e o decodificador de vídeo 30 podem suportar tamanhos de PU de  $2Nx2N$  ou  $NxN$  e interpredação em tamanhos de PU simétricos de  $2Nx2N$ ,  $2NxN$ ,  $Nx2N$ ,  $NxN$ ,  $2NxN$ ,  $nLx2N$ ,  $nRx2N$  ou similar. O codificador de vídeo 20 e o decodificador de vídeo 30 podem também suportar particionamento assimétrico para tamanhos de PU de  $2NxN$ ,  $2NxN$ ,  $nLx2N$  e  $nRx2N$ . Em alguns exemplos, a unidade de processamento de predição 100 pode realizar particionamento geométrico para particionar o bloco de vídeo de uma CU dentre PUs da CU ao longo de uma fronteira que não encontra os lados do bloco de vídeo da CU em ângulos retos.

[0081] A unidade de interpredação 121 pode realizar interpredação em cada PU da CU. A interpredação pode fornecer compactação temporal. Para realizar a interpredação em uma PU, a unidade de estimativa de movimento 122 pode gerar informações de movimento para a PU. A unidade de compensação de movimento 124 pode gerar um bloco de vídeo previsto para a PU com base nas informações de movimento e amostras decodificadas das imagens diferentes da imagem associada à CU (por exemplo, imagens de referência). Nesta descrição, um bloco de vídeo previsto

gerado pela unidade de compensação de movimento 124 pode ser denominado como bloco de vídeo interprevisto.

[0082] As fatias podem ser fatias I, fatias P ou fatias B. A unidade de estimativa de movimento 122 e a unidade de compensação de movimento 124 podem realizar diferentes operações para uma PU de uma CU dependendo se a PU está em uma fatia I, uma fatia P ou uma fatia B. Em uma fatia I, todas as PUs são intraprevistas. Por conseguinte, se a PU estiver em uma fatia I, a unidade de estimativa de movimento 122 e a unidade de compensação de movimento 124 não realizam interpredação na PU.

[0083] Se a PU estiver em uma fatia P, a imagem que contém a PU está associada a uma lista de imagens de referência referida como "lista 0". Cada uma das imagens de referência na lista 0 contém amostras que podem ser usadas para interpredação de outras imagens. Quando a unidade de estimativa de movimento 122 realiza a operação de estimativa de movimento em relação a uma PU em uma fatia P, a unidade de estimativa de movimento 122 pode procurar as imagens de referência na lista 0 para um bloco de referência para a PU. O bloco de referência da PU pode ser um conjunto de amostras, por exemplo, um bloco de amostras, que corresponde de modo mais aproximado às amostras no bloco de vídeo da PU. A unidade de estimativa de movimento 122 pode usar uma variedade de métricas para determinar o quão próximo um conjunto de amostras em uma imagem de referência corresponde às amostras no bloco de vídeo de uma PU. Por exemplo, a unidade de estimativa de movimento 122 pode determinar o quão próximo um conjunto de amostras em uma imagem de referência corresponde às amostras no bloco de vídeo de uma PU pela soma da diferença absoluta (SAD), soma das diferenças de quadrados (SSD) ou outras métricas de diferença.

[0084] Após identificar um bloco de referência de uma PU em uma fatia P, a unidade de estimativa de movimento 122 pode gerar um índice de referência que indica a imagem de referência na lista 0 que contém o bloco de referência e um vetor de movimento que indica um deslocamento espacial entre a PU e o bloco de referência. Em vários exemplos, a unidade de estimativa de movimento 122 pode gerar vetores de movimento para graus de precisão variáveis. Por exemplo, a unidade de estimativa de movimento 122 pode gerar vetores de movimento com precisão de amostra de um quarto, precisão de amostra de um oitavo ou outra precisão de amostra fracionária. No caso de precisão de amostra fracionária, os valores do bloco de referência podem ser interpolados a partir de valores de amostra com posição de número inteiro na imagem de referência. A unidade de estimativa de movimento 122 pode emitir o índice de referência e o vetor de movimento como as informações de movimento da PU. A unidade de compensação de movimento 124 pode gerar um bloco de vídeo previsto da PU com base no bloco de referência identificado pelas informações de movimento da PU.

[0085] Se a PU estiver em uma fatia B, a imagem que contém a PU pode estar associada a duas listas de imagens de referência, referidas como "lista 0" e "lista 1". Em alguns exemplos, uma imagem que contém uma fatia B pode estar associada a uma combinação de lista que é uma combinação de lista 0 e lista 1.

[0086] Além disso, se a PU estiver em uma fatia B, a unidade de estimativa de movimento 122 pode realizar predição unidirecional ou predição bidirecional para uma PU. Quando a unidade de estimativa de movimento 122 realiza predição unidirecional para a PU, a unidade de estimativa de movimento 122 pode procurar as imagens de

referência da lista 0 ou da lista 1 para um bloco de referência para a PU. Desse modo, a unidade de estimativa de movimento 122 pode gerar um índice de referência que indica a imagem de referência na lista 0 ou lista 1 que contém o bloco de referência e um vetor de movimento que indica um deslocamento espacial entre a PU e o bloco de referência. A unidade de estimativa de movimento 122 pode emitir o índice de referência, um indicador de direção de predição e o vetor de movimento como as informações de movimento da PU. O indicador de direção de predição pode indicar se o índice de referência indica uma imagem de referência na lista 0 ou lista 1. A unidade de compensação de movimento 124 pode gerar o bloco de vídeo previsto da PU com base no bloco de referência indicado pelas informações de movimento da PU.

[0087] Quando a unidade de estimativa de movimento 122 realiza a predição bidirecional para uma PU, a unidade de estimativa de movimento 122 pode buscar as imagens de referência na lista 0 para um bloco de referência para a PU e também pode procurar as imagens de referência na lista 1 para outro bloco de referência para a PU. Desse modo, a unidade de estimativa de movimento 122 pode gerar índices de referência que indicam as imagens de referência na lista 0 e lista 1 que contêm os blocos de referência e os vetores de movimento que indicam deslocamentos espaciais entre os blocos de referência e a PU. A unidade de estimativa de movimento 122 pode emitir os índices de referência e os vetores de movimento da PU como as informações de movimento da PU. A unidade de compensação de movimento 124 pode gerar o bloco de vídeo previsto da PU com base nos blocos de referência indicados pelas informações de movimento da PU.

[0088] Em alguns casos, a unidade de

estimativa de movimento 122 não emite um conjunto completo de informações de movimento para uma PU à unidade de codificação por entropia 116. Em vez disso, a unidade de estimativa de movimento 122 pode sinalizar as informações de movimento de uma PU com referência às informações de movimento de outra PU. Por exemplo, a unidade de estimativa de movimento 122 pode determinar que as informações de movimento da PU são suficientemente similares às informações de movimento de uma PU vizinha. Nesse exemplo, a unidade de estimativa de movimento 122 pode indicar, em uma estrutura de sintaxe associada à PU, um valor que indica para o decodificador de vídeo 30 que a PU tem as mesmas informações de movimento que a PU vizinha. Em um outro exemplo, a unidade de estimativa de movimento 122 pode identificar, em uma estrutura de sintaxe associada com a PU, uma PU vizinha e uma diferença de vetor de movimento (MVD). A diferença de vetor de movimento indica uma diferença entre o vetor de movimento da PU e o vetor de movimento da PU vizinha indicada. O decodificador de vídeo 30 pode usar o vetor de movimento da PU vizinha indicada e a diferença de vetor de movimento para determinar o vetor de movimento da PU. Através da referência às informações de movimento de um primeiro PU ao sinalizar as informações de movimento de uma segunda PU, o codificador de vídeo 20 pode ter capacidade para sinalizar as informações de movimento da segunda PU usando menos bits.

[0089] Conforme discutido adicionalmente abaixo com referência à Figura 4, a unidade de processamento de predição 100 pode ser configurada para codificar (por exemplo, codificar ou decodificar) a PU (ou quaisquer outras unidades de vídeo ou blocos de camada de referência e/ou de camada de aperfeiçoamento) realizando-se os métodos ilustrados na Figura 4. Por exemplo, a unidade

de interpredição 121 (por exemplo, através da unidade de estimativa de movimento 122 e/ou unidade de compensação de movimento 124), a unidade de intrapredição 126 ou a unidade de predição intercamada 128 pode ser configurada para realizar os métodos ilustrados na Figura 4, em conjunto ou separadamente.

[0090] Como parte da realização de uma operação de codificação em uma CU, a unidade de intrapredição 126 pode realizar intrapredição em PUs da CU. A intrapredição pode fornecer compactação espacial. Quando a unidade de intrapredição 126 realiza a intrapredição em uma PU, a unidade de intrapredição 126 pode gerar dados preditos para a PU com base em amostras decodificadas de outras PUs na mesma imagem. Os dados preditos para a PU podem incluir um bloco de vídeo previsto e vários elementos de sintaxe. A unidade de intrapredição 126 pode realizar intrapredição em PUs nas fatias I, fatias P e fatias B.

[0091] Para realizar a intrapredição em uma PU, a unidade de intrapredição 126 pode usar múltiplos modos de intrapredição para gerar múltiplos conjuntos de dados preditos para uma PU. Quando a unidade de intrapredição 126 usa um modo de intrapredição para gerar um conjunto de dados preditos para a PU, a unidade de intrapredição 126 pode estender as amostras a partir de blocos de vídeo de PUs vizinhas através de bloco de vídeo da PU em uma direção e/ou gradiente associado ao modo de intrapredição. As PUs vizinhas podem estar acima, acima e à direita, acima e à esquerda ou à esquerda da PU, presumindo uma ordem de codificação do fundo para o topo e da esquerda para direita para PUs, CUs e CTUs. A unidade de intrapredição 126 pode usar vários números de modos de intrapredição, por exemplo, 33 modos de intrapredição direcionais, dependendo do tamanho da PU.

[0092] A unidade de processamento de predição 100 pode selecionar os dados preditos para uma PU dentre os dados preditos gerados pela unidade de compensação de movimento 124 para a PU ou os dados preditos gerados pela unidade de intrapredição 126 para a PU. Em alguns exemplos, a unidade de processamento de predição 100 seleciona os dados preditos para a PU com base nas métricas de taxa/distorção dos conjuntos de dados preditos.

[0093] Se a unidade de processamento de predição 100 selecionar os dados preditos gerados pela unidade de intra predição 126, a unidade de processamento de predição 100 pode sinalizar o modo de intrapredição que foi usado para gerar os dados preditos para as PUs, por exemplo, o modo de intrapredição selecionado. A unidade de processamento de predição 100 pode sinalizar o modo de intrapredição selecionado de várias maneiras. Por exemplo, pode ser provável que o modo de intrapredição selecionado seja o mesmo que o modo de intrapredição de uma PU vizinha. Em outras palavras, o modo de intrapredição da PU vizinha pode ser o modo mais provável para a PU atual. Desse modo, a unidade de processamento de predição 100 pode gerar um elemento de sintaxe para indicar que o modo de intrapredição selecionado é o mesmo que o modo de intra predição da PU vizinha.

[0094] Conforme discutido acima, o codificador de vídeo 20 pode incluir a unidade prevista intercamada 128. A unidade prevista intercamada 128 é configurada para prever um bloco atual (por exemplo, um bloco atual na EL) usando uma ou mais diferentes camadas que estão disponíveis em SVC (por exemplo, uma camada de base ou referência). Tal predição pode ser chamada de predição de intercamada. A unidade prevista intercamada 128 utiliza os métodos de predição para reduzir a redundância de intercamada e

aprimorando, desse modo, a eficiência de codificação e reduzindo os requisitos de recurso computacional. Alguns exemplos de predição de intercamada incluem intrapredição de intercamada, predição de movimento de intercamada e predição residual de intercamada. A intrapredição de intercamada usa a reconstrução de blocos colocalizados na camada de base para prever o bloco atual na camada de aperfeiçoamento. A predição de movimento de intercamada usa informações de movimento da camada de base para prever movimento na camada de aperfeiçoamento. A predição residual de intercamada usa o resíduo da camada de base para prever o resíduo da camada de aperfeiçoamento. Cada um dos esquemas de predição de intercamada é discutido abaixo em maiores detalhes.

[0095] Após a unidade de processamento de predição 100 selecionar os dados preditos para PUs de uma CU, a unidade de geração residual 102 pode gerar dados residuais para a CU através da subtração (por exemplo, indicado pelo sinal de menos) dos blocos de vídeo preditos das PUs da CU do bloco de vídeo da CU. Os dados residuais de uma CU podem incluir blocos de vídeo residuais em 2D que correspondem a componentes de amostra diferentes das amostras no bloco de vídeo da CU. Por exemplo, os dados residuais podem incluir um bloco de vídeo residual que corresponde a diferenças entre componentes de luminância de amostras nos blocos de vídeo preditos das PUs da CU e componentes de luminância de amostras no bloco de vídeo original da CU. Além disso, os dados residuais da CU podem incluir blocos de vídeo residuais que correspondem às diferenças entre componentes de crominância de amostras nos blocos de vídeo preditos das PUs da CU e as componentes de crominância das amostras no bloco de vídeo original da CU.

[0096] A unidade de processamento de predição

100 pode realizar o particionamento de árvore quaternária para particionar os blocos de vídeo residuais de uma CU em sub-blocos. Cada bloco de vídeo residual não dividido pode estar associado a uma TU diferente da CU. Os tamanhos e posições dos blocos de vídeo residuais associados às TUs de uma CU podem ou não podem ser com base nos tamanhos e posições de blocos de vídeo associados com as PUs da CU. Uma estrutura em árvore quaternária referida como uma "árvore quaternária residual" (QT) pode incluir nós associados a cada um dos blocos de vídeo residuais. As TUs de uma CU podem corresponder a nós folha da RQT.

[0097] A unidade de processamento de transformada 104 pode gerar um ou mais blocos de coeficiente de transformada para cada TU de uma CU através da aplicação de uma ou mais transformadas a um bloco de vídeo residual associado à TU. Cada um dos blocos de coeficiente de transformada pode ser uma matriz em 2D de coeficientes de transformada. A unidade de processamento de transformada 104 pode aplicar várias transformadas ao bloco de vídeo residual associado a uma TU. Por exemplo, a unidade de processamento de transformada 104 pode aplicar uma transformada discreta de cosseno (DCT), uma transformada direcional ou uma transformada conceitualmente similar ao bloco de vídeo residual associado a uma TU.

[0098] Após a unidade de processamento de transformada 104 gerar um bloco de coeficiente de transformada associado a uma TU, a unidade de quantização 106 pode quantizar os coeficientes de transformada no bloco de coeficiente de transformada. A unidade de quantização 106 pode quantizar um bloco de coeficiente de transformada associado a uma TU de uma CU com base em um valor de QP associado à CU.

[0099] O codificador de vídeo 20 pode associar

um valor de QP a uma CU de várias maneiras. Por exemplo, o codificador de vídeo 20 pode realizar uma análise de taxa de distorção em um bloco em árvore associado à CU. Na análise da taxa de distorção, o codificador de vídeo 20 pode gerar múltiplas representações codificadas do bloco em árvore através da realização de uma operação de codificação de múltiplas vezes no bloco em árvore. O codificador de vídeo 20 pode associar valores de QP diferentes à CU quando o codificador de vídeo 20 gera representações codificadas diferentes do bloco em árvore. O codificador de vídeo 20 pode sinalizar que um determinado valor de QP está associado à CU quando o determinado valor de QP está associado à CU em uma representação codificada do bloco em árvore que tem um débito mais inferior e métrica de distorção.

[00100] A unidade de quantização inversa 108 e a unidade de transformada inversa 110 podem aplicar quantização inversa e transformadas inversas ao bloco de coeficiente de transformada, respectivamente, para reconstruir um bloco de vídeo residual a partir do bloco de coeficiente de transformada. A unidade de reconstrução 112 pode adicionar o bloco de vídeo residual reconstruído às amostras correspondentes a partir de um ou mais blocos de vídeo preditos de amostra gerados pela unidade de processamento de predição 100 para produzir um bloco de transformada reconstruído associado a uma TU. Através da reconstrução dos blocos de vídeo para cada TU de uma CU, dessa maneira, o codificador de vídeo 20 pode reconstruir o bloco de vídeo da CU.

[00101] Após a unidade de reconstrução 112 reconstruir o bloco de vídeo de uma CU, a unidade de filtro 113 pode realizar uma operação de desbloqueio para reduzir os artefatos de bloqueio no bloco de vídeo associado à CU.

Após realizar as uma ou mais operações de desbloqueio, a unidade de filtro 113 pode armazenar o bloco de vídeo reconstruído da CU em buffer de imagem decodificada 114. A unidade de estimativa de movimento 122 e a unidade de compensação de movimento 124 podem usar uma imagem de referência que contém o bloco de vídeo reconstruído para realizar a interpredição nas PUs de imagens subsequentes. Além disso, a unidade de intrapredição 126 pode usar blocos de vídeo reconstruídos no buffer de imagem decodificada 114 para realizar intrapredição em outras PUs na mesma imagem que a CU.

[00102] A unidade de codificação por entropia 116 pode receber dados a partir de outros componentes funcionais de codificador de vídeo 20. Por exemplo, a unidade de codificação por entropia 116 pode receber blocos de coeficiente de transformada a partir da unidade de quantização 106 e pode receber elementos de sintaxe a partir da unidade de processamento de predição 100. Quando a unidade de codificação por entropia 116 recebe os dados, a unidade de codificação por entropia 116 pode realizar uma ou mais operações de codificação por entropia para gerar dados codificados por entropia. Por exemplo, o codificador de vídeo 20 pode realizar uma operação de codificação de comprimento variável adaptativo ao contexto (CAVLC), uma operação de CABAC, uma operação de codificação de comprimento de variável para variável (V2V), uma operação de codificação aritmética binária adaptativa a contexto com base em sintaxe (SBAC), uma operação de codificação de Particionamento por Entropia com Intervalo de Probabilidade (PIPE) ou um outro tipo de operação de codificação por entropia nos dados. A unidade de codificação por entropia 116 pode emitir um fluxo de bits que inclui os dados codificados por entropia.

[00103] Como parte da realização de uma operação de codificação por entropia de dados, a unidade de codificação por entropia 116 pode selecionar um modelo de contexto. Se a unidade de codificação por entropia 116 estiver realizando uma operação de CABAC, o modelo de contexto pode indicar estimativas de probabilidades de números binários que têm valores particulares. No contexto de CABAC, o termo “número binário” é usado para se referir a um bit de uma versão binarizada de um elemento de sintaxe.

#### CODIFICADOR DE VÍDEO DE MÚLTIPLA CAMADA

[00104] A Figura 2B é um diagrama de blocos que ilustra um exemplo de um codificador de vídeo de múltipla camada 23 que pode implantar técnicas de acordo com os aspectos descritos nesta descrição. O codificador de vídeo 23 pode ser configurado para processar quadros de vídeo de múltipla camada, tais como para codificação de múltiplas visualizações e SHVC. Além disso, o codificador de vídeo 23 pode ser configurado para realizar qualquer uma ou todas as técnicas desta descrição.

[00105] O codificador de vídeo 23 inclui um codificador de vídeo 20A e codificador de vídeo 20B, cada um dos quais pode ser configurado como o codificador de vídeo 20 e pode realizar as funções descritas acima em relação ao codificador de vídeo 20. Além disso, conforme indicado pela reutilização de números de referência, os codificadores de vídeo 20A e 20B podem incluir pelo menos alguns dos sistemas e subsistemas como o codificador de vídeo 20. Embora o codificador de vídeo 23 seja ilustrado conforme incluindo dois codificadores de vídeo 20A e 20B, o codificador de vídeo 23 não é limitado como tal e pode incluir qualquer número de camadas de codificador de vídeo 20. Em algumas modalidades, o codificador de vídeo 23 pode

incluir um codificador de vídeo 20 para cada imagem ou quadro em uma unidade de acesso. Por exemplo, uma unidade de acesso que inclui cinco imagens pode ser processada ou codificada por um codificador de vídeo que inclui cinco camadas de codificador. Em algumas modalidades, o codificador de vídeo 23 pode incluir mais camadas de codificador que quadros em uma unidade de acesso. Em alguns tais casos, algumas das camadas de codificador de vídeo podem estar inativas ao processar algumas unidades de acesso.

[00106] Além dos codificadores de vídeo 20A e 20B, o codificador de vídeo 23 pode incluir uma unidade de reamostragem 90. A unidade de reamostragem 90 pode, em alguns casos, superamostrar uma camada de base de um quadro de vídeo recebido para, por exemplo, criar uma camada de aperfeiçoamento. A unidade de reamostragem 90 pode superamostrar informações particulares associadas à camada de base recebida de um quadro, mas não outras informações. Por exemplo, a unidade de reamostragem 90 pode superamostrar o tamanho espacial ou número de pixels da camada de base, mas o número de fatias ou a contagem de ordem de imagem pode permanecer constante. Em alguns casos, a unidade de reamostragem 90 não pode processar o vídeo recebido e/ou pode ser opcional. Por exemplo, em alguns casos, a unidade de processamento de predição 100 pode realizar a superamostragem. Em algumas modalidades, a unidade de reamostragem 90 é configurada para superamostrar uma camada e reorganizar, redefinir, modificar ou ajustar uma ou mais fatias para cumprir com um conjunto de regras limites de fatia e/ou regras de varredura. Embora primeiramente descrito como superamostragem de uma camada de base ou de uma camada inferior em uma unidade de acesso, em alguns casos, a unidade de reamostragem 90 pode

subamostrar uma camada. Por exemplo, se durante a transmissão contínua de um vídeo a largura de banda for reduzida, um quadro pode ser subamostrado em vez de superamostrado.

[00107] A unidade de reamostragem 90 pode ser configurada para receber uma imagem ou um quadro (ou informações de imagem associadas à imagem) a partir do buffer de imagem decodificado 114 do codificador de camada inferior (por exemplo, do codificador de vídeo 20A) e para superamostrar a imagem (ou as informações de imagem recebidas). Essa imagem superamostrada pode, então, ser fornecida para a unidade de processamento de predição 100 de um codificador de camada superior (por exemplo, o codificador de vídeo 20B) configurado para codificar uma imagem na mesma unidade de acesso como o codificador de camada inferior. Em alguns casos, o codificador de camada superior é uma camada removida do codificador de camada inferior. Em outros casos, pode haver um ou mais c de camada superior entre o codificador de vídeo da camada 0 e o codificador da camada 1 da Figura 2B.

[00108] Em alguns casos, a unidade de reamostragem 90 pode ser omitida ou desviada. Em tais casos, a imagem do buffer de imagem decodificado 114 do codificador de vídeo 20A pode ser fornecida diretamente ou pelo menos sem ser fornecida para a unidade de reamostragem 90, para a unidade de processamento de predição 100 do codificador de vídeo 20B. Por exemplo, se os dados de vídeo fornecidos para o codificador de vídeo 20B e para a imagem de referência a partir do buffer de imagem decodificado 114 do codificador de vídeo 20A forem do mesmo tamanho ou resolução, a imagem de referência pode ser fornecida para o codificador de vídeo 20B sem qualquer reamostragem.

[00109] Em algumas modalidades, o codificador

de vídeo 23 subamostra os dados de vídeo a serem fornecidos ao codificador de camada inferior com uso da unidade de subamostragem 94 antes de fornecer os dados de vídeo para o codificador de vídeo 20A. Alternativamente, a unidade de subamostragem 94 pode ser uma unidade de reamostragem 90 com capacidade para superamostrar ou subamostrar os dados de vídeo. Em ainda outras modalidades, a unidade de subamostragem 94 pode ser omitida.

[00110] Conforme ilustrado na Figura 2B, o codificador de vídeo 23 pode incluir adicionalmente um multiplexador 98 ou mux. O mux 98 pode emitir um fluxo de bits combinado a partir do codificador de vídeo 23. O fluxo de bits combinado pode ser criado através da retirada de um fluxo de bits de cada um dos codificadores de vídeo 20A e 20B e através da alteração de qual fluxo de bits é emitido em um determinado tempo. Enquanto, em alguns casos, os bits dos dois (ou mais no caso de mais que duas camadas de codificador de vídeo) fluxos de bits podem ser alterado um bit por vez, em muitos casos, os fluxos de bits são combinados de modo diferente. Por exemplo, o fluxo de bits de saída pode ser criado através da alteração do fluxo de bits selecionado um bloco por vez. Em um outro exemplo, o fluxo de bits de saída pode ser criado através da emissão de uma razão diferente de 1:1 de blocos de cada um dos codificadores de vídeo 20A e 20B. Por exemplo, dois blocos podem ser emitidos a partir do codificador de vídeo 20B para cada saída de bloco do codificador de vídeo 20A. Em algumas modalidades, o fluxo de saída do mux 98 pode ser pré-programado. Em outras modalidades, o mux 98 pode combinar os fluxos de bits a partir dos codificadores de vídeo 20A, 20B com base em um sinal de controle recebido a partir de um sistema externo ao codificador de vídeo 23, tal como, a partir de um processador em um dispositivo de

fonte que inclui o módulo de fonte 12. O sinal de controle pode ser gerado com base na resolução ou taxa de bits de um vídeo a partir da fonte de vídeo 18, com base em uma largura de banda do enlace 16, com base em uma subscrição associada a um usuário (por exemplo, uma subscrição paga versus uma subscrição livre) ou com base em qualquer outro fator para determinar uma saída de resolução desejada a partir do codificador de vídeo 23.

#### DECODIFICADOR DE VÍDEO

[00111] A Figura 3A é um diagrama de blocos que ilustra um exemplo de um decodificador de vídeo que pode implantar técnicas de acordo com os aspectos descritos nesta descrição. O decodificador de vídeo 30 pode ser configurado para processar uma única camada de um quadro de vídeo, tal como para HEVC. Além disso, o decodificador de vídeo 30 pode ser configurado para realizar qualquer uma ou todas as técnicas desta descrição. Como um exemplo, a unidade de compensação de modo 162 e/ou a unidade de intrapredição 164 podem ser configurados para realizar qualquer e todas as técnicas descritas nessa descrição. Em uma modalidade, o decodificador de vídeo 30 pode incluir opcionalmente a unidade de predição de intercamada 166 que é configurada para realizar qualquer uma ou todas as técnicas descritas nesta descrição. Em outras modalidades, a predição de intercamada pode ser realizada pela unidade de processamento de predição 152 (por exemplo, unidade de compensação de movimento 162 e/ou unidade de intrapredição 164), no caso em que a unidade de predição de intercamada 166 pode ser omitida. Entretanto, aspectos dessa descrição não são tão limitados. Em alguns exemplos, as técnicas descritas nessa descrição podem ser compartilhadas entre os vários componentes de decodificador de vídeo 30. Em alguns exemplos, adicional ou alternativamente, um processador

(não mostrado) pode ser configurado para realizar qualquer uma ou todas as técnicas descritas nesta descrição.

[00112] Para propósitos explicativos, essa descrição descreve o decodificador de vídeo 30 no contexto de codificação de HEVC. No entanto, as técnicas dessa descrição podem ser aplicáveis a outros padrões ou métodos de codificação. O exemplo representado na Figura 3A é para um codec de camada única. No entanto, conforme será descrito adicionalmente em relação à Figura 3B, algum ou todo o decodificador de vídeo 30 pode ser duplicado para processar um codec de camada múltipla.

[00113] No exemplo da Figura 3A, o decodificador de vídeo 30 inclui uma pluralidade de componentes funcionais. Os componentes funcionais de decodificador de vídeo 30 incluem uma unidade de decodificação por entropia 150, uma unidade de processamento de predição 152, uma unidade de quantização inversa 154, uma unidade de transformada inversa 156, uma unidade de reconstrução 158, uma unidade de filtro 159 e um buffer de imagem decodificado 160. A unidade de processamento de predição 152 inclui uma unidade de compensação de movimento 162, uma unidade de intrapredição 164 e uma unidade de predição de intercamada 166. Em alguns exemplos, o decodificador de vídeo 30 pode realizar uma passagem de decodificação geralmente recíproca à passagem de codificação descrita em relação ao codificador de vídeo 20 da Figura 2A. Em outros exemplos, o decodificador de vídeo 30 pode incluir mais, menos ou diferentes componentes funcionais.

[00114] O decodificador de vídeo 30 pode receber um fluxo de bits que compreende os dados de vídeo codificados. O fluxo de bits pode incluir uma pluralidade de elementos de sintaxe. Quando o decodificador de vídeo 30

recebe o fluxo de bits, a unidade de decodificação por entropia 150 pode realizar uma operação de análise no fluxo de bits. Como um resultado de realização da operação de análise no fluxo de bits, a unidade de decodificação por entropia 150 pode extrair elementos de sintaxe do fluxo de bits. Como parte da Realização da operação de análise, a unidade de decodificação por entropia 150 pode decodificar por entropia elementos de sintaxe decodificados por entropia no fluxo de bits. A unidade de processamento de predição 152, a unidade de quantização inversa 154, a unidade de transformada inversa 156, a unidade de reconstrução 158 e a unidade de filtro 159 podem realizar uma operação de reconstrução que gera dados de vídeo decodificados com base nos elementos de sintaxe extraídos do fluxo de bits.

[00115] Conforme discutido acima, o fluxo de bits pode compreender uma série de unidades de NAL. As unidades de NAL do fluxo de bits podem incluir unidades de NAL de conjunto de parâmetros de vídeo, unidades de NAL de conjunto de parâmetros de sequência, unidades de NAL de conjunto de parâmetros de imagem, unidades de NAL de SEI e assim por diante. Como parte da realização da operação de análise no fluxo de bits, a unidade de decodificação por entropia 150 pode realizar operações de análise que extraem e decodificam por entropia os conjuntos de parâmetros de sequência a partir das unidades de NAL de conjunto de parâmetros de sequência, conjuntos de parâmetros de imagem a partir das unidades de NAL de conjunto de parâmetros de imagem, dados de SEI a partir das unidades de NAL de SEI e assim por diante.

[00116] Além disso, as unidades de NAL do fluxo de bits podem incluir unidades de NAL de fatia codificada. Como parte da realização da operação de análise no fluxo de

bits, a unidade de decodificação por entropia 150 pode realizar operações de análise que extraem e decodificam por entropia as fatias codificadas a partir das unidades de NAL de fatia codificada. Cada uma das fatias codificadas pode incluir um cabeçalho de fatia e dados de fatia. O cabeçalho de fatia pode conter elementos de sintaxe que pertencem a uma fatia. Os elementos de sintaxe no cabeçalho de fatia podem incluir um elemento de sintaxe que identifica um conjunto de parâmetros de imagem associado a uma gravura que contém a fatia. A unidade de decodificação por entropia 150 pode realizar operações de decodificação por entropia, tais como operações de decodificação de CABAC, em elementos de sintaxe no cabeçalho de fatia codificada para recuperar o cabeçalho de fatia.

[00117] Como parte da extração dos dados de fatia das unidades de NAL de fatia codificada, a unidade de decodificação por entropia 150 pode realizar operações de análise que extraem elementos de sintaxe das CUs codificadas nos dados de fatia. Os elementos de sintaxe extraídos podem incluir elementos de sintaxe associados aos blocos de coeficiente de transformada. A unidade de decodificação por entropia 150 pode, então, realizar as operações de decodificação de CABAC em alguns dos elementos de sintaxe.

[00118] Após a unidade de decodificação por entropia 150 realizar uma operação de análise em uma CU não particionada, o decodificador de vídeo 30 pode realizar uma operação de reconstrução na CU não particionada. Para realizar a operação de reconstrução em uma CU não particionada, o decodificador de vídeo 30 pode realizar uma operação de reconstrução em cada TU da CU. Através da realização da operação de reconstrução para cada TU da CU, o decodificador de vídeo 30 pode reconstruir um bloco de

vídeo residual associado à CU.

[00119] Como parte da realização de uma operação de reconstrução em uma TU, a unidade de quantização inversa 154 pode quantizar de modo inverso, por exemplo, desquantizar, um bloco de coeficiente de transformada associado à TU. A unidade de quantização inversa 154 pode quantizar de modo inverso o bloco de coeficiente de transformada de uma maneira similar aos processos de quantização inversa propostos para HEVC ou definidos pelo padrão de decodificação H.264. A unidade de quantização inversa 154 pode usar um parâmetro de quantização QP calculado pelo codificador de vídeo 20 para uma CU do bloco de coeficiente de transformada para determinar um grau de quantização e, da mesma forma, um grau de quantização inversa para aplicação pela unidade de quantização inversa 154.

[00120] Após a unidade de quantização inversa 154 quantizar de modo inverso um bloco de coeficiente de transformada, a unidade de transformada inversa 156 pode gerar um bloco de vídeo residual para a TU associada ao bloco de coeficiente de transformada. A unidade de transformada inversa 156 pode aplicar uma transformada inversa ao bloco de coeficiente de transformada a fim de gerar o bloco de vídeo residual para a TU. Por exemplo, a unidade de transformada inversa 156 pode aplicar uma DCT inversa, uma transformada de número inteiro inversa, uma transformada de Karhunen-Loeve inversa (KLT), uma transformada rotacional inversa, uma transformada direcional inversa ou uma outra transformada inversa ao bloco de coeficiente de transformada. Em alguns exemplos, a unidade de transformada inversa 156 pode determinar a aplicação de uma transformada inversa ao bloco de coeficiente de transformada com base na sinalização do

codificador de vídeo 20. Em tais exemplos, a unidade de transformada inversa 156 pode determinar a transformada inversa com base em uma transformada sinalizada no nó raiz de uma árvore quaternária para um bloco em árvore associado ao bloco de coeficiente de transformada. Em outros exemplos, a unidade de transformada inversa 156 pode inferir a transformada inversa a partir de uma ou mais características de codificação, tal como tamanho de bloco, modo de codificação ou similares. Em alguns exemplos, a unidade de transformada inversa 156 pode aplicar uma transformada inversa colocada em cascata.

[00121] Em alguns exemplos, a unidade de compensação de movimento 162 pode aprimorar o bloco de vídeo previsto de a PU através da realização da interpolação com base em filtros de interpolação. Os identificadores para filtros de interpolação a serem usados para compensação de movimento com precisão de subamostra podem estar incluídos nos elementos de sintaxe. A unidade de compensação de movimento 162 pode usar os mesmos filtros de interpolação usados pelo codificador de vídeo 20 durante a geração do bloco de vídeo previsto da PU para calcular os valores interpolados para amostras abaixo do número inteiro de um bloco de referência. A unidade de compensação de movimento 162 pode determinar os filtros de interpolação usados pelo codificador de vídeo 20 de acordo com as informações de sintaxe recebidas e usar os filtros de interpolação para produzir o bloco de vídeo previsto.

[00122] Conforme discutido adicionalmente abaixo com referência à Figura 4, a unidade de processamento de predição 152 pode codificar (por exemplo, codificar ou decodificar) a PU (ou quaisquer outras unidades de vídeo ou blocos de camada de referência e/ou de camada de aperfeiçoamento) através da realização dos

métodos ilustrados na Figura 4. Por exemplo, a unidade de compensação de movimento 162, a unidade de intrapredição 164 ou a unidade de predição intercamada 166 pode ser configurada para realizar os métodos ilustrados nas Figuras 4, em conjunto ou separadamente.

[00123] Se uma PU for codificada usando a intrapredição, a unidade de intrapredição 164 pode realizar a intrapredição para gerar um bloco de vídeo previsto para a PU. Por exemplo, a unidade de intrapredição 164 pode determinar um modo de intrapredição para a PU com base em elementos de sintaxe no fluxo de bits. O fluxo de bits pode incluir elementos de sintaxe que a unidade de intrapredição 164 pode usar para determinar o modo de intrapredição da PU.

[00124] Em algumas instâncias, os elementos de sintaxe podem indicar que a unidade de intrapredição 164 deve usar o modo de intrapredição de uma outra PU para determinar o modo de intrapredição da PU atual. Por exemplo, pode ser provável que o modo de intrapredição da PU atual seja o mesmo do modo de intrapredição de uma PU vizinha. Em outras palavras, o modo de intrapredição da PU vizinha pode ser o modo mais provável para a PU atual. Por conseguinte, nesse exemplo, o fluxo de bits pode incluir um elemento de sintaxe pequeno que indica que o modo de intrapredição da PU é o mesmo do modo de intrapredição da PU vizinha. A unidade de intrapredição 164 pode, então, usar o modo de intrapredição para gerar dados de predição (por exemplo, amostras previstas) para a PU com base nos blocos de vídeo de PU espacialmente vizinhas.

[00125] Conforme discutido acima, o decodificador de vídeo 30 também pode incluir a unidade de predição de intercamada 166. A unidade prevista intercamada 166 é configurada para prever um bloco atual (por exemplo,

um bloco atual na EL) usando uma ou mais diferentes camadas que estão disponíveis em SVC (por exemplo, uma camada de base ou referência). Tal predição pode ser chamada de predição de intercamada. A unidade prevista intercamada 166 utiliza os métodos de predição para reduzir a redundância de intercamada e aprimorando, desse modo, a eficiência de codificação e reduzindo os requisitos de recurso computacional. Alguns exemplos de predição de intercamada incluem intrapredição de intercamada, predição de movimento de intercamada e predição residual de intercamada. A intrapredição de intercamada usa a reconstrução de blocos colocalizados na camada de base para prever o bloco atual na camada de aperfeiçoamento. A predição de movimento de intercamada usa informações de movimento da camada de base para prever movimento na camada de aperfeiçoamento. A predição residual de intercamada usa o resíduo da camada de base para prever o resíduo da camada de aperfeiçoamento. Cada um dos esquemas de predição de intercamada é discutido abaixo em maiores detalhes.

[00126] A unidade de reconstrução 158 pode usar os blocos de vídeo residuais associados às TUs de uma CU e os blocos de vídeo preditos da PUs da CU, por exemplo, dados de intrapredição ou dados de interpredição, conforme aplicável, para reconstruir o bloco de vídeo da CU. Desse modo, o decodificador de vídeo 30 pode gerar um bloco de vídeo previsto e um bloco de vídeo residual com base em elementos de sintaxe no fluxo de bits e pode gerar um bloco de vídeo com base no bloco de vídeo previsto e no bloco de vídeo residual.

[00127] Após a unidade de reconstrução 158 reconstruir o bloco de vídeo da CU, a unidade de filtro 159 pode realizar uma operação de desbloqueio para reduzir artefatos de bloqueio associados à CU. Após a unidade de

filtro 159 realizar uma operação de desbloqueio para reduzir os artefatos de bloqueio associados à CU, o decodificador de vídeo 30 pode armazenar o bloco de vídeo da CU em buffer de imagem decodificado 160. O buffer de gravura decodificada 160 pode fornecer gravuras de referência para compensação, intrapredição e apresentação de movimento subsequente em um dispositivo de visor, tal como dispositivo de visor 32 da Figura 1A ou 1B. Por exemplo, o decodificador de vídeo 30 pode realizar, com base nos blocos de vídeo no buffer de gravura decodificada 160, operações de intrapredição ou interpredição em PUs de outras CUs.

#### DECODIFICADOR DE CAMADA MÚLTIPLA

[00128] A Figura 3B é um diagrama de blocos que ilustra um exemplo de um decodificador de vídeo de camada múltipla 33 que pode implantar técnicas de acordo com os aspectos descritos nesta descrição. O decodificador de vídeo 33 pode ser configurado para processar quadros de vídeo de múltipla camada, tais como para codificação de múltiplas visualizações e SHVC. Além disso, o decodificador de vídeo 33 pode ser configurado para realizar qualquer uma ou todas as técnicas desta descrição.

[00129] O decodificador de vídeo 33 inclui um decodificador de vídeo 30A e decodificador de vídeo 30B, cada um dos quais pode ser configurado como o decodificador de vídeo 30 e pode realizar as funções descritas acima em relação ao decodificador de vídeo 30. Além disso, conforme indicado pela reutilização de números de referência, os decodificadores de vídeo 30A e 30B podem incluir pelo menos alguns dos sistemas e subsistemas como o decodificador de vídeo 30. Embora o decodificador de vídeo 33 seja ilustrado conforme incluindo dois decodificadores de vídeo 30A e 30B, o decodificador de vídeo 33 não é limitado como tal e pode

incluir qualquer número de camadas de decodificador de vídeo 30. Em algumas modalidades, o decodificador de vídeo 33 pode incluir um decodificador de vídeo 30 para cada imagem ou quadro em uma unidade de acesso. Por exemplo, uma unidade de acesso que inclui cinco imagens pode ser processada ou decodificada por um decodificador de vídeo que inclui cinco camadas de decodificador. Em algumas modalidades, o decodificador de vídeo 33 pode incluir mais camadas de decodificador que quadros em uma unidade de acesso. Em alguns tais casos, algumas das camadas de decodificador de vídeo podem ser inativas ao processar algumas unidades de acesso.

[00130] Além dos decodificadores de vídeo 30A e 30B, o decodificador de vídeo 33 pode incluir uma unidade de superamostragem 92. Em algumas modalidades, a unidade de superamostragem 92 pode superamostrar uma camada de base de um quadro de vídeo recebido para criar uma camada melhorada para ser adicionada à lista de imagem de referência para o quadro ou unidade de acesso. Essa camada melhorada pode ser armazenada no buffer de imagem decodificado 160. Em algumas modalidades, a unidade de superamostragem 92 pode incluir algumas das ou todas as modalidades descritas em relação à unidade de reamostragem 90 da Figura 2A. Em algumas modalidades, a unidade de superamostragem 92 é configurada para superamostrar uma camada e reorganizar, redefinir, modificar ou ajustar uma ou mais fatias para cumprir com um conjunto de regras limites de fatia e/ou regras de varredura. Em alguns casos, a unidade de superamostragem 92 pode ser uma unidade de reamostragem configurada para superamostrar e/ou subamostrar uma camada de um quadro de vídeo recebido.

[00131] A unidade de superamostragem 92 pode ser configurada para receber uma imagem ou um quadro (ou

informações de imagem associadas à imagem) a partir do buffer de imagem decodificado 160 do decodificador de camada inferior (por exemplo, do decodificador de vídeo 30A) e para superamostrar a imagem (ou as informações de imagem recebidas). Essa imagem superamostrada pode, então, ser fornecida para a unidade de processamento de predição 152 de um decodificador de camada superior (por exemplo, o decodificador de vídeo 30B) configurado para decodificar uma imagem na mesma unidade de acesso como do decodificador de camada inferior. Em alguns casos, o decodificador de camada superior é uma camada removida do decodificador de camada inferior. Em outros casos, pode haver um ou mais decodificadores de camada superior entre o decodificador da camada 0 e o decodificador da camada 1 da Figura 3B.

[00132] Em alguns casos, a unidade de superamostragem 92 pode ser omitida ou desviada. Em tais casos, a imagem do buffer de imagem decodificado 160 do decodificador de vídeo 30A pode ser fornecida diretamente ou pelo menos sem ser fornecida para a unidade de superamostragem 92, para a unidade de processamento de predição 152 do decodificador de vídeo 30B. Por exemplo, se os dados de vídeo fornecidos para o decodificador de vídeo 30B e para a imagem de referência a partir do buffer de imagem decodificado 160 do decodificador de vídeo 30A forem do mesmo tamanho ou resolução, a imagem de referência pode ser fornecida para o decodificador de vídeo 30B sem superamostragem. Além disso, em algumas modalidades, a unidade de superamostragem 92 pode ser uma unidade de reamostragem 90 configurada para superamostrar ou subamostrar uma imagem de referência recebida a partir do buffer de imagem decodificado 160 do decodificador de vídeo 30A.

[00133] Conforme ilustrado na Figura 3B, o

decodificador de vídeo 33 pode incluir adicionalmente um desmultiplexador 99 ou demux. O demux 99 pode dividir um fluxo de bits de vídeo codificado em múltiplos fluxos de bits com cada saída de fluxo de bits pelo demux 99 que é fornecido para um decodificador de vídeo diferente 30A e 30B. Os múltiplos fluxos de bits podem ser criados através do recebimento de um fluxo de bits e cada um dos decodificadores de vídeo 30A e 30B recebe uma porção do fluxo de bits em um determinado tempo. Enquanto, em alguns casos, os bits do fluxo de bits recebido no demux 99 podem ser alternados um bit por vez entre cada um dos decodificadores de vídeo (por exemplo, decodificadores de vídeo 30A e 30B no exemplo da Figura 3B), em muitos casos, o fluxo de bits é dividido de modo diferente. Por exemplo, o fluxo de bits pode ser dividido através da alteração na qual decodificador de vídeo recebe o fluxo de bits um bloco por vez. Em um outro exemplo, o fluxo de bits pode ser dividido por uma razão diferente de 1: 1 de blocos para cada um dos decodificadores de vídeo 30A e 30B. Por exemplo, dois blocos podem ser fornecidos para o decodificador de vídeo 30B para cada bloco fornecido para o decodificador de vídeo 30A. Em algumas modalidades, a divisão do fluxo de bits pelo demux 99 pode ser pré-programada. Em outras modalidades, o demux 99 pode dividir o fluxo de bits com base em um sinal de controle recebido a partir de um sistema externo ao decodificador de vídeo 33, tal como a partir de um processador em um dispositivo de destino que inclui o módulo de destino 14. O sinal de controle pode ser gerado com base na resolução ou taxa de bits de um vídeo a partir da interface de entrada 28, com base em uma largura de banda do enlace 16, com base em uma subscrição associada a um usuário (por exemplo, uma subscrição paga versus uma subscrição livre) ou com base em

qualquer outro fator para determinar uma resolução obtinível pelo decodificador de vídeo 33.

#### IMAGENS DE PONTO DE ACESSO INTRA-ALEATÓRIO (IRAP)

[00134] Alguns esquemas de codificação de vídeo podem fornecer vários pontos de acesso aleatório por todo o fluxo de bits de modo que o fluxo de bits possa ser decodificado iniciando a partir de qualquer um desses pontos de acesso aleatório sem precisar decodificar quaisquer imagens que precedem esses pontos de acesso aleatório no fluxo de bits. Em tais esquemas de codificação de vídeo, todas as imagens que seguem um ponto de acesso aleatório na ordem de saída (por exemplo, que incluem essas imagens que estão na mesma unidade de acesso que a imagem que fornece o ponto de acesso aleatório) podem ser corretamente decodificadas sem usar quaisquer imagens que precedem o ponto de acesso aleatório. Por exemplo, mesmo se uma porção do fluxo de bits for perdida durante a transmissão ou durante a decodificação, um decodificador pode reiniciar a decodificação do fluxo de bits começando a partir do próximo ponto de acesso aleatório. O suporte para acesso aleatório pode facilitar, por exemplo, serviços de transmissão contínua dinâmica, operações de busca, comutação de canal, etc.

[00135] Em alguns esquemas de codificação, tais pontos de acesso aleatório podem ser fornecidos pelas imagens que são referidas como imagens de ponto de acesso intra-aleatório (IRAP). Por exemplo, um ponto de acesso aleatório (por exemplo, fornecido por uma imagem de camada de aperfeiçoamento de IRAP) em uma camada de aperfeiçoamento ("layerA") contida em uma unidade de acesso ("auA") pode fornecer acesso aleatório específico por camada de modo que, para cada camada de referência ("layerB") de layerA (por exemplo, uma camada de referência

que é uma camada que é usada para prever layerA) que tem um ponto de acesso aleatório contido em uma unidade de acesso ("auB") que está em layerB e precede auA na ordem de decodificação (ou um ponto de acesso aleatório contido em auA), as imagens em layerA que seguem auB na ordem de saída (que incluem essas imagens localizadas em auB), sejam corretamente decodificáveis sem a necessidade de decodificar quaisquer imagens em layerA que precede auB.

[00136] As imagens de IRAP podem ser codificadas com uso de intrapredição (por exemplo, codificadas sem se referir a outras imagens) e/ou predição intercamada e podem incluir, por exemplo, imagens de atualização de decodificador instantânea (IDR), imagens de acesso aleatório limpo (CRA) e imagens de acesso de enlace desfeito (BLA). Quando há uma imagem de IDR no fluxo de bits, todas as imagens que precedem a imagem de IDR na ordem de decodificação não são usadas para predição por imagens que seguem a imagem de IDR. Quando há uma imagem de CRA no fluxo de bits, as imagens que seguem a imagem de CRA podem ou não podem usar imagens que precedem a imagem de CRA na ordem de decodificação para predição. Essas imagens que seguem a imagem de CRA na ordem de decodificação, mas usam imagens que precedem a imagem de CRA na ordem de decodificação podem ser referidas como imagens precedentes omitidas de acesso aleatório (RASL). Um outro tipo de imagem que pode seguir uma imagem de IRAP na ordem de decodificação e preceder a mesma na ordem de saída é uma imagem precedente decodificável de acesso aleatório (RADL), que pode não conter referências a quaisquer imagens que precedem a imagem de IRAP na ordem de decodificação. As imagens de RASL podem ser descartadas pelo decodificador se as imagens que precedem a imagem de CRA não estiverem disponíveis. Uma imagem de BLA indica ao decodificador que

as imagens que precedem a imagem de BLA não podem estar disponíveis para o decodificador (por exemplo, devido ao fato de que os dois fluxos de bits são unidos e a imagem de BLA é a primeira imagem do segundo fluxo de bits na ordem de decodificação). Uma unidade de acesso (por exemplo, um grupo de imagens que consiste em todas as imagens codificadas associadas ao mesmo tempo de saída ao longo das múltiplas camadas) que contém uma imagem de camada de base (por exemplo, que tem um valor de ID de camada de "0") que é uma imagem de IRAP pode ser denominada como uma unidade de acesso de IRAP.

#### UNIDADES DE CAMADA DE ABSTRAÇÃO DE REDE (NAL) E CONJUNTOS DE PARÂMETROS

[00137] Conforme discutido acima, os parâmetros usados por um codificador ou um decodificador podem ser agrupados em conjuntos de parâmetros com base no nível de codificação em que os mesmos podem ser utilizados. Por exemplo, os parâmetros que são utilizados por uma ou mais sequências de vídeo codificado no fluxo de bits podem estar incluídos em um conjunto de parâmetros de vídeo (VPS) e os parâmetros que são utilizados por uma ou mais imagens em uma sequência de vídeo codificado podem estar incluídos em um conjunto de parâmetros de sequência (SPS). De modo similar, os parâmetros que são utilizados por uma ou mais fatias em uma imagem podem estar incluídos em um conjunto de parâmetros de imagem (PPS) e os outros parâmetros que são específicos para uma única fatia podem estar incluídos em um cabeçalho de fatia. De modo similar, a indicação de qual(ais) conjunto(s) de parâmetros uma camada particular está usando (por exemplo, ativo) em um determinado tempo pode ser fornecida em vários níveis de codificação. Por exemplo, se o cabeçalho de fatia de uma fatia na camada particular se referir a um PPS, o PPS é ativado para a

fatia ou a imagem que contém a fatia. De forma semelhante, se o PPS se referir a um SPS, o SPS é ativado para a imagem ou a sequência de vídeo codificado que contém a imagem, e se o SPS se referir a um VPS, o VPS é ativado para a sequência de vídeo codificado ou a camada de vídeo que contém a sequência de vídeo codificado.

[00138] Tais conjuntos de parâmetros podem ser fornecidos em um fluxo de bits na forma de unidades de NAL de conjunto de parâmetros (por exemplo, unidade de NAL de SPS, unidade de NAL de PPS, etc.). Uma unidade de NAL pode compreender uma carga útil de sequência de bytes bruta (RBSP) e um cabeçalho de unidade de NAL. A RBSP pode especificar um conjunto de parâmetros ID (por exemplo, ID de SPS) e o cabeçalho de unidade de NAL pode especificar a ID de camada, que pode indicar quais camadas podem usar o SPS.

#### ATIVAÇÃO DE CARGA ÚTIL DE SEQUÊNCIA DE BYTES BRUTA (RBSP) DE CONJUNTO DE PARÂMETROS DE SEQUÊNCIA (SPS)

[00139] Uma RBSP de SPS inclui parâmetros que podem ser referidos por uma ou mais RBSPs de conjunto de parâmetros de imagem (PPS) ou uma ou mais unidades de NAL de SEI que contêm uma mensagem de SEI de conjuntos de parâmetros ativos. Cada RBSP de SPS pode inicialmente ser considerada como não sendo ativa para qualquer camada no início do processo de decodificação. Para cada camada, no máximo uma RBSP de SPS é considerada como sendo ativa em qualquer determinado momento durante o processo de decodificação e a ativação de qualquer RBSP de SPS particular para uma camada particular resulta na desativação da RBSP de SPS anteriormente ativo para aquela camada particular, se houver.

[00140] Uma RBSP de SPS pode ser a RBSP de SPS ativo para mais que uma camada. Por exemplo, se tanto uma

camada de base quanto uma camada de aperfeiçoamento contiverem uma imagem que se refere a um PPS que, por sua vez, se refere a um SPS que tem uma ID de SPS de 3, em que o SPS tem uma ID de SPS de 3 é a RBSP de SPS ativo tanto para a camada de referência quanto para a camada de aperfeiçoamento.

[00141] Quando uma RBSP de SPS (por exemplo, que tem uma ID de SPS particular) já não estiver ativa para uma camada de não base particular (por exemplo, que tem um valor de ID de camada diferente de zero ou uma ID de camada maior que 0) que tem uma ID de camada (por exemplo, `nh_layer_id`) de X e a RBSP de SPS é referida por ativação em uma RBSP de conjunto de parâmetros de imagem (PPS) (por exemplo, em que a ID de PPS é igual ao SPS particular do RBSP de SPS), a RBSP de SPS é ativada para a camada de não base particular. Esse SPS pode ser referido como a RBSP de SPS ativo para a camada de não base particular até que a mesma seja desativada pela ativação de uma outra RBSP de SPS para a camada de não base particular.

#### IMAGEM DE INICIALIZAÇÃO DE CAMADA (LIP)

[00142] Em alguns esquemas de codificação, uma imagem de inicialização de camada ("imagem de LIP") pode ser definida como uma imagem que é uma imagem de IRAP que tem um sinalizador `NoRaslOutputFlag` (por exemplo, um sinalizador que indica que as imagens de RASL não devem ser emitidas se o valor for estabelecido como "1" e indica que as imagens de RASL devem ser emitidas se o valor for estabelecido como "0") estabelecido como um valor de "1" ou uma imagem que está contida em uma unidade de acesso de IRAP inicial, que é uma unidade de acesso de IRAP em que a imagem de camada de base (por exemplo, uma imagem que tem um valor de ID de camada de "0" ou ID de menor camada definido no fluxo de bits) tem o `NoRaslOutputFlag`

estabelecido como um valor de "1".

[00143] Em algumas modalidades, um SPS pode ser ativado em cada imagem de LIP. Por exemplo, cada imagem de IRAP que tem um sinalizador NoRaslOutputFlag estabelecido como um valor de "1" ou cada imagem que está contida em uma unidade de acesso de IRAP inicial, um novo SPS, que pode ser diferente (por exemplo, especificando diferentes resoluções de imagem, etc.) do SPS que foi ativado anteriormente. No entanto, em um caso em que a imagem de LIP não é uma imagem de IRAP (por exemplo, qualquer imagem contida em uma unidade de acesso de IRAP inicial) e a imagem de camada de base na unidade de acesso de IRAP inicial é uma imagem de IDR com um sinalizador NoClrasOutputFlag (por exemplo, um sinalizador que indica que as imagens de salto de acesso aleatório de camada transversal não devem ser emitidas se definidas como sendo 1 e indica que as imagens de salto de acesso aleatório de camada transversal devem ser emitidas se o valor for estabelecido como "0") estabelecido como um valor de "0", não seria permitido que a imagem de LIP ative um novo SPS. Se um novo SPS for ativado em tal imagem de LIP em tal caso, particularmente, quando os conteúdos da RBSP de SPS do novo SPS forem diferentes daqueles do SPS que foi anteriormente ativo antes da unidade de acesso de IRAP inicial, poderia haver problemas em resoluções de imagem de diferenciação e resiliência de erro. Por exemplo, o novo SPS pode atualizar a resolução e previsão temporal de uso para se referir a imagens de tamanhos diferentes.

[00144] Em algumas modalidades, NoClRasOutputFlag e NoRaslOutputFlag podem ser variáveis derivadas com base nas informações incluídas no fluxo de bits. Por exemplo, NoRaslOutputFlag pode ser derivado para cada imagem de IRAP (por exemplo, em BL e/ou EL) e

NoClRasOutputFlag pode ser derivado apenas para as imagens de camada mais baixa (por exemplo, imagens de BL). O valor de cada um dentre NoClRasOutputFlag e NoRaslOutputFlag pode indicar que algumas imagens no fluxo de bits não podem ser corretamente decodificáveis devido à indisponibilidade de certas imagens de referência. Tal indisponibilidade de imagens de referência pode ocorrer em pontos de acesso aleatório. As imagens de omissão de acesso aleatório entre camadas (CL-RAS) são, de algumas maneiras, o equivalente de camada múltipla das imagens de RASL. Se um decodificador começar a decodificar um fluxo de bits em um ponto de acesso aleatório (por exemplo, uma unidade de acesso que tem uma imagem de IRAP de BL) e a imagem EL na unidade de acesso não for uma imagem de IRAP, então, aquela imagem EL é uma imagem de CL-RAS. Todas as imagens na EL podem ser imagens de CL-RAS (por exemplo, decodificáveis, mas não corretamente decodificáveis) até uma imagem de IRAP ocorrer na EL. Quando tal imagem EL de IRAP for fornecida no fluxo de bits, pode-se dizer que a EL foi inicializada.

#### DURAÇÃO DE RBSP DE SPS ATIVO

[00145] A fim de impedir a ativação de um novo SPS por uma imagem de LIP que é uma imagem de não IRAP além de em um ponto de emenda, que pode causar os problemas descritos acima, o RBSP de SPS ativado pode ser forçado a permanecer ativo por um determinado período de tempo. Em algumas modalidades, um RBSP de SPS ativado para uma camada particular deve permanecer ativo pela sequência de vídeo em camadas codificada inteira (CLVS) daquela camada particular. Um CLVS pode se referir a uma sequência de imagens codificadas, que estão na mesma camada (por exemplo, que tem o mesmo valor de ID de camada), que consiste, na ordem de decodificação, em uma imagem de IRAP com o NoRaslOutputFlag igual a um valor de "1" ou uma

imagem com o FirstPicInLayerDecodedFlag igual a um valor de "0" (por exemplo, indicando que a imagem é a primeira imagem na camada), seguida por todas as imagens codificadas, se houver, até, mas com a exceção da próxima imagem de IRAP com o NoRaslOutputFlag igual a um valor de "1" ou da próxima imagem com o FirstPicInLayerDecodedFlag igual a um valor de "0".

[00146] Em algumas modalidades, um RBSP de SPS ativado para uma camada particular deve permanecer ativa para uma sequência de imagens na ordem de decodificação na camada particular conforme a seguir: (1) Se o valor de nuh\_layer\_id for igual a 0, o SPS ativado permanecerá ativo pelo CVS inteiro; e (2) de outra forma (por exemplo, o ID de camada da camada particular não é igual a 0), o SPS ativado deve permanecer ativo com início a partir de uma imagem de LIP que é uma imagem de IRAP na camada particular, ou uma imagem de LIP que não é uma imagem de IRAP e que está contida em uma unidade de acesso de IRAP que contém uma imagem de camada de base que é uma imagem de IRAP e tem um valor de NoClrasOutputFlag de "1", até a próxima imagem de LIP que é uma imagem de IRAP na camada particular, ou uma unidade de acesso de IRAP inicial que contém uma imagem de camada de base que é uma imagem de IRAP e tem um valor de NoClrasOutputFlag de "1".

#### FLUXO DE BITS QUE INCLUI UM PONTO DE EMENDA

[00147] Em referência à Figura 4, um exemplo fluxo de bits que tem um ponto de emenda será descrito. A Figura 4 mostra um fluxo de bits de camada múltipla 400 criado através da junção de fluxos de bits 410 e 420. O fluxo de bits 410 incluem uma camada de aperfeiçoamento (EL) 410A e uma camada de base (BL) 410B e o fluxo de bits 420 inclui uma EL 420A e uma BL 420B. A EL 410A inclui uma imagem EL 412A e a BL 410B inclui uma imagem de BL 412B. A

EL 420A inclui as imagens de EF 422A, 424A e 426A e a BL 420B inclui as imagens de BL 422B, 424B e 426B. O fluxo de bits de camada múltipla 400 inclui adicionalmente unidades de acesso (AUs) 430 a 460. A AU 430 inclui a imagem EL 412A e a imagem de BL 412B, a AU 440 inclui a imagem EL 422A e a imagem de BL 422B, a AU 450 inclui a imagem EL 424A e a imagem de BL 424B e a AU 460 inclui a imagem EL 426A e a imagem de BL 426B. No exemplo da Figura 4, a imagem de BL 422B é uma imagem de IRAP e a imagem EL correspondente 422A na AU 440 é uma imagem posterior (por exemplo, uma imagem de não IRAP) e, consequentemente, a AU 440 é uma AU de IRAP não alinhada. Além disso, deve ser observado que a AU 440 é uma unidade de acesso que segue imediatamente um ponto de emenda 470.

[00148] Embora o exemplo da Figura 4 ilustre um caso em que dois diferentes fluxos de bits são unidos, em algumas modalidades, um ponto de emenda pode estar presente quando uma porção do fluxo de bits for removida. Por exemplo, um fluxo de bits pode ter porções A, B, e C, em que a porção B está entre as porções A e C. Se a porção B for removida do fluxo de bits, as porções restantes A e C podem ser unidas e o ponto em que as mesmas são unidas pode ser referido como um ponto de emenda. Mais geralmente, considera-se que um ponto de emenda conforme discutido no presente pedido possa estar presente quando uma ou mais sinalizadores ou parâmetros sinalizados ou derivado têm valores predeterminados. Por exemplo, sem receber uma indicação específica de que um ponto de emenda existe em uma localização particular, a decodificador pode determinar o valor de um sinalizador (por exemplo, NoClrasOutputFlag) e realizar uma ou mais técnicas descritas neste pedido com base no valor do sinalizador.

RESTRICÃO DE ATIVAÇÃO DE NOVO CONJUNTO DE

## PARÂMETROS

[00149] A Figura 5 é um fluxograma que ilustra um método 500 para codificar informações de vídeo, de acordo com uma modalidade da presente descrição. As etapas ilustradas na Figura 5 podem ser realizadas por um codificador (por exemplo, o codificador de vídeo conforme mostrado na Figura 2A ou à Figura 2B), um decodificador (por exemplo, o decodificador de vídeo conforme mostrado na Figura 3A ou à Figura 3B) ou qualquer outro componente. Para conveniência, o método 500 é descrito conforme realizado por um codificador, que pode ser o codificador, o decodificador ou um outro componente.

[00150] O método 500 começa no bloco 501. No bloco 505, o codificador determina se uma imagem de camada de aperfeiçoamento (EL) é uma imagem de IRAP. Por exemplo, se imagem EL for qualquer uma dentre de uma imagem de IDR, uma imagem de CRA, ou uma imagem de BLA, o codificador pode determinar a imagem EL como uma imagem de IRAP. Em algumas modalidades, o codificador determinar que a imagem EL é uma imagem de IRAP verificando-se o tipo de unidade de NAL associado à imagem EL. Se o codificador determinar que a imagem EL é uma imagem de IRAP, o método 500 prossegue para o bloco 520. Se o codificador determinar que a imagem EL não é uma imagem de IRAP, o processo 500 prossegue para o bloco 510.

[00151] No bloco 510, o codificador determinar se a imagem EL está em um ponto de emenda. Um ponto de emenda pode indicar o ponto em que dois fluxos de bits se unem. Por exemplo, a última imagem de um primeiro fluxo de bits pode ser imediatamente seguida, na ordem de codificação, pela primeira imagem de um segundo fluxo de bits. Em algumas modalidades, o codificador pode determinar se a imagem EL está em um ponto de emenda determinando-se

se a imagem EL é a primeira imagem que segue o ponto de emenda na ordem de codificação. Se o codificador determinar que a imagem EL está em um ponto de emenda, o método 500 prossegue para o bloco 520. Se o codificador determinar que a imagem EL não está em um ponto de emenda, o método 500 prossegue para o bloco 515. Conforme notado acima com referência à Figura 4, a determinação em 510 pode compreender determinar se um ou mais sinalizadores ou parâmetros sinalizados ou derivados têm valores predeterminados. Por exemplo, sem receber uma indicação específica de que um ponto de emenda existe em um local, um codificador ou um decodificador pode determinar o valor de um sinalizador (por exemplo, NoClrasOutputFlag), e prosseguir para o bloco 515 ou o bloco 520 com base na determinação.

[00152] No bloco 515, o codificador impede a ativação de um novo conjunto de parâmetros. Por exemplo, o codificador pode impedir que a imagem EL ative um novo SPS. O codificador pode impedir a ativação de um novo conjunto de parâmetros fazendo-se com que o conjunto de parâmetros ativado anteriormente permaneça ativo até a próxima imagem de IRAP na EL ser processada. Alternativamente, o codificador pode impedir a ativação de um novo conjunto de parâmetros fazendo-se com que o conjunto de parâmetros ativado anteriormente permaneça ativo até o próximo ponto de emenda. Em ainda outro exemplo, o codificador pode impedir a ativação de um novo conjunto de parâmetros fazendo-se com que o conjunto de parâmetros ativado anteriormente permaneça ativo até o fim da sequência de vídeo codificado que contém a imagem EL.

[00153] No bloco 520, o codificador permite a ativação de um novo conjunto de parâmetros. Por exemplo, o codificador pode fazer com que a camada de aperfeiçoamento

seja associada a um novo conjunto de parâmetros que tem parâmetros diferentes do conjunto de parâmetros que foi ativado anteriormente para a EL. O método 500 termina em 525.

[00154] Conforme discutido acima, um ou mais componentes de codificador de vídeo 20 da Figura 2A, codificador de vídeo 23 da Figura 2B, decodificador de vídeo 30 da Figura 3A ou decodificador de vídeo 33 da Figura 3B (por exemplo, a unidade de predição intercamada 128 e/ou unidade de predição intercamada 166) pode ser usado para implantar qualquer uma das técnicas discutidas na presente descrição, como determinação de se a imagem EL é uma imagem de IRAP, determinação de se a imagem EL está em um ponto de emenda, e impedir ou permitir a ativação de um novo conjunto de parâmetros.

[00155] No método 500, um ou mais dos blocos mostrados na Figura 5 podem ser removidos (por exemplo, não realizados) e/ou a ordem em que o método é realizado pode ser comutada. Por exemplo, embora o bloco 510 seja mostrado na Figura 5, o mesmo pode ser removido para simplificar o processo de codificação. Desse modo, as modalidades da presente descrição não são limitadas a ou pelo exemplo mostrado na Figura 5 e outras variações podem ser implantadas sem se afastar do espírito desta descrição.

#### RESTRICOES EM CONJUNTOS DE PARÂMETROS NUNCA ATIVADOS

[00156] Em alguns esquemas de codificação, pode haver restrições de conformidade de fluxo de bits (por exemplo, restrições que podem ser determinada como aplicáveis e então aderidas a, por exemplo, um codificador) especificado para conjuntos de parâmetros ativos (por exemplo, PPS ativo ou SPS ativo), mostrados em itálico: "pps\_scaling\_list\_ref\_layer\_id" especifica o valor do

nuh\_layer\_id da camada para a qual o PPS ativo está associado com os mesmos dados de lista de escalonamento como o PPS atual. O valor de pps\_scaling\_list\_ref\_layer\_id precisa estar na faixa de 0 a 62, inclusive. Quando o sinalizador avc\_base\_layer\_flag for igual a 1, o mesmo é um requisito de conformidade de fluxo de bits que pps\_scaling\_list\_ref\_layer\_id seja maior do que 0. É um requisito de conformidade de fluxo de bits que, quando um PPS com nuh\_layer\_id igual a nuhLayerIdA estiver ativo para uma camada com nuh\_layer\_id igual a nuhLayerIdB e pps\_infer\_scaling\_list\_flag no PPS for igual a 1, o pps\_infer\_scaling\_list\_flag será igual a 0 para o PPS que está ativo para a camada com nuh\_layer\_id igual a pps\_scaling\_list\_ref\_layer\_id. É um requisito da conformidade de fluxo de bits que, quando um PPS com nuh\_layer\_id igual a nuhLayerIdA estiver ativo para uma camada com nuh\_layer\_id igual a nuhLayerIdB, a camada com nuh\_layer\_id igual a pps\_scaling\_list\_ref\_layer\_id será uma camada de referência direta ou indireta da camada com nuh\_layer\_id igual a nuhLayerIdB.

[00157] "Em alguns esquemas de codificação, um conjunto de parâmetros que nunca é ativado por qualquer camada pode ser, independentemente, necessário para satisfazer as seguintes restrições, mostradas em itálico: "Todas as restrições que são expressas na relação entre os valores dos elementos de sintaxe e os valores de variáveis derivadas daqueles elementos de sintaxe em VPPSs, SPPSs e PPSs e outros elementos de sintaxe são expressões de restrições que se aplicam somente aos VPS ativo, o SPS ativo, e o PPS ativo. Se qualquer RBSP de VPS, RBSP de SPS e RBSP de PPS estiver presente que nunca é ativado no fluxo de bits, seus elementos de sintaxe terão os valores que se conformariam às restrições especificadas se o mesmo fosse

ativado a título de referência em um fluxo de bits de outra forma conformante."

[00158] Entretanto, para conjuntos de parâmetros como VPS, SPS, PPS, etc. que nunca são ativados, pode ser difícil verificar se as primeiras restrições em itálico acima estão satisfeitas para os conjuntos de parâmetros que estão em um fluxo de bits, mas nunca ativado, considerando até mesmo a segunda restrição em itálico. Isso se deve ao fato de que se um dado conjunto de parâmetros satisfaz as restrições se estiver ativado para uma camada, o mesmo conjunto de parâmetros pode não satisfazer as restrições se ativado para outra camada. Múltiplos conjuntos de parâmetros podem ser sinalizados para uma dada camada através do fluxo de bits (por exemplo, antes da ativação, independentemente se os mesmos são ativados ou não), então se um fluxo de bits particular é conformante não pode ser determinado a menos que aqueles conjuntos de parâmetros sejam ativados para uma ou mais camadas.

#### MUDANÇAS ÀS RESTRIÇÕES EM CONJUNTOS DE PARÂMETROS NUNCA ATIVADOS

[00159] Em algumas modalidades, a verificação de conformidade é excluída do elemento de sintaxe de verificação de conformidade de fluxo de bits do conjunto de parâmetros nunca ativado cuja restrição de conformidade de fluxo de bits depende de outros elementos de sintaxe de outros conjuntos de parâmetros. A restrição discutida acima pode ser modificada conforme a seguir: "Se qualquer RBSP de VPS, RBSP de SPS e RBSP de PPS estiver presente que não é ativado no fluxo de bits, os elementos de sintaxe daquele RBSP de conjunto de parâmetros particular terá valores que se conformariam àquelas restrições especificadas que não são dependentes do valor de qualquer elemento de sintaxe de

outro RBSP de conjunto de parâmetros

[00160] Em algumas modalidades, alguns elementos de sintaxe particulares do conjunto de parâmetros nunca ativado podem ser omitidos da verificação de conformidade de fluxo de bits de acordo com alguma regra. Por exemplo, a verificação de conformidade verifica a restrição de conformidade para o conjunto de parâmetros nunca ativado para a camada para a qual esse conjunto de parâmetros é sinalizado. A restrição discutida acima pode ser modificada conforme a seguir: Se qualquer RBSP de VPS, RBSP de SPS e RBSP de PPS estiver presente que nunca é ativado no fluxo de bits, seus elementos de sintaxe terão os valores que se conformariam às restrições especificadas se o mesmo fosse ativado a título de referência com o Id de camada sinalizado no conjunto de parâmetros em um fluxo de bits de outra forma conformante."

#### INFORMAÇÕES DE USABILIDADE DE VÍDEO DE (VUI) SPS

[00161] As VUI de SPS incluem informações como razão de aspecto de amostra, overscan, formato de fonte de vídeo (PAL/NTSC etc., faixa de valor de amostra, formato de cor de fonte), informações de campo/quadro, restrições de fluxo de bits (incluindo as restrições de fluxo de bits entre camadas). Tais informações podem incluir restrições de fluxo de bits entre camadas, o que pode não ser específico de camada, mas, ao invés disso, se aplicar a todas as camadas. Portanto, em alguns esquemas de codificação existentes, tais informações podem ser repetidas em cada SPS.

[00162] Em algumas modalidades, todas as informações VUI de SPS podem estar incluídas nas VUI de VPS, e se para uma camada particular qualquer parâmetro de VUI para um SPS deve ter um valor diferentes daquele nas VUI de VPS, as VUI de SPS inteiras podem ser sinalizadas

diretamente no SPS. De outra forma, o SPS não contém parâmetros de VUI e as VUI são inferidas a partir do VPS.

[00163] As alterações exemplificativas de sintaxe e semântica que podem ser implantadas para facilitar o manuseio das informações descritas acima, as alterações a seguir podem ser feitas, quando adições são mostradas em itálico:

vps_vui( ){	Descriptor
...	
<i>sps_vui_in_vps_flag</i>	<i>u(1)</i>
<i>if( sps_vui_in_vps_flag )</i>	
<i>vui_parameters()</i>	
}	

TABELA 1: SINTAXE EXEMPLIFICATIVA DE VPS\_VUI( )

[00164] A semântica para *sps\_vui\_in\_vps\_flag* pode ser lida como a seguir: "*sps\_vui\_in\_vps\_flag* igual a 1 especifica que a estrutura de sintaxe *vui\_parameters()* está presente no VPS e se aplica a todos os SPSs que se referem ao VPS e têm *sps\_vui\_from\_vps\_flag* igual a 1. *sps\_vui\_in\_vps\_flag* igual a 0 especifica que a estrutura de sintaxe *vui\_parameters()* não está presente no VPS. Quando *vps\_vui\_present\_flag* for igual a 0, o valor de *sps\_vui\_in\_vps\_flag* é inferido como igual a 0."

seq_parameter_set_rbsp()	Descriptor
...	
if( <i>nuh_layer_id</i> > 0 )	
<b>sps_vui_from_vps_flag</b>	<i>u(1)</i>
if( ! <i>sps_vui_from_vps_flag</i> )	
<b>vui_parameters_present_flag</b>	<i>u(1)</i>
if( <i>vui_parameters_present_flag</i> )	
vui_parameters()	
<b>sps_extension_flag</b>	<i>u(1)</i>
if( <i>sps_extension_flag</i> ) {	
sps_extension()	
<b>sps_extension2_flag</b>	<i>u(1)</i>
if( <i>sps_extension2_flag</i> )	
while( more_rbsp_data() )	
<b>sps_extension_data_flag</b>	<i>u(1)</i>
}	
rbsp_trailing_bits()	
}	

TABELA 2 SINTAXE EXEMPLIFICATIVA PARA  
SEQ\_PARAMETER\_SET\_RBSP()

[00165] A semântica para *sps\_vui\_from\_vps\_flag* pode ser lida como a seguir: "*sps\_vui\_from\_flag* igual a 1 especifica que a estrutura de sintaxe *vui\_parameters()* aplicável ao SPS é inferida como idêntica àquela no VPS ao qual o SPS se refere. *sps\_vui\_from\_vps\_flag* igual a 0 especifica que a estrutura de sintaxe *vui\_parameters()* aplicável ao SPS não é inferida do VPS ao qual o SPS se refere. Quando não estiver presente, o valor de *sps\_vui\_from\_vps\_flag* é inferido como igual a 0. Quando *sps\_vui\_in\_vps\_flag* for igual a 0, o valor de

`sps_vui_from_vps_flag` será igual a 0. "

[00166] A semântica para `vui_parameters_present_flag` pode ser lida como a seguir: "`vui_parameters_present_flag` igual a 1 especifica que a estrutura de sintaxe `vui_parameters()` conforme especificada em Annex E está presente. `vui_parameters_present_flag` igual a 0 especifica que a estrutura de sintaxe `vui_parameters()` conforme especificada em Annex E não está presente. Quando não estiver presente, o valor de `vui_parameters_present_flag` é inferido como igual a 0. "

#### OUTRAS CONSIDERAÇÕES

[00167] As informações e os sinais revelados podem ser representados com uso de qualquer um dentre uma variedade de tecnologias e técnicas diferentes. Por exemplo, dados, instruções, comandos, informações, sinais, bits, símbolos, e circuitos integrados que podem ser referenciados por toda a descrição acima podem ser representados por tensões, correntes, ondas eletromagnéticas, partículas ou campos magnéticos, partículas ou campos ópticos ou qualquer combinação dos mesmos.

[00168] As várias etapas de blocos lógicos ilustrativos, módulos, circuitos e algoritmo descritos em conjunto com as modalidades reveladas no presente documento podem ser implantadas como hardware eletrônico, software de computador ou combinações de ambos. Para ilustrar claramente essa intercambiabilidade de hardware e software, vários componentes, blocos, módulos, circuitos, e etapas ilustrativos têm sido descritos acima geralmente em termos de funcionalidade dos mesmos. Se tal funcionalidade é implantada como hardware ou software, depende das restrições de projeto e pedido particular impostas no sistema geral. Os indivíduos versados na técnica podem

implantar a funcionalidade descrita em modos variáveis para cada aplicação em particular, porém tais decisões de implantação não devem ser interpretadas como causadoras de um afastamento do escopo da presente descrição.

[00169] As técnicas descritas no presente documento podem ser implantadas em hardwares, softwares, firmwares ou qualquer combinação dos mesmos. Tais técnicas podem ser implantadas em qualquer um dentre uma variedade de dispositivos tal como computadores de propósito geral, fones de comunicação sem fio dispositivo ou dispositivos de circuito integrado que têm múltiplos usos que incluem aplicação em fones de comunicação sem fio e outros dispositivos. Quaisquer recursos descritos como módulos ou componentes podem ser implantados juntamente em um dispositivo lógico integrado ou separadamente como separados, mas dispositivos lógicos interoperáveis. Se implantadas em software, as técnicas podem ser realizadas pelo menos em parte por um meio de armazenamento de dados legível por computador que compreende código de programa que inclui instruções que, quando executadas, realizam um ou mais dos métodos descritos acima. O meio de armazenamento de dados legível por computador pode formar parte de um produto de programa de computador, que pode incluir materiais para pacotes. O meio legível por computador pode compreender meios de armazenamento de dados ou de memória, tal como memória de acesso aleatório (RAM) tal como memória de acesso aleatório dinâmico sincrônico (SDRAM), memória apenas de leitura (ROM), memória de acesso aleatório não volátil (NVRAM), memória apenas para leitura programável eletricamente apagável (EEPROM), memória flash, meios de armazenamento de dados ópticos ou magnéticos e similares. As técnicas adicional ou alternativamente, podem ser realizadas pelo menos em parte por um meio de

comunicação legível por computador que transporta ou comunica o código de programa na forma de instruções ou estruturas de dados e que podem ser acessadas, lidas e/ou executadas por um computador, tal como ondas ou sinais propagados.

[00170] O código de programa pode ser executado por um processador, que pode incluir um ou mais processadores, tais como um ou mais processadores de sinal digital (DSPs), microprocessadores de propósito geral, circuitos integrados para aplicação específica (ASICs), matrizes lógicas programáveis em campo (FPGAs) ou outro conjunto de circuitos lógico distinto ou integrado equivalente. Tal processador pode ser configurado para realizar qualquer uma das técnicas descritas nesta descrição. Um processador de propósito geral pode ser um microprocessador, mas, na alternativa, o processador pode ser qualquer processador, controlador, microcontrolador ou máquina de estado convencional. Um processador também pode ser implantado como uma combinação de dispositivos de computação, por exemplo, uma combinação de um DSP e um microprocessador, uma pluralidade de microprocessadores, um ou mais microprocessadores em conjunto com um DSP núcleo ou qualquer outra tal configuração. Consequentemente, o termo "processador", conforme usado no presente documento pode se referir a qualquer um dentre a estrutura anterior, qualquer combinação da estrutura anterior ou qualquer outra estrutura ou aparelho adequado para a implantação dos conjuntos de procedimentos descritos no presente documento. Além disso, em alguns aspectos, a funcionalidade descrita no presente documento pode ser fornecida dentro dos módulos de hardware ou módulos de software dedicados configurados para codificar e decodificar ou incorporados em um codificador-decodificador de vídeo combinado (CODEC).

Também, as técnicas podem ser totalmente implantadas em um ou mais circuitos ou elementos lógicos.

[00171] As técnicas desta descrição podem ser implantadas em uma ampla variedade de dispositivos ou aparelhos, incluindo um monofone, um circuito integrado (IC) ou um conjunto de ICs (por exemplo, um conjunto de chips). Vários componentes, módulos ou unidades são descritos nessa descrição para enfatizar os aspectos funcionais dos dispositivos configurados para realizar as técnicas reveladas, mas não exigem necessariamente a realização por diferentes unidades de hardware. Em vez disso, conforme descrito acima, várias unidades podem ser combinadas em uma unidade de hardware de codec ou fornecidas por uma coleção de unidades de hardware interoperacionais, que incluem um ou mais processadores conforme descrito acima, em conjunto com software e/ou firmware adequados.

[00172] Várias modalidades da invenção foram descritas. Essas e outras modalidades estão dentro do escopo das reivindicações a seguir.

## REIVINDICAÇÕES

1. Aparelho configurado para codificar (code) informações de vídeo, o aparelho compreendendo:

uma unidade de memória configurada para armazenar informações de vídeo associadas a uma camada de referência (RL) e uma camada de aperfeiçoamento (EL), a RL tendo uma imagem RL em uma primeira unidade de acesso e a EL tendo uma primeira imagem EL na primeira unidade de acesso, em que uma segunda imagem EL que precede imediatamente a primeira imagem EL na ordem de codificação (coding) está associada a um primeiro conjunto de parâmetros; e

um processador em comunicação com a unidade de memória, o processador configurado para:

determinar que a primeira imagem EL não é uma imagem de ponto de acesso intra-aleatório (IRAP);

o aparelho caracterizado pelo fato de que o processador é adicionalmente configurado para:

determinar que um primeiro sinalizador associado com a imagem RL tem um primeiro valor indicativo de imagens de salto de acesso aleatório de camada transversal sendo permitidas de serem emitidas ou um segundo valor indicativo de imagens de salto de acesso aleatório de camada transversal não sendo permitidas de serem emitidas; e

com base em uma determinação de que a primeira imagem EL não é uma imagem IRAP e que o sinalizador associado com a imagem RL tem um primeiro valor indicativo de imagens de salto de acesso aleatório de camada transversal sendo permitidas para emissão, abster-se de ativar um segundo conjunto de parâmetros que é diferente do primeiro conjunto de parâmetros na primeira imagem EL tal que a o primeiro

conjunto de parâmetros permanece ativo na primeira imagem EL, ou com base em uma determinação de que a primeira imagem EL não é uma imagem IRAP e que o sinalizador associado com a imagem RL tem o segundo valor, associar a primeira imagem EL a um segundo conjunto de parâmetros que é diferente do primeiro conjunto de parâmetros tal que um novo conjunto de parâmetros, diferente do conjunto de parâmetros ativo previamente, se torna a primeira imagem EL;

codificar (encode) ou decodificar a primeira imagem EL em um fluxo de bits com base no primeiro conjunto de parâmetros;

determinar que a primeira imagem EL é indicada para não ser uma imagem IRAP;

determinar que a primeira imagem EL é uma imagem IRAP com base pelo menos em parte em um sinalizador ou um elemento de sintaxe que indica se a primeira imagem EL é uma imagem IRAP;

em que a primeira imagem EL não segue imediatamente um ponto de emenda em ordem de codificação e a segunda imagem EL não precede imediatamente um ponto de emenda em ordem de codificação.

2. Aparelho configurado para codificar informações de vídeo, de acordo com a reivindicação 1, caracterizado pelo fato de que o processador é configurado adicionalmente para:

em resposta à abster-se de ativar um segundo conjunto de parâmetros, fazer com que a EL permaneça associada ao primeiro conjunto de parâmetros pelo menos até um próximo ponto de emenda na EL.

3. Aparelho configurado para codificar informações de vídeo, de acordo com a reivindicação 1, caracterizado pelo fato de que o processador é configurado adicionalmente para:

fazer com que, com base em uma determinação de que a primeira imagem EL é uma imagem IRAP, a primeira imagem EL seja associada ao segundo conjunto de parâmetros.

4. Aparelho configurado para codificar informações de vídeo, de acordo com a reivindicação 1, caracterizado pelo fato de que o processador é configurado para abster-se de associar a primeira imagem EL ao segundo conjunto de parâmetros pelo menos em parte ao:

fazer com que a EL permaneça associada ao primeiro conjunto de parâmetros pelo menos até que uma próxima imagem IRAP na EL seja processada.

5. Aparelho configurado para codificar informações de vídeo, de acordo com a reivindicação 1, caracterizado pelo fato de que o processador é configurado para determinar se a primeira imagem EL é uma imagem IRAP pelo menos em parte processando-se informações de cabeçalho associadas à primeira imagem EL.

6. Aparelho configurado para codificar informações de vídeo, de acordo com a reivindicação 1, caracterizado pelo fato de que a imagem RL é associada ao primeiro conjunto de parâmetros, e em que o processador é configurado adicionalmente para fazer com que a RL permaneça associada ao primeiro conjunto de parâmetros para toda uma sequência de vídeo codificada.

7. Aparelho configurado para codificar informações de vídeo, de acordo com a reivindicação 1, caracterizado

pelo fato de que o aparelho compreende pelo menos um dentre um codificador (encoder) ou um decodificador, e em que o processador é configurado adicionalmente para pelo menos um entre codificar (encode) ou decodificar as informações de vídeo em um fluxo de bits.

8. Aparelho configurado para codificar informações de vídeo, de acordo com a reivindicação 1, caracterizado pelo fato de que o aparelho compreende um dispositivo selecionado a partir de um grupo que consiste em: um computador, um notebook, um computador do tipo laptop, um computador do tipo tablet, um set-top box, um fone de telefone, um telefone inteligente, um smartpad, uma televisão, uma câmera, um dispositivo de exibição, um reproduutor de mídia digital, um console de jogos eletrônicos e um computador de bordo.

9. Método para codificar (encoding) informações de vídeo, o método compreendendo:

determinar que uma primeira imagem de camada de aperfeiçoamento (EL) em uma primeira unidade de acesso de uma EL não é uma imagem de ponto de acesso intra-aleatório (IRAP), em que uma segunda imagem EL que precede imediatamente a primeira imagem EL na ordem de codificação está associada a um primeiro conjunto de parâmetros;

o método caracterizado pelo fato de que compreende adicionalmente:

determinar que um sinalizador associado com a imagem RL tem um primeiro valor indicativo de imagens de salto de acesso aleatório de camada transversal sendo permitidas de serem emitidas ou um segundo valor indicativo

de imagens de salto de acesso aleatório de camada transversal não sendo permitidas de serem emitidas; e

com base em uma determinação de que a primeira imagem EL não é uma imagem IRAP e que o sinalizador associado com a imagem RL tem um valor indicativo de imagens de salto de acesso aleatório de camada transversal, abster-se de ativar um segundo conjunto de parâmetros que é diferente do primeiro conjunto de parâmetros na primeira imagem EL tal que o primeiro conjunto de parâmetros permanece ativo na primeira imagem EL, ou com base em uma determinação de que a primeira imagem EL não é uma imagem IRAP e que o sinalizador associado com a imagem RL tem um segundo valor, associar a primeira imagem EL a um segundo conjunto de parâmetros que é diferente do primeiro conjunto de parâmetros tal que um novo conjunto de parâmetros, diferente do conjunto de parâmetros ativo previamente se torna a primeira imagem EL;

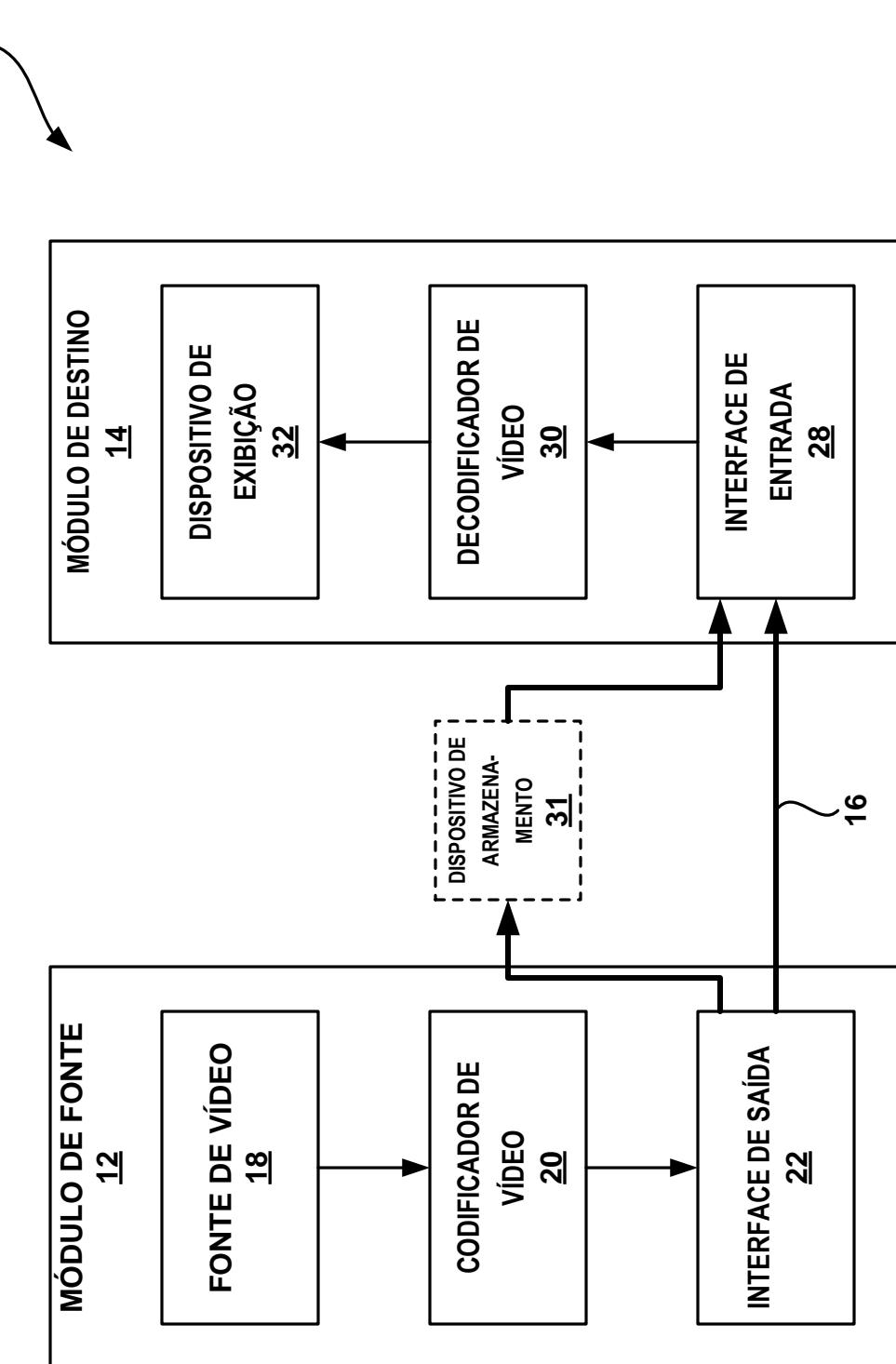
codificar (encoding), com base no primeiro conjunto de parâmetros, a primeira imagem EL em um fluxo de bits;

determinar que a primeira imagem EL é indicada para não ser uma imagem IRAP; e

determinar que a primeira imagem EL é uma imagem IRAP com base pelo menos em parte em um sinalizador ou um elemento de sintaxe que indica se a primeira imagem EL é uma imagem IRAP;

em que a primeira imagem EL não segue imediatamente um ponto de emenda em ordem de codificação e a segunda imagem EL não precede imediatamente um ponto de emenda em ordem de codificação.

10. Memória legível por computador, caracterizada pelo fato de que possui instruções armazenadas na mesma que, quando executadas, fazem com que um computador realize o método conforme definido na reivindicação 9.



**FIG. 1A**

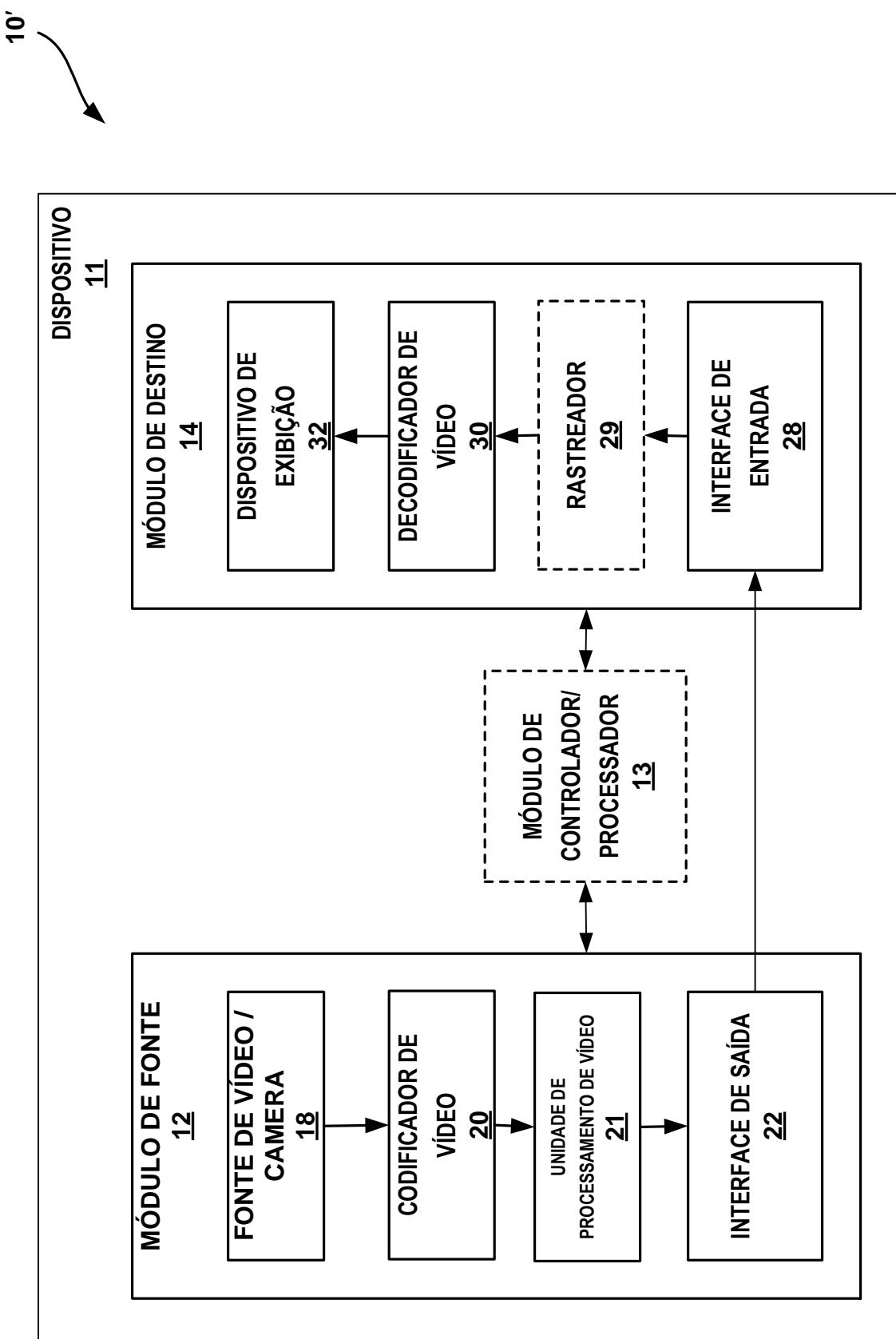


FIG. 1B

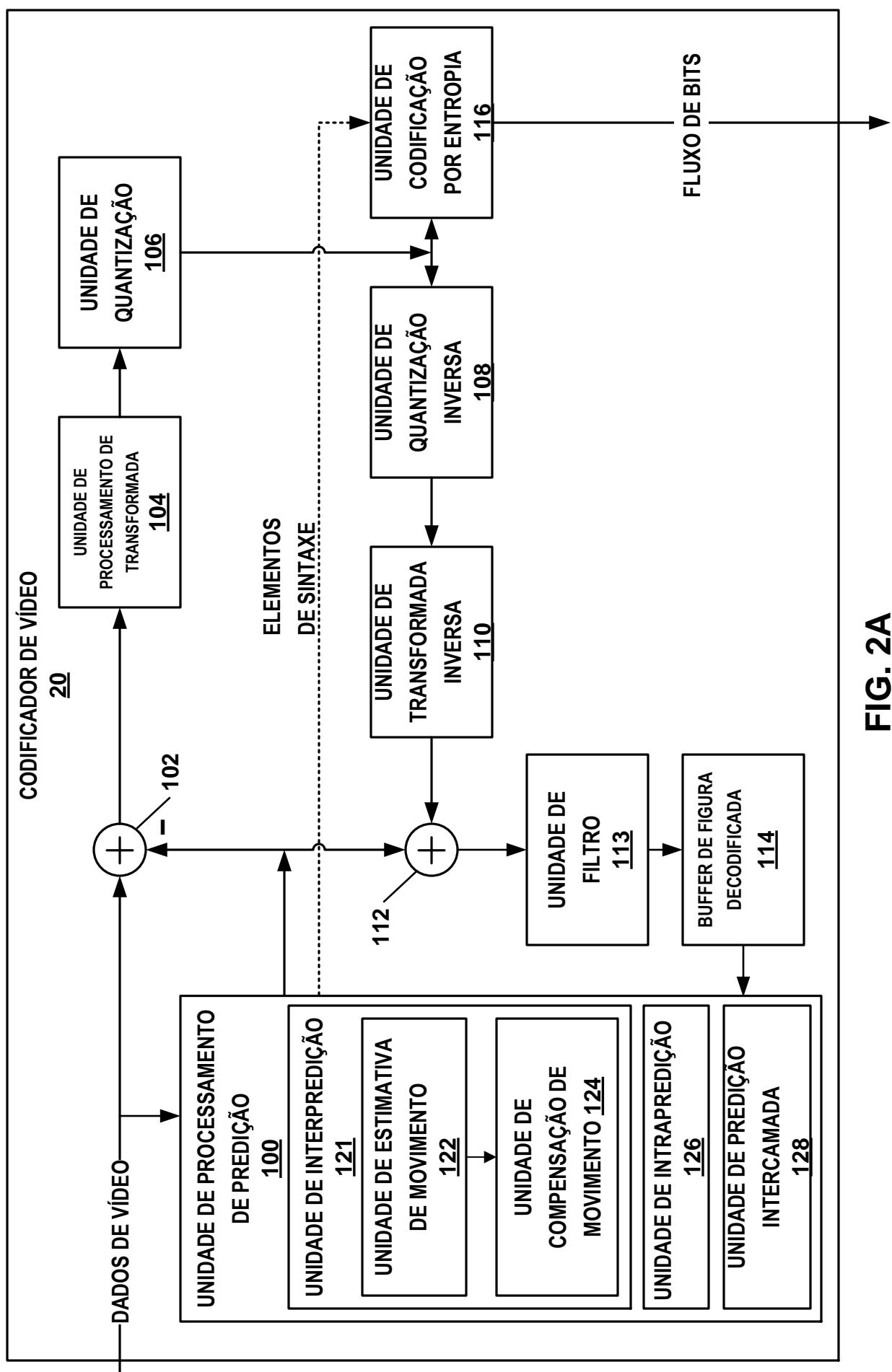
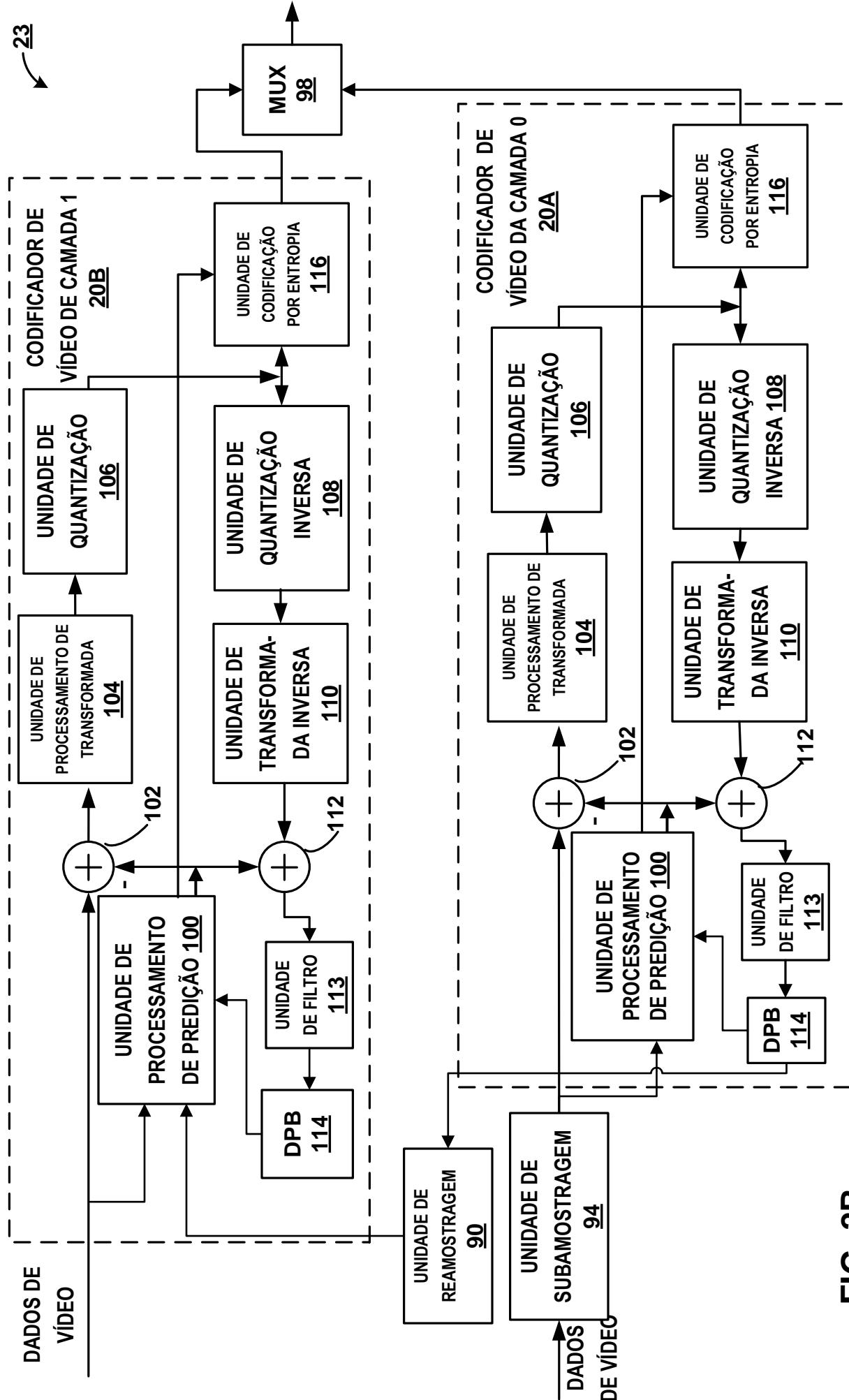
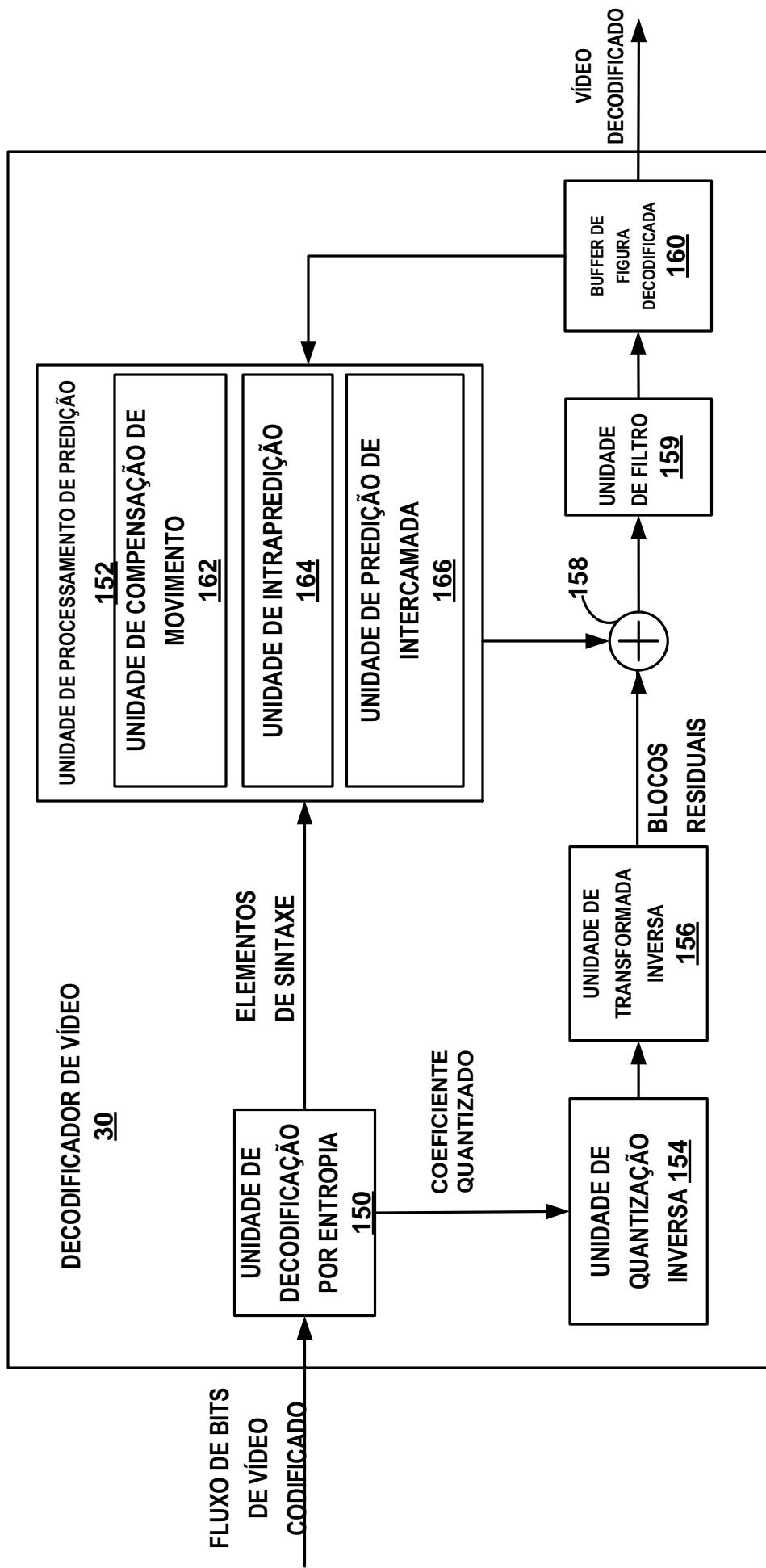
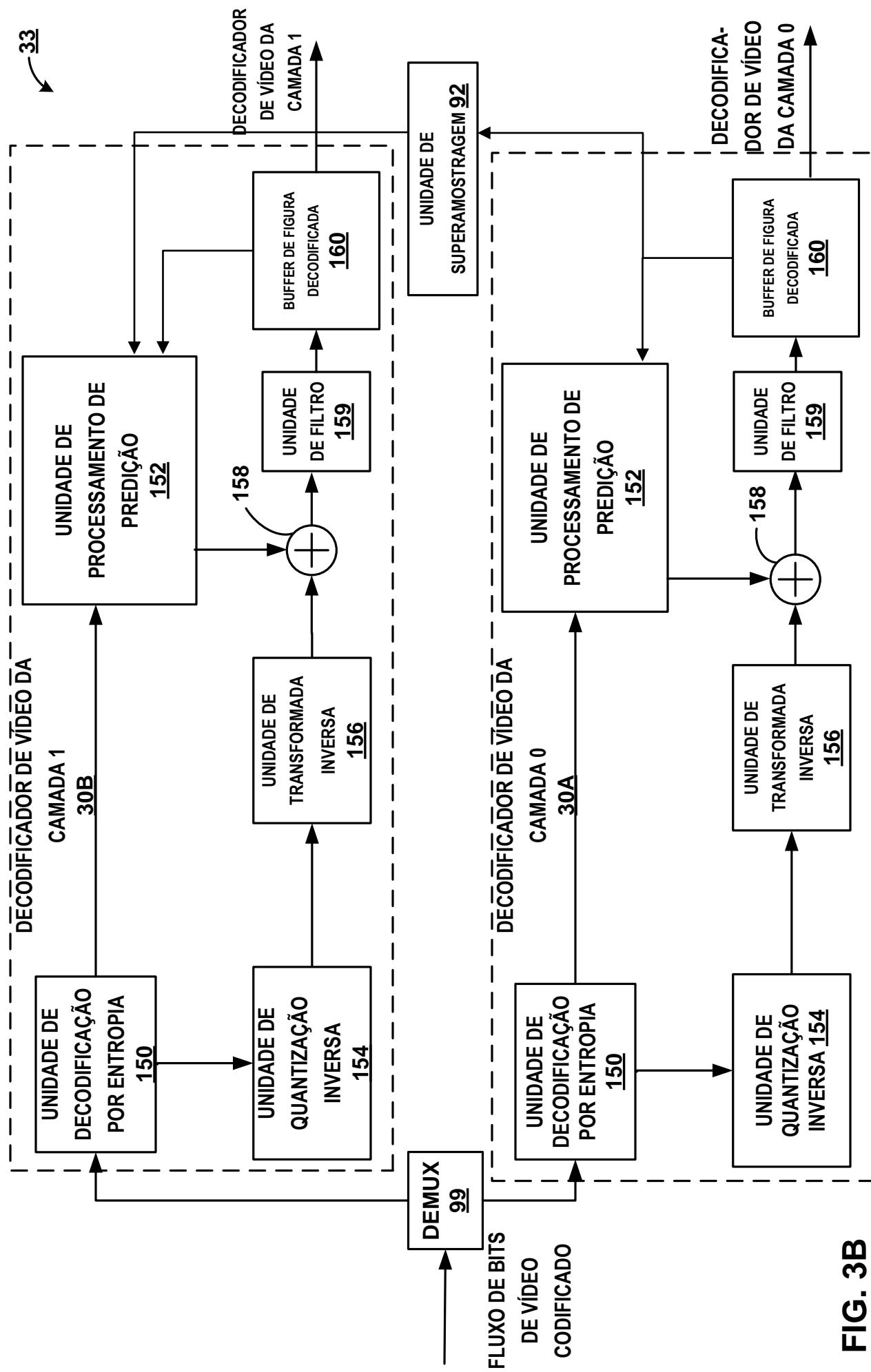


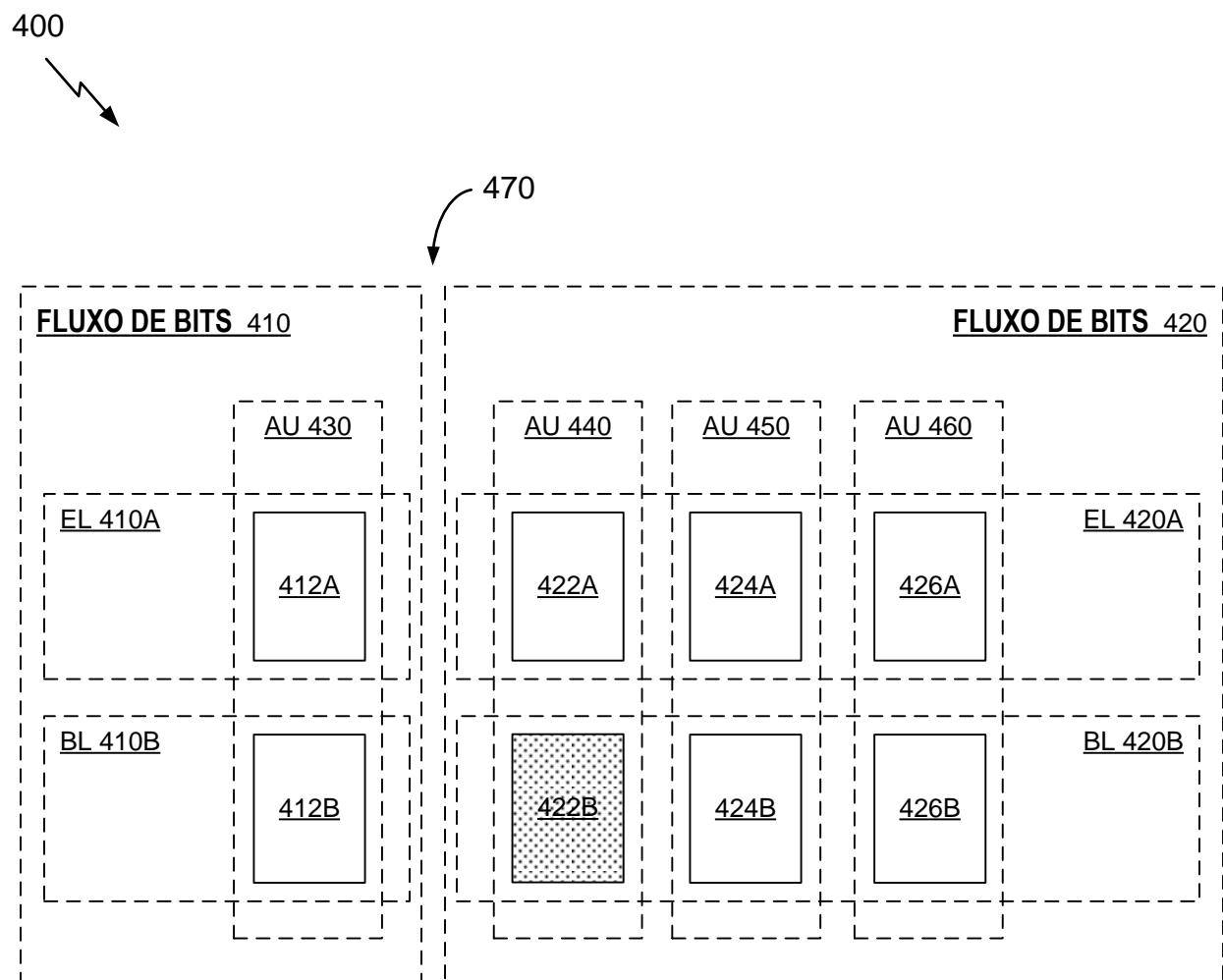
FIG. 2A

**FIG. 2B**





**FIG. 3B**

**FIG. 4**

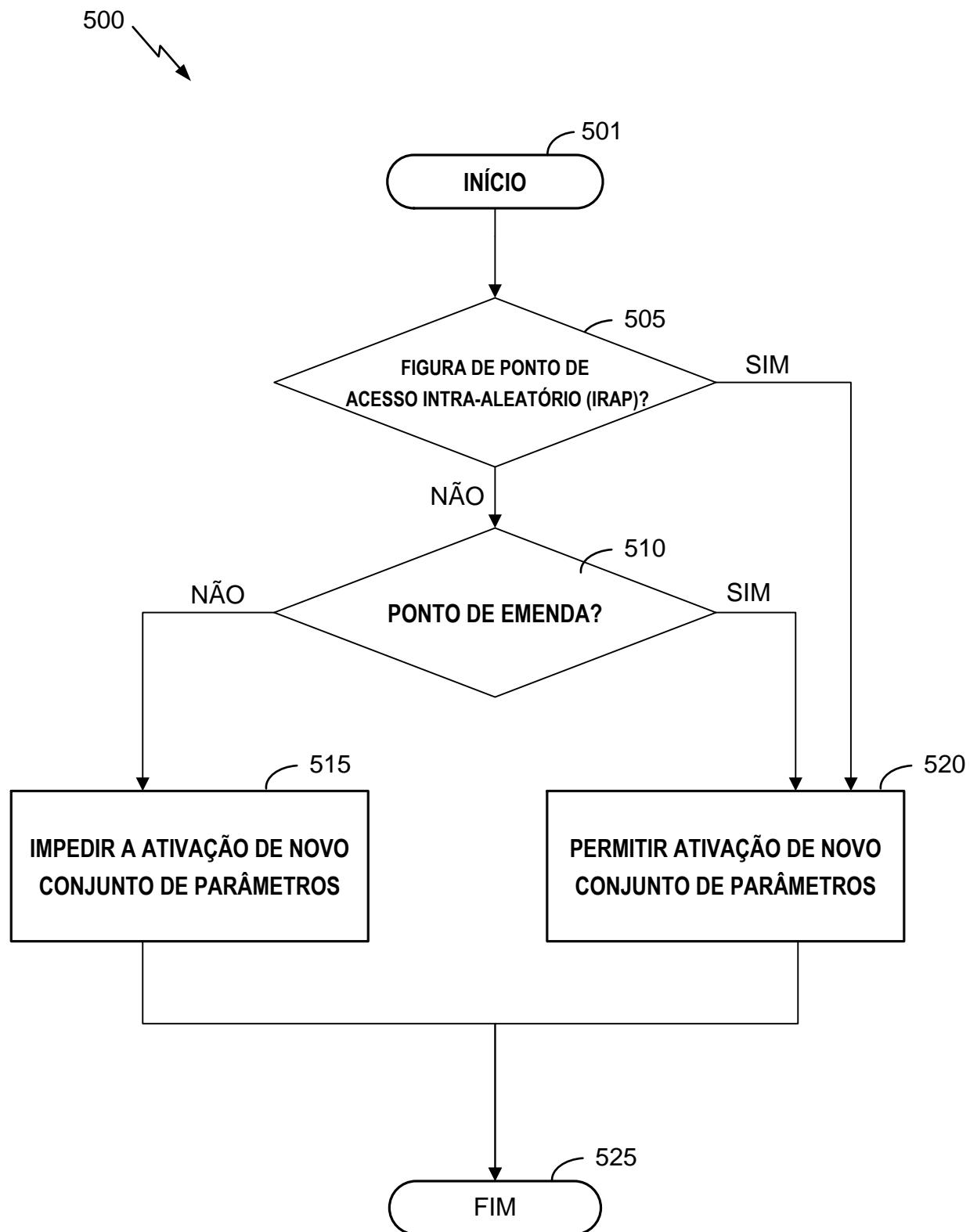


FIG. 5