



República Federativa do Brasil

Ministério do Desenvolvimento, Indústria,
Comércio e Serviços

Instituto Nacional da Propriedade Industrial

(11) BR 112016029611-7 B1

(22) Data do Depósito: 17/06/2015

(45) Data de Concessão: 21/11/2023

(54) Título: APARELHO E MÉTODO PARA CODIFICAR INFORMAÇÕES DE VÍDEO EM CODIFICAÇÃO DE VÍDEO DE ALTA EFICIÊNCIA, E, MEMÓRIA LEGÍVEL POR COMPUTADOR

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

(30) Prioridade Unionista: 16/06/2015 US 14/741,253; 20/06/2014 US 62/015,346.

(73) Titular(es): QUALCOMM INCORPORATED.

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

(86) Pedido PCT: PCT US2015036279 de 17/06/2015

(87) Publicação PCT: WO 2015/195832 de 23/12/2015

(85) Data do Início da Fase Nacional: 16/12/2016

(57) Resumo: REDEFINIÇÃO DE CONTAGEM DE ORDEM DE FIGURAÇÃO TOTAL PARA CODECS DE MÚLTIPLAS CAMADAS. Trata-se de um aparelho para codificar informações de vídeo de acordo com determinados aspectos que inclui uma unidade de memória configurada para armazenar informações de vídeo associadas a uma pluralidade de camadas e um processador. O processador é configurado para obter informações associadas a uma unidade de acesso (AU) atual a ser codificada, em que a AU atual contém figurações de uma ou mais camadas dentre a pluralidade de camadas. O processador também é configurado para redefinir uma contagem de ordem de figuração (POC) de uma camada incluída na AU atual por meio de (1) redefinição apenas de um bit mais significativo (MSB) da POC ou (2) redefinição tanto do MSB da POC quanto de um menos significativo (LSB) da POC. O processador é adicionalmente configurado para, para figurações em uma ou mais AUs subsequentes à AU atual em ordem de decodificação: definir um valor de um primeiro sinalizador indicativo da possibilidade de uma redefinição da POC ser uma redefinição total.

**"APARELHO E MÉTODO PARA CODIFICAR INFORMAÇÕES DE VÍDEO EM
CODIFICAÇÃO DE VÍDEO DE ALTA EFICIÊNCIA, E, MEMÓRIA LEGÍVEL
POR COMPUTADOR"**

CAMPO DA TÉCNICA

[0001] Esta revelação refere-se ao campo de codificação e compressão de vídeo, que inclui tanto codificação de vídeo de camada única quanto codificação de vídeo com múltiplas camadas. A codificação de vídeo com múltiplas camadas pode incluir codificação de vídeo escalonável, codificação de vídeo com múltiplas camadas, codificação de vídeo tridimensional (3D), etc.

ANTECEDENTES

[0002] Os recursos de vídeo digitais podem ser incorporados em uma ampla faixa de dispositivos, incluindo televisões digitais, sistemas de difusão direta digital, sistemas de difusão sem fio, assistentes pessoais digitais (PDAs), computadores do tipo laptop ou de mesa, câmeras digitais, dispositivos de gravação digital, reprodutores de mídia digital, dispositivos de jogos em vídeo, consoles de videogame, radiotelefone celular ou a satélite, dispositivos de vídeo para teleconferência, e semelhantes. Os dispositivos de vídeo digital implantam técnicas de compactação de vídeo, como aquelas descritas nos padrões definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificação de Vídeo Avançada (AVC), o padrão de Codificação de Vídeo de Alta Eficiência (HEVC) e extensões de tais padrões. Os dispositivos de vídeo podem transmitir, receber, codificar, decodificar e/ou armazenar informações de vídeo digitais mais eficientemente 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, uma porção de um quadro de vídeo, etc.) pode ser particionada em blocos de vídeo, que também podem ser denominados de blocos de 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 a 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 às amostras de referência em blocos vizinhos na mesma predição de imagem ou 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 e 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 compressão 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 submetidos a uma varredura a fim de produzir um vetor monodimensional de coeficientes de transformada, e a

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

SUMÁRIO

[0005] Os sistemas, métodos e dispositivos desta revelação têm, cada um, diversos aspectos inovadores, nenhum dos quais é responsável exclusivamente pelos atributos desejáveis revelados no presente documento. Os detalhes de um ou mais exemplos são estabelecidos nos desenhos anexos e na descrição abaixo, que não estão destinados a limitar o escopo total dos conceitos inventivos descritos no presente documento. Outros recursos, objetivos e vantagens serão evidentes a partir da descrição e dos desenhos, e a partir das reivindicações.

[0006] A codificação de vídeo escalonável (SVC) se refere à codificação de vídeo na qual uma camada-base (BL), ocasionalmente, denominada de camada de referência (RL), e uma ou mais camadas de aprimoramento escalonável (ELs) são usadas. Na SVC, a camada-base pode portar dados de vídeo com um nível-base de qualidade. As uma ou mais camadas de aprimoramento podem portar dados de vídeo adicionais para suportar, por exemplo, níveis de razão espacial, temporal e/ou de sinal/ruído (SNR) superiores. As camadas de melhoria podem ser definidas em relação a uma camada previamente codificada. Por exemplo, uma camada de fundo pode servir como uma BL, ao passo que uma camada de topo pode servir como uma EL. As camadas intermediárias podem servir tanto como ELs como RLs, ou ambos. Por exemplo, uma camada intermediária (por exemplo, uma camada que não é a camada mais inferior tampouco a camada mais superior) pode ser uma EL para as camadas abaixo da camada intermediária, tais como, a camada-base ou quaisquer camadas de aprimoramento intervenientes e, ao mesmo tempo, serve como uma RL para uma ou mais camadas de aprimoramento acima da camada

intermediária. De modo semelhante, na extensão com Múltiplas Vistas ou 3D do padrão HEVC, há múltiplas vistas, e as informações de uma vista podem ser utilizadas para codificar (code) (por exemplo, codificar (encode) ou decodificar) as informações de outra vista (por exemplo, estimativa de movimento, predição de vetor de movimento e/ou outras redundâncias).

[0007] Um aparelho para codificar informações de vídeo de acordo com certos aspectos, o qual inclui uma unidade de memória e um processador. A unidade de memória é configurada para armazenar informações de vídeo associadas a uma pluralidade de camadas. O processador é configurado para obter informações associadas a uma unidade de acesso (AU) atual a ser codificada, em que a AU atual contém imagens de uma ou mais camadas dentre a pluralidade de camadas. O processador é adicionalmente configurado para determinar se a AU atual inclui uma primeira camada que contém uma imagem de ponto de acesso intra-aleatório (IRAP). O processo é adicionalmente configurado para redefinir uma contagem de ordem de imagem (POC) da segunda camada na AU atual, em resposta à determinação de que a AU atual inclui (1) uma primeira camada que contém uma imagem de IRAP e (2) uma segunda camada que não contém nenhuma imagem ou que contém uma imagem descartável.

[0008] Um aparelho para codificar informações de vídeo de acordo com certos aspectos, o qual inclui uma unidade de memória e um processador. A unidade de memória é configurada para armazenar informações de vídeo associadas a uma pluralidade de camadas. O processador é configurado para obter informações associadas a uma unidade de acesso (AU) atual a ser codificada, em que a AU atual contém imagens de uma ou mais camadas dentre a pluralidade de camadas. O processador também é configurado para redefinir uma contagem

de ordem de imagem (POC) de uma camada incluída na AU atual por meio de (1) redefinição apenas de um bit mais significativo (MSB) da POC ou (2) redefinição tanto do MSB da POC quanto de um menos significativo (LSB) da POC. O processador é adicionalmente configurado para, para imagens em uma ou mais AUs subsequentes à AU atual em ordem de decodificação: definir um valor de um primeiro sinalizador indicativo da possibilidade de uma redefinição da POC ser uma redefinição total.

BREVE DESCRIÇÃO DOS DESENHOS

[0009] A Figura 1A é um diagrama de blocos que ilustra um sistema exemplificativo de codificação e decodificação de vídeo que pode utilizar técnicas de acordo com aspectos descritos nesta revelação.

[0010] A Figura 1B é um diagrama de blocos que ilustra outro sistema exemplificativo de codificação e decodificação de vídeo que pode realizar técnicas de acordo com aspectos descritos nesta revelação.

[0011] 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 aspectos descritos nesta revelação.

[0012] A Figura 2B é um diagrama de blocos que ilustra um exemplo de um codificador de vídeo que pode implantar técnicas de acordo com aspectos descritos nesta revelação.

[0013] A Figura 3A é um diagrama de blocos que ilustra um exemplo de um decodificador de vídeo que pode implantar técnicas de acordo com aspectos descritos nesta revelação.

[0014] A Figura 3B é um diagrama de blocos que ilustra um exemplo de um decodificador de vídeo que pode implantar técnicas de acordo com aspectos descritos nesta

revelação.

[0015] A Figura 4 é um diagrama de blocos que ilustra uma conimagem exemplificativa de imagens em camadas diferentes.

[0016] A Figura 5 é um diagrama de blocos que ilustra uma conimagem exemplificativa de imagens em camadas diferentes.

[0017] A Figura 6 é um diagrama de blocos que ilustra uma conimagem exemplificativa de imagens em camadas diferentes.

[0018] A Figura 7 é um diagrama de blocos que ilustra uma conimagem exemplificativa de imagens em camadas diferentes.

[0019] A Figura 8 é um fluxograma que ilustra um método para codificar informações de vídeo de acordo com um ou mais aspectos da presente revelação.

[0020] A Figura 9 é um fluxograma que ilustra um método para codificar informações de vídeo de acordo com um ou mais aspectos da presente revelação.

DESCRIÇÃO DETALHADA

[0021] Em geral, esta revelação se refere a codificação em camada única, assim como predição intercamada para codificação de vídeo escalonável no contexto de codecs de vídeo avançado, como HEVC (Codificação de Vídeo de Alta Eficiência). Mais especificamente, a presente revelação se refere a sistemas e métodos para redefinição de contagem de ordem de imagem (POC) para codecs de múltiplas camadas.

[0022] Na descrição abaixo, as técnicas de H.264/Codificação de Vídeo Avançada (AVC) relacionadas a certas modalidades são descritas; o padrão de HEVC e as técnicas relacionadas também são discutidos. Embora determinadas modalidades sejam descritas no presente documento no contexto dos padrões HEVC e/ou H.264, uma pessoa

da habilidade comum na técnica pode observar que os sistemas e os 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 dentre os seguintes padrões: O Setor de Padronização de Telecomunicações (ITU-T) da União Internacional de Telecomunicações (ITU) H.261, o Visual do Grupo de Especialistas em Imagem em Movimento (MPEG) 1 (MPEG-1) da Organização Internacional de Normalização (ISO) e da Comissão Eletrotécnica Internacional (IEC) (ISO/IEC), o Visual ITU-T H.262 ou ISO/IEC MPEG-2, ITU-T H.263, o Visual ISO/IEC MPEG-4 e ITU-T H.264 (também conhecido como ISO/IEC MPEG-4 AVC), que inclui suas extensões de Codificação de Vídeo Escalonável (SVC) e de Codificação de Vídeo com Múltiplas Camadas (MVC).

[0023] A HEVC segue, geralmente, o enquadramento de padrões de codificação de vídeo anteriores em muitos aspectos. A unidade de predição em HEVC é diferente das unidades de predição (por exemplo, macrobloco) em certos padrões de codificação de vídeo anteriores. De fato, não há conceito de macrobloco em HEVC, conforme entendido em determinados padrões de codificação de vídeo anteriores. Um macrobloco é substituído por uma estrutura hierárquica com base em um esquema de árvore quadrática, que pode fornecer alta flexibilidade, dentre outros possíveis benefícios. Por exemplo, dentro do esquema de HEVC, são definidos três tipos de blocos, a Unidade de Codificação (CU), a Unidade de Predição (PU) e a Unidade de Transformada (TU). A CU pode se referir à unidade básica de divisão de região. A CU pode ser considerada análoga ao conceito de macrobloco, porém, a HEVC não restringe o tamanho máximo das CUs e pode permitir a divisão recursiva em quatro CUs de tamanho igual a fim de

melhorar a capacidade de adaptação ao 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 padrões de imagem irregulares de maneira eficaz. A TU pode ser considerada a unidade básica de transformada. A TU pode ser definida independentemente da PU; no entanto, o tamanho de uma TU pode se limitar ao tamanho da CU à qual a TU pertence. Essa separação da estrutura de bloco em três conceitos diferentes pode permitir que cada unidade seja otimizada de acordo com a função respectiva da unidade, o que pode resultar em eficiência de codificação aprimorada.

[0024] Apenas para propósitos de ilustração, determinadas modalidades reveladas no presente documento são descritas com exemplos que incluem apenas duas camadas (por exemplo, uma camada inferior, tal como, a camada-base e uma camada superior, tal como, a camada de aprimoramento) de dados de vídeo. Uma "camada" de dados de vídeo pode se referir, de modo geral, a uma sequência de imagens que têm pelo menos uma característica comum, como uma vista, uma taxa de quadro, uma resolução ou similares. Por exemplo, uma camada pode incluir dados de vídeo associados a uma vista particular (por exemplo, perspectiva) de dados de vídeo de múltiplas vistas. Como outro exemplo, uma camada pode incluir dados de vídeo associada a uma camada particular de dados de vídeo escalonáveis. Desse modo, esta revelação pode se referir de modo intercambiável a uma camada e uma vista de dados de vídeo. Ou seja, uma vista de dados de vídeo pode ser referida como uma camada de dados de vídeo, e uma camada de dados de vídeo pode ser referida como uma vista de dados de vídeo. Além disso, um codec com múltiplas camadas (também referido como um codificador de vídeo com múltiplas camadas ou codificador e decodificador com múltiplas camadas) pode

se referir, de modo conjunto, a um codec com múltiplas vistas ou um codec escalonável (por exemplo, um codec configurado para codificar e/ou decodificar dados de vídeo que usam MV-HEVC, 3D-HEVC, SHVC, ou outra técnica de codificação com múltiplas camadas). A codificação (encoding) de vídeo e a decodificação de vídeo podem ser referidas, ambas, de modo geral, como codificação de vídeo. Deve-se entender que tais exemplos podem ser aplicáveis às configurações que incluem múltiplas camadas-base e/ou de aprimoramento. Além disso, a fim de facilitar a exemplificação, a revelação a seguir inclui os termos "quadros" ou "blocos" com referência a determinadas modalidades. Entretanto, esses termos não devem ser limitativos. Por exemplo, as técnicas descritas abaixo podem ser usadas com quaisquer unidades de vídeo, tais como, blocos (por exemplo, CU, PU, TU, macroblocos, etc.), fatias, quadros, etc.

PADRÕES DE CODIFICAÇÃO DE VÍDEO

[0025] 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 em amostras dispostos em linhas horizontais e verticais. O número de pixels em uma única imagem é tipicamente de dezenas de milhares. Cada pixel contém tipicamente informações de luminância e de croma. Sem compactação, a quantidade absoluta de informações a ser transmitida de um codificador de imagem para um decodificador de imagem tornaria a transmissão de imagem em tempo real impossível. A fim de reduzir a quantidade de informações a ser transmitida, foram desenvolvidos vários métodos de compressão diferentes, tais como, os padrões JPEG, MPEG e H.263.

[0026] Os padrões de codificação de vídeo incluem ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 ou

ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (também chamado de ISO/IEC MPEG-4 AVC), incluindo suas extensões de SVC e de MVC.

[0027] Além disso, um novo padrão de codificação de vídeo, a saber, Codificação de Vídeo de Alta Eficiência (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 de HEVC 10 é o documento JCTVC-L1003, Bross et al, "High Efficiency Video codificação (HEVC) Text Specification Draft 10", Equipe Colaborativa Conjunta em Codificação de Vídeo (JCT-VC - Joint Collaborative Team on Video Coding) da ITU-T SGI 6 WP3 e ISO/IEC JTC1/SC29/WG11, 12ª reunião: Genebra, Suíça, 14 de janeiro de 2013 a 23 de janeiro de 2013. A extensão com múltiplas vistas para HEVC, a saber, MV-HEVC e a extensão escalonável para HEVC, denominada de SHVC, também estão em desenvolvimento pela JCT-3V (Equipe Colaborativa Conjunta em Desenvolvimento de Extensão de Vídeo 3D da ITU-T/ISO/IEC - Equipe Colaborativa Conjunta em ITU-T/ISO/IEC de Desenvolvimento da Extensão de Codificação de Vídeo 3D) e pela JCT-VC, respectivamente.

VISÃO GERAL

[0028] O elemento de sintaxe poc_reset_idc pode indicar se a POC deveria ser redefinida para uma imagem. O elemento de sintaxe poc_reset_idc pode indicar se o bit mais significativo (MSB) da POC deve ser redefinido ou tanto o MSB quanto o bit menos significativo (LSB) da POC devem ser redefinidos, ou se nenhum deve ser redefinido. Por exemplo, o valor de 0 para o poc_reset_idc indica que a POC não é redefinida. O valor de 1 para poc_reset_idc indica que o MSB de POC deve ser redefinido. O valor de 2 para poc_reset_idc

indica que tanto o MSB de POC quanto o LSB devem ser redefinidos. O valor de 3 para o poc_reset_idc indica que a redefinição foi indicada para uma imagem anterior. Por exemplo, o valor de poc_reset_idc para a imagem anterior foi tanto 1 quanto 2. O valor de 3 para poc_reset_idc pode ser usado de modo que, quando a imagem em que a POC deve ser redefinida for perdida (por exemplo, durante o processo de decodificação), a POC possa ser adequadamente redefinida em imagens subsequentes.

[0029] O full_poc_reset_flag pode indicar se a redefinição para a imagem anterior foi apenas para o MSB de POC, ou tanto para o MSB de POC quanto o LSB. Por exemplo, o valor de 0 para full_poc_reset_flag indica que apenas o MSB deve ser redefinido. O valor de 1 para o full_poc_reset_flag indica que tanto o MSB quanto o LSB devem ser redefinidos. O sinalizador full_poc_reset_flag pode ser usado em conexão com poc_reset_idc. Por exemplo, quando o valor de poc_reset_idc for 3, o full_poc_reset_flag pode indicar se a redefinição de POC para a imagem anterior for apenas para o MSB ou tanto para o MSB quanto o LSB.

[0030] Em versões anteriores de SHVC e MV-HEVC (por exemplo, SHVC Working Draft 6, MV-HEVC Working Draft 8, etc.), determinadas restrições ou restrições se aplicam, por exemplo, em relação ao poc_reset_idc. Entretanto, essas restrições não redefinem adequadamente a POC quando uma imagem não está presente ou quando uma imagem é descartável. Além disso, nas versões anteriores de SHVC e MV-HEVC, não há restrições quanto ao valor de full_poc_reset_flag de uma imagem com base em poc_reset_idc da AU de redefinição de POC no mesmo período de redefinição de POC. Um valor incorreto de full_poc_reset_flag pode resultar em o mecanismo de redefinição de POC não funcionar adequadamente.

[0031] A fim de solucionar esses e outros

desafios, as técnicas de acordo com determinados aspectos redefinem a POC quando uma imagem não está presente (por exemplo, em falta ou ausente) ou quando uma imagem é descartável. As técnicas também impõem uma restrição quanto ao valor de `full_poc_reset_flag` de uma imagem com base no valor de `poc_reset_idc`. Dessa forma, as técnicas podem assegurar que a POC seja redefinida corretamente.

SISTEMA DE CODIFICAÇÃO DE VÍDEO

[0032] Diversos aspectos dos sistemas, aparelhos e métodos inovadores são descritos mais completamente a doravante no presente documento com referência aos desenhos anexos. Entretanto, a presente revelação pode ser incorporada de várias maneiras diferentes e não deve ser interpretada como limitada a qualquer estrutura ou função específica apresentada ao longo de toda a presente revelação. Em vez disso, esses aspectos são fornecidos de modo que a presente revelação seja minuciosa e completa, e transmita plenamente o escopo da revelação para as pessoas versadas na técnica. Com base nos ensinamentos no presente documento, uma pessoa versada na técnica deve observar que o escopo da revelação está destinado a cobrir qualquer aspecto dos sistemas, aparelhos e métodos inovadores revelados no presente documento, sejam os mesmos implantados independentemente ou em combinação com qualquer outro aspecto da revelação. Por exemplo, um aparelho pode ser implantado ou um método pode ser praticado com o uso de vários dos aspectos estabelecidos no presente documento. Além disso, o escopo da presente revelação está destinado a cobrir tal aparelho ou método que é praticado com o uso de outra estrutura, funcionalidade ou estrutura e funcionalidade adicionalmente, ou que não sejam os vários aspectos da presente revelação estabelecidos no presente documento. Deve ser entendido que qualquer aspecto revelado

no presente documento pode ser incorporado por um ou mais elementos de uma reivindicação.

[0033] Embora sejam descritos os aspectos particulares no presente documento, muitas variações e permutações desses aspectos são abrangidas pelo escopo da revelação. Embora alguns benefícios e vantagens dos aspectos preferenciais sejam mencionados, o escopo da revelação não está destinado a ser limitado a benefícios, usos ou objetivos particulares. Em vez disso, os aspectos da revelação destinam-se a ser amplamente aplicáveis a diferentes tecnologias sem fio, conimagens 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 revelação em vez de limitantes, sendo que o escopo da revelação é definido pelas reivindicações anexas e equivalentes das mesmas.

[0034] Os desenhos anexos ilustram exemplos. Os elementos indicados por numerais de referência nos desenhos anexos correspondem aos elementos indicados por numerais de referência similares na descrição a seguir. Na presente revelação, os elementos que têm nomes que começam com palavras ordinais (por exemplo, "primeiro(a)", "segundo(a)", "terceiro(a)" e assim por diante) não implicam necessariamente que os elementos têm uma ordem particular. De preferência, tais palavras ordinais são suadas apenas para diferenciar os elementos de tipos iguais ou semelhantes.

[0035] A Figura 1A é um diagrama de blocos que ilustra um sistema de codificação de vídeo 10 exemplificativo que pode utilizar as técnicas de acordo com os aspectos descritos nesta revelação. Conforme usado descrito no presente documento, o termo "codificador de vídeo" se refere, de modo geral, tanto a codificadores de vídeo quanto a

decodificadores de vídeo. Nesta revelação, os termos "codificação de vídeo" ou "codificação" podem se referir, de modo geral, 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 se estender a outros dispositivos relacionados, tais como transcodificadores (por exemplo, dispositivos que podem decodificar um fluxo de bits e recodificar outro fluxo de bits) e Middleboxes (por exemplo, dispositivos que podem modificar, transformar e/ou, de outro modo, manipular um fluxo de bits).

[0036] Conforme mostrado na Figura 1A, o sistema de codificação de vídeo 10 inclui um dispositivo de origem 12 que gera dados de vídeo codificados a serem decodificados posteriormente por um dispositivo de destino 14. No exemplo da Figura 1A, o dispositivo de origem 12 e o dispositivo de destino 14 constituem dispositivos separados. Entretanto, verifica-se que os dispositivos de origem e de destino 12, 14 podem estar no mesmo dispositivo ou podem ser parte do mesmo, conforme mostrado no exemplo da Figura 1B.

[0037] Com referência, mais uma vez, à Figura 1A, o dispositivo de origem 12 e o dispositivo de destino 14 podem compreender, respectivamente, qualquer um dentre uma ampla faixa de dispositivos, incluindo computadores de mesa, computadores do tipo notebook (por exemplo, computadores do tipo laptop), computadores do tipo tablet, decodificadores de sinais, aparelhos de telefone, tais como, os então chamados telefones "inteligentes", os então chamados dispositivos do tipo pad "inteligentes", televisões, câmeras, dispositivos de exibição, reprodutores de mídia digital, consoles de videogame, dispositivo de transmissão contínua de vídeo, ou semelhantes. Em alguns casos, o dispositivo de origem 12 e o dispositivo de destino 14 podem

ser equipados para comunicação sem fio.

[0038] O dispositivo de destino 14 pode receber, por meio do enlace 16, os dados de vídeo codificados a serem decodificados. O enlace 16 pode compreender qualquer tipo de meio ou dispositivo com capacidade de mover os dados de vídeo codificados do dispositivo de origem 12 para o dispositivo de destino 14. No exemplo da Figura 1A, o enlace 16 pode compreender um meio de comunicação para possibilitar que o dispositivo de origem 12 transmita os dados de vídeo codificado ao dispositivo 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. O meio de comunicação pode formar parte de uma rede com base em pacote, como uma rede de área local, uma rede de área ampla ou uma rede global, como a Internet. O meio de comunicação pode incluir roteadores, comutadores, estações-base, ou qualquer outro equipamento que possa ser útil que para facilitar a comunicação do dispositivo de origem 12 com o dispositivo de destino 14.

[0039] Alternativamente, dados codificados podem ser emitidos de uma interface de emissão 22 para um dispositivo de armazenamento opcional 31. De modo similar, os dados codificados podem ser acessados a partir do dispositivo de armazenamento 31 por uma interface de entrada 28, por exemplo, do dispositivo de destino 14. O dispositivo de armazenamento 31 pode incluir qualquer um dentre uma variedade de meios de armazenamento de dados distribuídos ou acessados localmente como um disco rígido, memória flash, 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 manter o vídeo codificado gerado pelo dispositivo de origem 12. O dispositivo de destino 14 pode acessar dados de vídeo armazenados a partir do dispositivo de armazenamento por meio de transmissão 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 dispositivo de destino 14. Os servidores de arquivo exemplificativos incluem um servidor da web (por exemplo, para um site da web), um servidor de Protocolo de Transferência de Arquivos (FTP), dispositivos de armazenamento anexado à rede (NAS) ou uma unidade de disco local. O dispositivo de destino 14 pode acessar os dados de vídeo codificado através de qualquer conexão de dados padrão, incluindo uma conexão à Internet. Isso pode incluir um canal sem fio (por exemplo, uma conexão de rede de área local sem fio (WLAN)), uma conexão cabeada (por exemplo, uma linha de assinatura digital (DSL), um modem a cabo, etc.), ou uma combinação de ambos que seja adequada para acessar dados de vídeo codificados armazenados em um servidor de arquivos. A transmissão de dados de vídeo codificados do dispositivo 31 de armazenamento pode ser uma transmissão contínua, uma transmissão de transferência por download ou uma combinação das mesmas.

[0040] As técnicas da presente revelação não estão necessariamente limitadas às aplicações ou definições sem fio. As técnicas podem ser aplicadas à codificação de vídeo em apoio a qualquer uma dentre uma variedade de aplicações de multimídia, tais como, difusões de televisão remotas, transmissões de televisão a cabo, transmissões de

televisão a satélite, transmissões contínuas de vídeo, por exemplo, por meio da Internet (por exemplo, transmissão contínua adaptativa dinâmica através de Protocolo de Transferência de Hipertexto (HTTP), etc.), codificação de vídeo digital para armazenamento em um meio de armazenamento de dados, decodificação de vídeo digital armazenada em um meio de armazenamento de dados, ou outras aplicações. Em alguns exemplos, o sistema de codificação de vídeo 10 pode ser configurado para suportar transmissão de vídeo unidirecional ou bidirecional a fim de suportar aplicações, tais como, transmissão contínua de vídeo, reprodução de vídeo, difusão de vídeo e/ou videotelefonia.

[0041] No exemplo da Figura 1A, o dispositivo de origem 12 inclui uma fonte de vídeo 18, um codificador (encoder) de vídeo 20 e a interface de saída 22. Em alguns casos, a interface de saída 22 pode incluir um modulador/demodulador (modem) e/ou a transmissor. No dispositivo de origem 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 de vídeo que contém vídeo capturado anteriormente, uma interface de alimentação de vídeo para receber vídeo de um fornecedor de conteúdo de vídeo e/ou um sistema de computação gráfica para gerar dados de computação gráfica, 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 dispositivo de fonte 12 e o dispositivo 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 revelação podem ser aplicáveis à codificação de vídeo em geral e podem ser aplicadas às aplicações com fio e/ou sem fio.

[0042] O vídeo capturado, pré-capturado ou

gerado por computador pode ser codificado pelo codificador de vídeo 20. Os dados de vídeo codificados podem ser transmitidos para o dispositivo de destino 14 através da interface de emissão 22 do dispositivo de origem 12. Os dados de vídeo codificados também podem (ou, alternativamente) ser armazenados no dispositivo de armazenamento 31 para acesso posterior pelo dispositivo de destino 14 ou outros dispositivos, para decodificação e/ou reprodução. O codificador de vídeo 20 ilustrado nas Figuras 1A e 1B podem 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.

[0043] No exemplo da Figura 1A, o dispositivo 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 dispositivo de destino 14 pode receber os dados de vídeo codificados pelo enlace 16 e/ou a partir do dispositivo de armazenamento 31. Os dados de vídeo codificados comunicados pelo enlace 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, como o decodificador de vídeo 30, na decodificação dos dados de vídeo. Tais elementos de sintaxe podem estar incluídos com os dados de vídeo codificados transmitidos em um meio de comunicação, armazenados em um meio de armazenamento ou armazenados 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.

[0044] O dispositivo de exibição 32 pode ser integrado ou externo ao dispositivo de destino 14. Em alguns exemplos, o dispositivo de destino 14 pode incluir um dispositivo de exibição integrado e também ser configurado para fazer interface com um dispositivo de exibição externo. Em outros exemplos, o dispositivo de destino 14 pode ser um dispositivo de exibição. Em general, o dispositivo de exibição 32 exibe os dados de vídeo decodificados a um usuário e pode compreender qualquer um dentre uma variedade de dispositivos de exibição, tais como, um visor de cristal líquido (LCD), um visor de plasma, um visor de diodo de emissão de luz orgânico (OLED) ou outro tipo de dispositivo de exibição.

[0045] Em aspectos relacionados, a Figura 1B mostra um sistema de codificação e decodificação de vídeo exemplificativo 10' em que os dispositivos de origem e de destino 12, 14 estão em um dispositivo 11, ou fazem parte do mesmo. O dispositivo 11 pode ser um aparelho de telefone, tal como um telefone "inteligente" ou semelhantes. O dispositivo 11 pode incluir um dispositivo de controlador/processador opcional 13 em comunicação de operação com os dispositivos de origem e de destino 12, 14. O sistema 10' da Figura 1B e componentes do mesmo são, de outro modo, similares ao sistema 10 da Figura 1A e aos componentes do mesmo.

[0046] O codificador de vídeo 20 e o decodificador de vídeo 30 podem operar de acordo com um padrão de codificação de vídeo, como 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, AVC, ou extensões de tais padrões. Os conjuntos de procedimentos desta revelação, entretanto, não se limitam a qualquer padrão de codificação particular. Outros exemplos padrões de compressão de vídeo incluem MPEG-2 e ITU-T H.263.

[0047] Embora não sejam mostrados nos exemplos das Figuras 1A e 1B, 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 gerenciar a codificação tanto do áudio quanto do vídeo em um fluxo de dados comum ou fluxos de dados separado. Caso seja aplicável, em alguns exemplos, as unidades MUX-DEMUX podem se conformar ao protocolo multiplexador ITU H.223, ou outros protocolos como o protocolo de datagrama de usuário (UDP).

[0048] Cada um dentre o codificador de vídeo 20 e o decodificador de vídeo 30 pode ser implantado como qualquer um dentre uma variedade de conjuntos de circuitos de codificador adequado, tal como um ou mais microprocessadores, processadores de sinal digital (DSPs), circuitos integrados específicos para aplicação (ASICs), matrizes de porta programável em 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 revelação. Cada um dentre o codificador de vídeo 20 e o decodificador de vídeo 30 pode ser incluído em um ou mais codificadores ou decodificadores, em que qualquer um dos quais pode ser integrado como parte de um

codificador/decodificador combinado (CODEC) em um respectivo dispositivo.

PROCESSO DE CODIFICAÇÃO DE VÍDEO

[0049] 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 faz parte de um vídeo. Em alguns exemplos, uma imagem pode ser denominada de "quadro" de vídeo. Quando o codificador de vídeo 20 codificar 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.

[0050] 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 realizar 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 (SPSs), conjuntos de parâmetros de imagem (PPSs), conjuntos de parâmetros de adaptação (APSS) e outras estruturas de sintaxe. Um SPS pode conter parâmetros aplicáveis a zero ou mais sequências de imagens. Um PPS pode conter parâmetros aplicáveis a zero ou mais imagens. Um APS pode conter parâmetros aplicáveis a zero ou mais imagens. Os parâmetros em uma APS podem ser parâmetros que podem estar mais propensos a mudar do que os parâmetros em uma PPS.

[0051] 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 de árvore. Em alguns exemplos, um bloco de árvore pode ser denominado de unidade de codificação maior (LCU). Os blocos de árvores de HEVC podem ser amplamente análogos aos macroblocos de padrões anteriores, tais como, H.264/AVC. Entretanto, um bloco de 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 particionamento de QuadTree para particionar os blocos de vídeo de blocos de árvore em blocos de vídeo associados a CUs, por isso o nome "blocos de árvore".

[0052] 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 exemplos, uma fatia compreende um número inteiro de blocos de árvores. Em outros exemplos, um limite de uma fatia pode estar dentro de um bloco de árvore.

[0053] Como parte da realização de uma operação de codificação em uma imagem, o codificador de vídeo 20 pode realizar as operações de codificação em cada fatia da imagem. Quando o codificador de vídeo 20 realizar 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 "fatia codificada".

[0054] A fim de gerar uma fatia codificada, o codificador de vídeo 20 pode realizar operações de codificação em cada bloco de árvore em uma fatia. Quando o codificador de vídeo 20 realizar uma operação de codificação em um bloco de árvore, o codificador de vídeo 20 pode gerar um bloco de árvore codificado. O bloco de árvore codificado pode compreender dados que representam uma versão codificada

do bloco de árvore.

[0055] Quando o codificador de vídeo 20 gerar uma fatia codificada, o codificador de vídeo 20 pode realizar as operações de codificação (por exemplo, codificar) nos blocos de árvores na fatia, de acordo com uma ordem de varredura por rastreo. Por exemplo, o codificador de vídeo 20 pode codificar os blocos de árvores da fatia em que uma ordem que procede da esquerda para a direita ao longo de uma fileira mais superior de blocos de árvores na fatia, em seguida, da esquerda para a direita ao longo de uma próxima fileira inferior de blocos de árvores, e assim por diante até que o codificador de vídeo 20 tenha codificado cada um dos blocos de árvores na fatia.

[0056] Como resultado da codificação dos blocos de árvores de acordo com a ordem de varredura por rastreo, os blocos de árvores acima e à esquerda de um determinado bloco de árvore podem ter sido codificados, porém, os blocos de árvores abaixo e à direita do determinado bloco de árvore ainda não foram codificados. Consequentemente, o codificador de vídeo 20 pode ter capacidade para acessar informações geradas codificando-se os blocos de árvores acima e à esquerda do determinado bloco de árvore durante a codificação do determinado bloco de árvore. Entretanto, o codificador de vídeo 20 pode não ter capacidade para acessar as informações geradas codificando-se blocos de árvores abaixo e à direita do determinado bloco de árvore durante a codificação do determinado bloco de árvore.

[0057] A fim de gerar um bloco de árvore codificado, o codificador de vídeo 20 pode realizar recursivamente a partição de árvore quadrática no bloco de vídeo do bloco de árvore para dividir o bloco de vídeo em blocos de vídeo progressivamente menores. Cada um dos blocos de vídeo menores pode estar associado a uma CU diferente.

Por exemplo, o codificador de vídeo 20 pode particionar o bloco de vídeo de um bloco de árvore em quatro sub-blocos com tamanhos iguais, particionar um ou mais dentre os sub-blocos em quatro sub-blocos com tamanhos iguais, e assim por diante. Uma CU particionada pode ser uma CU cujo bloco de vídeo é particionado em blocos de vídeo associados a outras CUs. Uma CU não particionada pode ser uma CU cujo bloco de vídeo é particionado em blocos de vídeo associados a outras CUs.

[0058] Um ou mais elementos de sintaxe no fluxo de bits pode indicar um número máximo de vezes que o codificador de vídeo 20 pode particionar o bloco de vídeo de um bloco de árvore. Um bloco de vídeo de um CU pode ter o formato quadrado. O tamanho do bloco de vídeo de uma CU (por exemplo, o tamanho da CU) pode estar em uma faixa de pixels 8x8 até o tamanho de um bloco de vídeo de um bloco de árvore (por exemplo, o tamanho do bloco de árvore) com um máximo de pixels 64x64 ou mais.

[0059] O codificador de vídeo 20 pode realizar as operações de codificação (por exemplo, codificar) em cada CU de um bloco de árvore, de acordo com uma ordem de varredura-z. Em outras palavras, o codificador de vídeo 20 pode codificar uma CU superior à esquerda, uma CU superior à direita, uma CU de fundo à esquerda e, em seguida, uma CU de fundo à direita, nessa ordem. Quando o codificador de vídeo 20 realizar uma operação de codificação em uma CU particionada, o codificador de vídeo 20 pode codificar CUs associadas aos 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 associado a um sub-bloco superior à esquerda, uma CU associada a um sub-bloco superior à direita, um CU associada a um sub-bloco de fundo à esquerda e, em seguida, uma CU

associada a um sub-bloco de fundo à direita, nessa ordem.

[0060] Como resultado da codificação das CUs de um bloco de 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 determinada CU podem ter sido codificadas. As CUs abaixo e à direita da determinada CU ainda não foram codificadas. Consequentemente, o codificador de vídeo 20 pode ter capacidade para acessar as informações gerada codificando-se algumas CUs que circundam a determinada CU durante a codificação da determinada CU. Entretanto, o codificador de vídeo 20 pode não ter capacidade para acessar as informações gerada codificando-se outras CUs que circundam a determinada CU durante a codificação da determinada CU.

[0061] Quando o codificador de vídeo 20 codificar uma CU não particionada, o codificador de vídeo 20 pode gerar um ou mais unidades de predição (PUs) para a CU. Cada uma das PUs da CU pode ser 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 previsto para cada PU da CU. O bloco de vídeo previsto de uma PU pode ser um bloco de amostras. O codificador de vídeo 20 pode usar intrapredição ou interpredição para gerar o bloco de vídeo previsto para uma PU.

[0062] Quando o codificador de vídeo 20 usar a intrapredição para gerar o bloco previsto de uma PU, o codificador de vídeo 20 pode gerar o bloco previsto da PU com base nas amostras decodificadas da imagem associada à PU. Caso o codificador de vídeo 20 use intrapredição para gerar os blocos de vídeo previstos das PUs de uma CU, a CU é uma CU intraprevista. Quando o codificador de vídeo 20 usar a interpredição para gerar o bloco previsto da PU, o codificador de vídeo 20 pode gerar o bloco previsto da PU

com base nas amostras decodificadas da imagem associada à PU com base em amostras decodificadas de uma ou mais imagens diferentes da imagem associada à PU. Caso o codificador de vídeo 20 use interpredição para gerar os blocos de vídeo previstos das PUs de uma CU, a CU é uma CU interprevista.

[0063] Adicionalmente, quando o codificador de vídeo 20 usar interpredição para gerar um bloco de vídeo previsto para um 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 à PU. Em alguns exemplos, um bloco de referência de uma PU também pode ser denominado como "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.

[0064] Após o codificador de vídeo 20 gerar os blocos de vídeo previstos 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 previstos para os PUs da CU. Os dados residuais para a CU podem indicar diferenças entre amostras nos blocos de vídeo previstos para as PUs da CU e para o bloco de vídeo original da CU.

[0065] Adicionalmente, como parte da realização de uma operação de codificação em uma CU não particionada, o codificador de vídeo 20 pode realizar a partição de árvore quadrática recursiva nos dados residuais da CU a fim de 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 ser associada a um bloco de vídeo residual

diferente.

[0066] O codificador de vídeo 20 pode aplicar uma ou mais transformadas aos blocos de vídeo residuais associados às TUs para gerar os 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.

[0067] 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. A quantização geralmente se refere a um processo no qual coeficientes de transformada são quantizados para possivelmente reduzir a quantidade de dados usada para representar os coeficientes de transformada, o que fornece uma compactação adicional. O processo de quantização pode reduzir a profundidade de bits associada a alguns ou todos os coeficientes de transformada. Por exemplo, um coeficiente de transformada w -bit pode ser arredondado para baixo para um coeficiente de transformada m -bit durante a quantização, em que w é maior que m .

[0068] 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 ajustando-se o valor de QP associados à CU.

[0069] Após o codificador de vídeo 20 quantiza um bloco de coeficiente de transformada, o codificador de vídeo 20 pode gerar definições 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 contexto (CAVLC), codificação por entropia de particionamento de intervalo de probabilidade (PIPE) ou outra codificação aritmética binária também podem ser usadas.

[0070] 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 bytes que contêm os dados. Por exemplo, uma unidade de NAL pode conter dados que representam um conjunto de parâmetros de vídeo, um conjunto de parâmetros de sequência, um conjunto de parâmetros de imagem, uma fatia codificada, Informações de Aprimoramento Suplementar (SEI), um delimitador de unidade de acesso, dados de preenchimento ou outro tipo de dados. Os dados em uma unidade de NAL podem incluir várias estruturas de sintaxe.

[0071] 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 receber 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 realizar 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 geralmente recíproco ao processo realizado pelo codificador de vídeo 20 para gerar os elementos de sintaxe.

[0072] Após o decodificador de vídeo 30 extrair os elementos de sintaxe associados a uma CU, o decodificador de vídeo 30 pode gerar os blocos de vídeo previstos para as PUs da CU com base nos elementos de sintaxe. Além disso, o decodificador de vídeo 30 pode quantizar inversamente os blocos de coeficiente de transformada associados às TUs da CU. O decodificador de vídeo 30 pode realizar as transformadas inversas nos blocos de coeficiente de transformada para reconstruir os blocos de vídeo residuais associados às TUs da CU. Após gerar os blocos de vídeo previstos 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 previstos e nos blocos de vídeo residuais. Dessa forma, o decodificador de vídeo 30 pode reconstruir os blocos de vídeo das CUs com base nos elementos de sintaxe no fluxo de bits.

ENCODIFICADOR DE VÍDEO

[0073] A Figura 2A é um diagrama de blocos que ilustra um exemplo do codificador de vídeo 20 que pode implantar técnicas de acordo com aspectos descritos nesta revelação. 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 uma ou todas as técnicas desta revelação, incluindo, mas não limitado aos métodos de inferência da NoOutputOfPriorPicsFlag e processos relacionados descritos em maiores detalhes acima e abaixo em relação às Figuras 4 e 5. Como um exemplo, a unidade de processamento de predição 100 pode ser configurada para

realizar qualquer uma ou todas as técnicas descritas na presente revelação. Em outra modalidade, o codificador de vídeo 20 inclui uma unidade de predição intercamada opcional 128 que é configurada para realizar qualquer ou todas as técnicas descritas na presente revelação. Em outras modalidades, a predição intercamada pode ser realizada pela unidade de processamento de predição 100 (por exemplo, a unidade de interpredição 121 e/ou a unidade de intrapredição 126), nesse caso, a unidade de predição intercamada 128 pode ser omitida. Entretanto, os aspectos da presente revelação não se limitam a isso. Em alguns exemplos, as técnicas descritas na presente revelação 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 ou todas as técnicas descritas na presente revelação.

[0074] Para propósitos explicativos, a presente revelação descreve o codificador de vídeo 20 no contexto de codificação de HEVC. Entretanto, as técnicas da presente revelação podem ser aplicáveis a outros padrões ou métodos de codificação. O exemplo retratado na Figura 2 A é para um codec com uma camada. No entanto, conforme será descrito adicionalmente em referência à Figura 2B, alguns ou todos os codificadores de vídeo 20 podem ser duplicados para o processamento de um codec com múltiplas camadas.

[0075] O codificador de vídeo 20 pode realizar a intracodificação e a intercodificação de blocos de vídeo dentro de fatias de vídeo. A intracodificação tem base na predição espacial para reduzir ou remover a redundância espacial no vídeo dentro de um dado quadro ou imagem de vídeo. A intercodificação se baseia 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 diversos modos de codificação com base em espaço. Os intermodos, tais como a predição unidirecional (modo P) ou a predição bidirecional (modo B), podem se referir a qualquer um dentre os diversos modos de codificação com base em tempo.

[0076] No exemplo da Figura 2A, o codificador de vídeo 20 inclui uma pluralidade de componentes funcionais. Os componentes funcionais de codificador de vídeo 20 incluem 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 armazenamento temporário de imagem decodificada 114 e uma unidade de codificação por entropia 116. A unidade de processamento de predição 100 incluir 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 de predição intercamada 128. Em outros exemplos, o codificador de vídeo 20 pode incluir mais componentes funcionais, poucos componentes funcionais ou componentes funcionais diferentes. Além disso, a unidade de estimativa de movimento 122 e a unidade de compensação de movimento 124 podem ser altamente integradas, porém, são representadas no exemplo da Figura 2A separadamente para propósitos explicativos.

[0077] 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 da fonte de vídeo 18 (por exemplo, mostrado na Figura 1A ou 1B) ou outra fonte. Os dados de vídeo podem representar uma série de imagens. A

fim de 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 as 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 as operações de codificação em blocos de árvores na fatia.

[0078] Como parte da realização de uma operação de codificação em um bloco de árvore, a unidade de processamento de predição 100 pode realizar a partição de árvore quadrática no bloco de vídeo do bloco de 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 de árvore em quatro sub-blocos igualmente dimensionados, particionar um ou mais dentre os sub-blocos em quatro sub-sub-blocos igualmente dimensionados, e assim por diante.

[0079] Os tamanhos dos blocos de vídeo associados às CUs podem variar de amostras 8x8 até o tamanho do bloco de árvore com um máximo de amostras 64x64 ou maior. Nesta revelação, "NxN" e "N por N" podem ser usados de modo intercambiável para se referir às dimensões de amostras de um bloco de vídeo em termos de dimensões horizontal e vertical, por exemplo, amostras 16x16 ou amostras 16 por 16. Em geral, um bloco de vídeo 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 tem, em geral, N amostras em uma direção vertical e N amostras em uma direção horizontal, em que N representa um valor de número inteiro não negativo.

[0080] Além disso, como parte da realização da operação de codificação em um bloco de árvore, a unidade de processamento de predição 100 pode gerar uma estrutura de dados de árvore quadrática hierárquica para o bloco de árvore. Por exemplo, um bloco de árvore pode corresponder a um nó-raiz da estrutura de dados de árvore quadrática. Caso a unidade de processamento de predição 100 particione o bloco de vídeo do bloco de árvore em quatro sub-blocos, o nó-raiz tem quatro nós-filho na estrutura de dados de árvore quadrática. Cada um dos nós-filho corresponde a uma CU associada a um dentre os sub-blocos. Caso a unidade de processamento de predição 100 particione um dentre os sub-blocos em quatro sub-blocos, sendo que o nó corresponde à CU associada ao sub-bloco pode ter quatro nós-filho, cada um corresponde a uma CU associada a um dentre os sub-sub-blocos.

[0081] Cada nó da estrutura de dados de árvore quadrática pode conter dados de sintaxe (por exemplo, elementos de sintaxe) para o bloco de árvore ou CU correspondente. Por exemplo, um nó na árvore quadrática pode incluir um sinalizador de divisão que indica a possibilidade de o bloco de vídeo da CU correspondente ao nó ser particionado (por exemplo, dividido) em quatro sub-blocos. Os elementos de sintaxe para uma CU podem ser definidos recursivamente, e podem depender da possibilidade de o bloco de vídeo da CU ser dividido em sub-blocos. Uma CU cujo bloco de vídeo não é particionado pode corresponder a um nó-folha na estrutura de dados de árvore quadrática. Um bloco de árvore codificado pode incluir dados com base na estrutura de dados de árvore quadrática para um bloco de árvore correspondente.

[0082] O codificador de vídeo 20 pode realizar as operações de codificação em cada CU não particionada de um bloco de 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.

[0083] 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. Supondo que o tamanho de uma CU particular é $2N \times 2N$, o codificador de vídeo 20 e o decodificador de vídeo 30 podem suportar tamanhos de PU de $2N \times 2N$ ou $N \times N$ e interpredição em tamanhos simétricos de PU de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$, $2N \times nU$, $nL \times 2N$, $nR \times 2N$ ou semelhantes. O codificador de vídeo 20 e o decodificador de vídeo 30 também podem suportar partição assimétrica para os tamanhos de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ e $nR \times 2N$. Em alguns exemplos, a unidade de processamento de predição 100 pode realizar a partição geométrico de modo a particionar o bloco de vídeo de uma CU dentre as PUs da CU ao longo de um limite que não se encontra com os lados do bloco de vídeo da CU em ângulos retos.

[0084] A unidade de interpredição 121 pode realizar a interpredição em cada PU da CU. A interpredição pode fornecer a compactação temporal. Para realizar a interpredição 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 nas amostras decodificadas de imagens diferentes da imagem associada à CU (por exemplo, imagens de referência). Nesta revelação, um bloco de vídeo previsto gerado pela unidade de compensação de movimento 124 pode ser denominado como um bloco de vídeo interprevisto.

[0085] 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 operações diferentes para uma PU de uma CU dependendo de se a PU está em uma fatia I, uma fatia P ou uma fatia B. Em uma fatia I, todas as PUs são intraprevistas. Portanto, caso a PU esteja em uma fatia I, a unidade de estimativa de movimento 122 e a unidade de compensação de movimento 124 não realizam a interpredição na PU.

[0086] Caso a PU esteja em uma fatia P, a imagem que contém a PU está associada a uma lista de imagens de referência denominada como "lista 0". Cada uma das imagens de referência na lista 0 contém amostras que podem ser usadas para a interpredição de outras imagens. Quando a unidade de estimativa de movimento 122 realizar a operação de estimativa de movimento em relação a uma PU em uma fatia P, a unidade de estimativa de movimento 122 pode buscar nas imagens de referência da lista 0 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 correspondem mais proximamente 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 através da soma de diferença absoluta (SAD), soma de diferença quadrática (SSD) ou através de outras métricas de diferença.

[0087] 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 figura 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 variantes de precisão. Por exemplo, a unidade de estimativa de movimento 122 pode gerar vetores de movimento em precisão de amostra de um quarto, precisão de amostra de um oitavo ou outra precisão de amostra fracionária. No caso de da precisão de amostra fracionária, os valores de bloco de referência podem ser interpolados a partir de valores de amostra de 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.

[0088] Caso a PU esteja em uma fatia B, a imagem que contém a PU pode estar associada a duas listas de imagens de referência, denominadas como "lista 0" e "lista 1". Em alguns exemplos, uma imagem que contém uma fatia B pode ser associada a uma combinação de listas que é uma combinação da lista 0 com a lista 1.

[0089] Além disso, caso a PU esteja em uma fatia B, a unidade de estimativa de movimento 122 pode realizar predição unidirecional ou a predição bidirecional para a PU. Quando a unidade de estimativa de movimento 122 realiza a predição unidirecional para a PU, a unidade de estimativa de movimento 122 pode buscar nas imagens de referência da lista 0 ou da lista 1 um bloco de referência para a PU. Em seguida, a unidade de estimativa de movimento 122 pode gerar um índice de referência que indica imagem de referência na lista 0 ou na 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 na lista 1. A unidade de compensação de movimento 124 pode gerar os blocos de vídeo previsto da PU com base nos blocos de referência indicados pelas informações de movimento da PU.

[0090] Quando a unidade de estimativa de movimento 122 realiza a predição bidirecional para uma PU, a unidade de estimativa de movimento 122 pode buscar nas imagens de referência da lista 0 um bloco de referência para a PU, e pode também buscar nas imagens de referência da lista 1 outro bloco de referência para a PU. Em seguida, a unidade de estimativa de movimento 122 pode gerar índices de referência que indicam as imagens de referência na lista 0 e na 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 com as informações de movimento da PU. A unidade de compensação de movimento 124 pode gerar os blocos de vídeo previsto da PU com base nos blocos de referência indicados pelas informações de movimento da PU.

[0091] Em alguns exemplos, 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. Ao invés 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 semelhantes 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 da PU vizinha. Em outro exemplo, a unidade de estimativa de movimento 122 pode identificar, em uma estrutura de sintaxe associada à 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. Referindo-se às informações de movimento de uma primeira PU, durante a sinalização das informações de movimento de uma segunda PU, o codificador de vídeo 20 pode ter capacidade para sinalizar as informações de movimento da segunda PU como uso de menos bits.

[0092] Conforme ainda mais discutido abaixo, em referência às Figuras 8 a 9, a unidade de processamento de predição 100 pode ser configurada para codificar (por exemplo, codificar ou decodificar) a PU (ou quaisquer outros blocos de camada de referência e/ou de camada de aprimoramento ou unidades de vídeo) realizando-se os métodos ilustrados nas Figuras 8 a 9. 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 8 a 9, tanto juntos como separadamente.

[0093] Como parte de realizar 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 de predição para a PU com base em amostras decodificadas de outras PUs na mesma imagem. Os dados de predição para a PU podem incluir um bloco de vídeo previsto e vários elementos de sintaxe. A unidade de intrapredição 126 pode realizar a intrapredição em PUs em fatias I, fatias P e fatias B.

[0094] Para realizar a intrapredição em uma PU, a unidade de intrapredição 126 pode usar múltiplos modos de intrapredição para gerar múltiplos conjuntos de dados de predição para a PU. Quando a unidade de intrapredição 126 usa um modo de intrapredição para gerar um conjunto de dados de predição para a PU, a unidade de intrapredição 126 pode estender as amostras dos blocos de vídeo de PUs próximas através do bloco de vídeo da PU em uma direção e/ou um gradiente associados ao modo de intrapredição. As PUs próximas podem estar acima, acima e à direita, acima e à esquerda ou à esquerda da PU, assumindo uma ordem de codificação do fundo para o topo e da esquerda para direita para PUs, CUs e blocos de árvore. A unidade de intrapredição 126 pode usar várias quantidades de modos de intrapredição, por exemplo, 33 modos de intrapredição direcional, dependendo do tamanho da PU.

[0095] A unidade de processamento de predição 100 pode selecionar os dados de predição para uma PU dentre os dados de predição gerados pela unidade de compensação de movimento 124 para a PU ou os dados de predição gerados pela unidade de intrapredição 126 para a PU. Em alguns exemplos, a unidade de processamento de predição 100 seleciona os dados de predição para a PU com base nas métricas de taxa/distorção

dos conjuntos de dados de predição.

[0096] Caso a unidade de processamento de predição 100 selecione os dados de predição gerados pela unidade de intrapredição 126, a unidade de processamento de predição 100 pode sinalizar o modo de intrapredição que foi usado para gerar os dados de predição 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 formas. Por exemplo, pode ser provável que o modo de intrapredição selecionado seja igual ao modo de intrapredição de uma PU próxima. Em outras palavras, o modo de intrapredição da PU próxima pode ser o modo mais provável para a PU atual. Assim, a unidade de processamento de predição 100 pode gerar um elemento de sintaxe a fim de indicar que o modo de intrapredição selecionado é igual ao modo de intrapredição da PU próxima.

[0097] Conforme discutido acima, o codificador de vídeo 20 pode incluir a unidade de predição intercamada 128. A unidade de predição intercamada 128 é configurada para prever um bloco atual (por exemplo, um bloco atual na EL) com o uso de uma ou mais camadas diferentes que são disponíveis na SVC (por exemplo, uma camada-base ou de referência). Tal predição pode ser denominada como predição intercamada. A unidade de predição intercamada 128 utiliza os métodos de predição para reduzir a redundância intercamada, desse modo, aprimorando a eficiência de codificação e reduzindo as exigências de recurso computacional. Alguns exemplos de predição intercamada incluem intrapredição intercamada, predição de moção intercamada e predição residual intercamada. A intrapredição intercamada usa a reconstrução de blocos localizados simultaneamente na camada-base para prever o bloco atual na camada de aprimoramento. A predição de moção intercamada usa

as informações de movimento da camada-base para prever o movimento na camada de aprimoramento. A predição residual intercamada usa o resíduo da camada-base para prever o resíduo da camada de aprimoramento. Cada um dentre os esquemas de predição intercamada é discutido abaixo mais detalhadamente.

[0098] Após a unidade de processamento de predição 100 selecionar os dados de predição para as PUs de uma CU, a unidade de geração residual 102 pode gerar dados residuais para a CU subtraindo-se (por exemplo, indicados pelo sinal de menos) os blocos de vídeo previstos das PUs da CU do bloco de vídeo da CU. Os dados residuais de uma CU podem incluir blocos de vídeo residuais 2D que correspondem aos 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 às diferenças entre componentes de luminância de amostras nos blocos de vídeo previstos das PUs da CU e os componentes de luminância das 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 croma de amostras nos blocos de vídeo previstos das PUs da CU e os componentes de croma das amostras no bloco de vídeo original da CU.

[0099] A unidade de processamento de predição 100 pode realizar o particionamento de árvore quadrática para particionar os blocos de vídeo residuais da CU em sub-blocos. Cada bloco de vídeo residual não dividido pode ser 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 ser baseados nos tamanhos e nas posições de blocos de vídeo associados às PUs da CU. Uma estrutura de árvore quadrática conhecido como uma "árvore quadrática

residual” (RQT) pode incluir nós associados a cada um dos blocos de vídeo residuais. As TUs de uma CU podem corresponder aos nós-folha da RQT.

[0100] A unidade de processamento de transformada 104 pode gerar um ou mais blocos de coeficiente de transformada para cada TU de uma CU aplicando-se 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 2D de coeficientes de transformada. A unidade de processamento de transformada 104 pode aplicar diversas transformadas ao bloco de vídeo residual associado a uma TU. Por exemplo, a unidade de processamento de transformada 104 pode aplicar uma transformada de cosseno discreta (DCT), uma transformada direcional, ou uma transformada conceitualmente semelhante ao bloco de vídeo residual associado a uma TU.

[0101] 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.

[0102] 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 de árvore associado à CU. Na análise de taxa de distorção, o codificador de vídeo 20 pode gerar múltiplas representações codificadas do bloco de árvore realizando-se uma operação de codificação múltiplas vezes no bloco de árvore. O codificador de vídeo 20 pode associar diferentes valores de QP à CU quando o codificador de vídeo 20 gera diferentes representações codificadas do bloco de

árvore. O codificador de vídeo 20 pode sinalizar que um determinado valor de QP está associado à CU quando o determinado valor de QP estiver associado à CU em uma representação codificada do bloco de árvore que tem uma métrica de taxa de bits e de distorção mais inferior.

[0103] A unidade de quantização inversa 108 e a unidade de transformada inversa 110 pode 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 previstos de amostra gerados pela unidade de processamento de predição 100 para produzir um bloco de vídeo reconstruído associado a uma TU. Reconstruindo-se os blocos de vídeo para cada TU de uma CU, dessa maneira, o codificador de vídeo 20 pode reconstruir o bloco de vídeo da CU.

[0104] Após a unidade de reconstrução 112 reconstruir o bloco de vídeo de uma CU, a unidade de filtro 113 pode realizar uma operação de desblocagem a fim reduzir artefatos de blocagem no bloco de vídeo associado à CU. Após a realização das uma ou mais operações de desblocagem, a unidade de filtro 113 pode armazenar o bloco de vídeo reconstruído da CU no armazenamento temporário 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 em PUs de imagens subsequentes. Além disso, uma unidade de intrapredição 126 pode usar blocos de vídeo reconstruídos no armazenamento temporário de imagem decodificada 114 para realizar intrapredição em outras PUs na mesma imagem como da CU.

[0105] A unidade de codificação por entropia 116 pode receber dados provenientes 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 um ou mais operações de codificação por entropia de modo a gerar dados codificados por entropia. Por exemplo, o codificador de vídeo 20 pode realizar uma operação de codificação de comprimento variável adaptativa ao contexto (CAVLC), uma operação de CABAC, uma operação de codificação de comprimento variável-em-variável (V2V), uma operação de codificação aritmética binária adaptativa ao contexto com base em sintaxe (SBAC), uma operação de codificação por entropia de partição de intervalo de probabilidade (PIPE), ou outros tipos 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.

[0106] Com parte da realização de uma operação de codificação por entropia nos dados, a unidade de codificação por entropia 116 pode selecionar um modelo de contexto. Caso a unidade de codificação por entropia 116 realize uma operação de CABAC, o modelo de contexto pode indicar estimativas de probabilidades de compartimentos particulares que valores particulares. No contexto de CABAC, o termo "compartimento" é usado para se referir a um bit de uma versão binarizada de um elemento de sintaxe.

CODIFICADOR DE VÍDEO COM MÚLTIPLAS CAMADAS

[0107] A Figura 2B é um diagrama de blocos que ilustra um exemplo de um codificador de vídeo com múltiplas

camadas 23 (também referido simplesmente como codificador de vídeo com múltiplas camadas que 23) que pode implantar as técnicas em conformidade com os aspectos descritos na presente revelação. O codificador de vídeo 23 pode ser configurado para processar quadros de vídeo com múltiplas camadas, tais como, para a codificação SHVC e com múltiplas vistas. Além disso, o codificador de vídeo 23 pode ser configurado para realizar qualquer ou todas as técnicas da presente revelação.

[0108] O codificador de vídeo 23 inclui um codificador de vídeo 20A e um codificador de vídeo 20B, em que cada um 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 numerais de referência, os codificadores de vídeo 20A e 20B podem incluir pelo menos algum dentre os sistemas e subsistemas como o codificador de vídeo 20. Embora o codificador de vídeo 23 seja ilustrado como incluindo dois codificadores de vídeo 20A e 20B, o codificador de vídeo 23 não se limita a isso 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 do que quadros em uma unidade de acesso. Em alguns casos, algumas dentre as camadas de codificador de vídeo pode ser inativa quando o durante o processamento de algumas unidades de acesso.

[0109] 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, aumentar a resolução de uma camada-base de um quadro de vídeo recebido para, por exemplo, criar uma camada de aprimoramento. A unidade de reamostragem 90 pode aumentar a resolução de informações particulares associadas à camada-base recebida de um quadro, mas não outras informações. Por exemplo, a unidade de reamostragem 90 pode aumentar a resolução do tamanho espacial ou do número de pixels da camada-base, porém, o número de fatias ou a POC podem permanecer constantes. Em alguns casos, a unidade de reamostragem 90 pode não processar o vídeo recebido e/ou pode ser opcional. Por exemplo, em alguns casos, a unidade de processamento de predição 100 pode realizar aumento de resolução. Em algumas modalidades, a unidade de reamostragem 90 é configurada para realizar aumento de resolução de uma camada e reorganizar, redefinir, modificar ou ajustar uma ou mais fatias para estarem em conformidade com um conjunto de regras de limite de fatia e/ou regras de exame de varredura. Embora primariamente descrita como aumento de resolução de uma camada de base ou uma camada inferior em uma unidade de acesso, em alguns casos, a unidade de reamostragem 90 pode realizar a redução de resolução de uma camada. Por exemplo, caso durante a transmissão contínua de um vídeo a largura de banda seja reduzida, um quadro pode ter redução de resolução em vez de aumento de resolução.

[0110] A unidade de reamostragem 90 pode ser configurada para receber uma imagem ou quadro (ou informações de imagem associadas à imagem) do armazenamento temporário de imagem decodificada 114 do codificador de camada inferior (por exemplo, o codificador de vídeo 20A) e aumentar a resolução da imagem (ou as informações de imagem recebidas). Essa imagem de resolução aumentada 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 que 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 codificadores de camada superiores entre o codificador de vídeo de camada 0 e o codificador de camada 1 da Figura. 2B.

[0111] Em alguns casos, a unidade de reamostragem 90 pode ser omitida ou ignorada. Em tais casos, a imagem do armazenamento temporário de imagem decodificada 114 do codificador de vídeo 20A pode ser fornecido diretamente, ou pelo menos sem ser fornecido à unidade de reamostragem 90, à unidade de processamento de predição 100 do codificador de vídeo 20B. Por exemplo, caso os dados de vídeo fornecidos ao codificador de vídeo 20B e à imagem de referência do armazenamento temporário de imagem decodificada 114 do codificador de vídeo 20A tenham o mesmo tamanho ou resolução, a imagem de referência pode ser fornecida ao codificador de vídeo 20B sem qualquer reamostragem.

[0112] Em algumas modalidades, o codificador de vídeo 23 reduz a resolução dos dados de vídeo a serem fornecidos ao codificador de camada inferior com o uso da unidade de redução de resolução 94 antes que os dados de vídeo sejam fornecidos ao codificador de vídeo 20A. Alternativamente, a unidade de redução de resolução 94 pode ser uma unidade de reamostragem 90 com capacidade para aumentar ou reduzir a resolução dos dados de vídeo. Ainda em outras modalidades, a unidade de redução de resolução 94 pode ser omitida.

[0113] 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 combinados do codificador de vídeo 23. O fluxo de bits combinados pode ser criado tomando-se o fluxo de bits de cada um dentre os codificadores de vídeo 20A e 20B e alternando-se qual fluxo de bits é emitido em um determinado momento. Embora em alguns casos os bits dos dois (ou mais no caso de mais que duas camadas de encodificador de vídeo) fluxos de bits possam ser alternados um bit por vez, em muitos casos, os fluxos de bits são combinados diferentemente. Por exemplo, o fluxo de bits de saída pode ser criado alternando-se o fluxo de bits selecionado um bloco por vez. Em outro exemplo, o fluxo de bits de saída pode ser criado emitindo-se uma razão não 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 bloco emitido a partir do codificador de vídeo 20A. Em algumas modalidades, o fluxo de saída do mux 98 pode ser programado. Em outras modalidades, o mux 98 pode combinar os fluxos de bits dos codificadores de vídeo 20A, 20B com base em um sinal de controle recebido de um sistema externo ao codificador de vídeo 23, tal como, a partir de um processador em um dispositivo de origem que inclui o módulo de origem 12. O sinal de controle pode ser gerado com base na resolução ou taxa de bits de um vídeo da fonte de vídeo 18, com base em uma largura de banda do enlace 16, com base em uma assinatura associada a um usuário (por exemplo, uma assinatura paga versus uma assinatura gratuita), ou com base em qualquer outro fator para determinar uma saída de resolução desejada do codificador de vídeo 23.

DECODIFICADOR DE VÍDEO

[0114] A Figura 3A é um diagrama de blocos que ilustra um exemplo do decodificador de vídeo 30 que pode implantar técnicas de acordo com aspectos descritos nesta

revelação. O decodificador de vídeo 30 pode ser configurado para processar uma única camada de um quadro de vídeo, tal como, para HEVC. Adicionalmente, o decodificador de vídeo 30 pode ser configurado para realizar qualquer uma ou todas as técnicas desta revelação, incluindo, mas não limitado aos métodos de inferência da NoOutputOfPriorPicsFlag e processos relacionados descritos em maiores detalhes acima e abaixo em relação às Figuras 4 e 5. Por exemplo, a unidade de compensação de modo 162 e/ou a unidade de intrapredição 164 podem ser configuradas para realizar qualquer e todas as técnicas descritas na presente revelação. Em uma modalidade, o decodificador de vídeo 30 pode incluir opcionalmente a unidade de predição intercamada 166 que é configurada para realizar qualquer ou todas as técnicas descritas na presente revelação. Em outras modalidades, a predição intercamada pode ser realizada pela unidade de processamento de predição 152 (por exemplo, a unidade de compensação de movimento 162 e/ou a unidade de intrapredição 164), nesse caso, a unidade de predição intercamada 166 pode ser omitida. Entretanto, os aspectos da presente revelação não se limitam nesse sentido. Em alguns exemplos, as técnicas descritas nesta revelação 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 ou todas as técnicas descritas na presente revelação.

[0115] Para propósitos explicativos, a presente revelação descreve o decodificador de vídeo 30 no contexto de codificação de HEVC. Entretanto, as técnicas da presente revelação podem ser aplicáveis a outros padrões ou métodos de codificação. O exemplo retratado na Figura 3A é para um codec com camada única. No entanto, conforme será descrito adicionalmente em referência à Figura 3B, alguns ou todos os

decodificadores de vídeo 30 podem ser duplicados para o processamento de um codec com múltiplas camadas.

[0116] No exemplo da Figura 3A, o decodificador de vídeo 30 inclui uma pluralidade de componentes funcionais. Os componentes funcionais do 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 armazenamento temporário de imagem decodificada 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 intercamada 166. Em alguns exemplos, o decodificador de vídeo 30 pode realizar uma passagem de decodificação, em geral, 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 componentes funcionais, poucos componentes funcionais ou componentes funcionais diferentes.

[0117] O decodificador de vídeo 30 pode receber um fluxo de bits que compreende dados de vídeo codificados. O fluxo de bits pode incluir uma pluralidade de elementos de sintaxe. Quando o decodificador de vídeo 30 receber o fluxo de bits, a unidade de decodificação por entropia 150 pode realizar uma operação de análise no fluxo de bits. Como resultado da realização da operação de análise no fluxo de bits, a unidade de decodificação por entropia 150 pode extrair os 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 os elementos de sintaxe codificados 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.

[0118] 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 definição de parâmetro de vídeo, unidades de NAL de definição de parâmetro de sequência, unidades de NAL de definição de parâmetro 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 as definições de parâmetro de sequência das unidades de NAL de definição de parâmetro de sequência, a imagem definições de parâmetro das unidades de NAL de definição de parâmetro de imagem, os dados de SEI das unidades de NAL de SEI, e assim por diante.

[0119] Além disso, as unidades de NAL do fluxo de bits podem incluir unidades de NAL de fatia codificadas. Como parte da realização da operação de análise no fluxo de bits, a unidade de decodificação por entropia 150 pode realizar as operações de análise que extraem e decodificam por entropia as fatias codificadas 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 pertencentes 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 imagem que contém a fatia. A unidade de decodificação por entropia 150 pode realizar operações de decodificação por entropia, tal como operações de decodificação de CABAC, em elementos de sintaxe

no cabeçalho de fatia codificada para recuperar o cabeçalho de fatia.

[0120] Como parte de extrair os dados de fatia de unidades de NAL de fatia codificada, a unidade de decodificação por entropia 150 pode realizar operações de análise que extraem elementos de sintaxe de CUs codificadas nos dados de fatia. Os elementos de sintaxe extraídos podem incluir elementos de sintaxe associados a blocos de coeficiente de transformada. Em seguida, a unidade de decodificação por entropia 150 pode realizar as operações de decodificação de CABAC em alguns dos elementos de sintaxe.

[0121] 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. Realizando-se a operação de reconstrução para cada TU da CU, o decodificador de vídeo 30 pode reconstruir um bloco de vídeo residual associado à CU.

[0122] Como parte da realização de uma operação de reconstrução em uma TU, a unidade de quantização inversa 154 pode quantizar inversamente, por exemplo, desquantizar, um bloco de coeficiente de transformada associado à TU. A unidade de quantização inversa 154 pode quantizar inversamente o bloco de coeficiente de transformada de 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 a ser aplicado pela unidade de quantização inversa 154.

[0123] Após a unidade de quantização inversa 154 quantizar inversamente 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 outra transformada inversa ao bloco de coeficiente de transformada. Em alguns exemplos, a unidade de transformada inversa 156 pode determinar que uma transformada inversa se aplique 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 quadrática para um bloco de 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, tais como, tamanho de bloco, modo de codificação ou semelhantes. Em alguns exemplos, a unidade de transformada inversa 156 pode aplicar uma transformada inversa em cascata.

[0124] Em alguns exemplos, a unidade de compensação de movimento 162 pode refinar o bloco de vídeo previsto de uma PU realizando-se interpolação com base em filtros de interpolação. Os identificadores para filtros de

interpolação a serem usados para a compensação de movimento com precisão de redução de resolução 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 do PU a fim de calcular os valores interpolados para amostras de número abaixo de 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 pode usar os filtros de interpolação de modo a produzir o bloco de vídeo previsto.

[0125] Conforme ainda mais discutido abaixo, em referência às Figuras 8 a 9, a unidade de processamento de predição 152 pode codificar (por exemplo, codificar ou decodificar) a PU (ou quaisquer outros blocos de camada de referência e/ou de camada de aprimoramento ou unidades de vídeo) realizando-se os métodos ilustrados nas Figuras 8 a 9. 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 8 a 9, tanto juntos quanto separadamente.

[0126] Caso uma PU seja codificada como uso de 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.

[0127] Em alguns exemplos, os elementos de sintaxe podem indicar que a unidade de intrapredição 164

deve usar o modo de intrapredição de outra PU para determinar o modo de intrapredição da PU atual. Por exemplo, é provável que o modo de intrapredição da PU atual seja igual ao modo de intrapredição de uma PU próxima. Em outras palavras, o modo de intrapredição da PU próxima pode ser o modo mais provável para a PU atual. Portanto, nesse exemplo, o fluxo de bits pode incluir um elemento de sintaxe pequeno que indica que o modo de intrapredição da PU é igual ao modo de intrapredição da PU próxima. 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 PUs espacialmente próximas.

[0128] Conforme discutido acima, o decodificador de vídeo 30 também pode incluir a unidade de predição intercamada 166. A unidade de predição intercamada 166 é configurada para prever um bloco atual (por exemplo, um bloco atual na EL) com o uso de uma ou mais camadas diferentes que são disponíveis na SVC (por exemplo, uma camada-base ou de referência). Tal predição pode ser denominada como predição intercamada. A unidade de predição intercamada 166 utiliza os métodos de predição para reduzir a redundância intercamada, desse modo, aprimorando a eficiência de codificação e reduzindo as exigências de recurso computacional. Alguns exemplos de predição intercamada incluem intrapredição intercamada, predição de moção intercamada e predição residual intercamada. A intrapredição intercamada usa a reconstrução de blocos localizados simultaneamente na camada-base para prever o bloco atual na camada de aprimoramento. A predição de moção intercamada usa as informações de movimento da camada-base para prever o movimento na camada de aprimoramento. A predição residual intercamada usa o resíduo da camada-base para prever o resíduo da camada de aprimoramento. Cada um

dentre os esquemas de predição intercamada é discutido abaixo mais detalhadamente.

[0129] A unidade de reconstrução 158 pode usar os blocos de vídeo residuais associados às TUs de uma CU e os blocos de vídeo previstos das PUs da CU, por exemplo, tanto dados de intrapredição como dados de interpredição, conforme aplicável, para reconstruir o bloco de vídeo da CU. Assim, 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.

[0130] 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 os artefatos de bloqueio associados à CU. Após a unidade de filtro 159 realizar uma operação de desbloqueio para reduzir artefatos de bloqueio associados à CU, o decodificador de vídeo 30 pode armazenar o bloco de vídeo da CU em armazenamento temporário de imagem decodificada 160. O armazenamento temporário de imagem decodificada 160 pode fornecer imagens de referência para compensação de movimento subsequente, intrapredição e apresentação em um dispositivo de exibição, tal como o dispositivo de exibição 32 das Figuras 1A ou 1B. Por exemplo, o decodificador de vídeo 30 pode realizar, com base nos blocos de vídeo no armazenamento temporário de imagem decodificada 160, operações de intrapredição ou interpredição nas PUs ou outras CUs.

DECODIFICADOR COM MÚLTIPLAS CAMADAS

[0131] A Figura 3B é um diagrama de blocos que ilustra um exemplo de um decodificador de vídeo com múltiplas camadas 33 (também referido simplesmente como decodificador de vídeo com múltiplas camadas que 33) que pode implantar as

técnicas em conformidade com os aspectos descritos na presente revelação. O decodificador de vídeo 33 pode ser configurado para processar quadros de vídeo com múltiplas camadas, tais como para a codificação com múltiplas visualizações e SHVC. Além disso, o decodificador de vídeo 33 pode ser configurado para realizar qualquer parte ou todas as técnicas da presente revelação.

[0132] O decodificador de vídeo 33 inclui um decodificador de vídeo 30A e um decodificador de vídeo 30B, em que cada um 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 numerais 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 como inclusivo de dois decodificadores de vídeo 30A e 30B, o decodificador de vídeo 33 não se limita a isso 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 do que quadros em uma unidade de acesso. Em alguns casos, algumas dentre as camadas de decodificador de vídeo pode estar inativa quando o durante o processamento de algumas unidades de acesso.

[0133] Além dos decodificadores de vídeo 30A e 30B, o decodificador de vídeo 33 pode incluir uma unidade de aumento de resolução 92. Em algumas modalidades, a unidade

de aumento de resolução 92 pode aumentar a resolução de uma camada-base de um quadro de vídeo recebido para criar uma camada aprimorada a ser adicionada à lista de imagem de referência para o quadro ou unidade de acesso. Essa camada aprimorada pode ser armazenada no armazenamento temporário de imagem decodificada 160. Em algumas modalidades, a unidade de aumento de resolução 92 pode incluir algumas ou todas as modalidades descritas em relação à unidade de reamostragem 90 da Figura. 2A. Em algumas modalidades, a unidade de aumento de resolução 92 é configurada para aumentar a resolução de uma camada e reorganizar, redefinir, modificar ou ajustar uma ou mais fatias para estarem em conformidade com um conjunto de regras de limite de fatia e/ou regras de realização de varredura. Em alguns casos, a unidade de aumento de resolução 92 pode ser uma unidade de reamostragem configurada para aumentar a resolução e/ou reduzir a resolução de uma camada de um quadro de vídeo recebido.

[0134] A unidade de aumento de resolução 92 pode ser configurada para receber uma imagem ou quadro (ou informações de imagem associadas à imagem) do armazenamento temporário de imagem decodificada 160 do decodificador de camada inferior (por exemplo, o decodificador de vídeo 30A) e para aumentar a resolução da imagem (ou as informações de imagem recebidas). Essa imagem com aumento de resolução 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 que o 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 de camada 0 e o decodificador de camada 1 da

Figura 3B.

[0135] Em alguns casos, a unidade de aumento de resolução 92 pode ser omitida ou ignorada. Em tais casos, a imagem do armazenamento temporário de imagem decodificada 160 do decodificador de vídeo 30A pode ser fornecida diretamente, ou pelo menos sem ser fornecida à unidade de aumento de resolução 92, para a unidade de processamento de predição 152 do decodificador de vídeo 30B. Por exemplo, caso os dados de vídeo fornecidos ao decodificador de vídeo 30B e a imagem de referência do armazenamento temporário de imagem decodificada 160 do decodificador de vídeo 30A sejam do mesmo tamanho ou resolução, a imagem de referência pode ser fornecida ao decodificador de vídeo 30B sem aumento de resolução. Adicionalmente, em algumas modalidades, a unidade de aumento de resolução 92 pode ser uma unidade de reamostragem 90 configurada para aumentar a resolução ou reduzir a resolução de uma imagem de referência recebida do armazenamento temporário de imagem decodificada 160 do decodificador de vídeo 30A.

[0136] 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 alternando-se qual decodificador de vídeo recebe o fluxo de bits um bloco por vez. Em outro exemplo, o fluxo de bits pode ser dividido por uma razão de não 1:1 de blocos para qual cada um dentre os decodificadores de vídeo 30A e 30B. Por exemplo, dois blocos podem ser fornecidos ao decodificador de vídeo 30B para cada bloco fornecido ao decodificador de vídeo 30A. Em algumas modalidades, a divisão do fluxo de bits pelo demux 99 pode ser programada. Em outras modalidades, o demux 99 pode dividir o fluxo de bits com base em um sinal de controle recebido de um sistema externo ao decodificador de vídeo 33, tal como, de um processador em um dispositivo de destino incluindo o dispositivo 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 assinatura associada a um usuário (por exemplo, uma assinatura paga versus uma assinatura gratuita) ou com base em qualquer outro fator para determinar uma resolução obtenível pelo decodificador de vídeo 33.

IMAGENS DE PONTO DE ACESSO INTRA-ALEATÓRIO (IRAP)

[0137] Alguns esquemas de codificação de vídeo podem fornecer vários pontos de acesso aleatório ao longo de todo o fluxo de bits de modo que o fluxo de bits possa ser decodificado a partir de qualquer um desses pontos de acesso aleatórios sem precisar decodificar quaisquer imagens que precedam esses pontos de acesso aleatórios no fluxo de bits. Em tais esquemas de codificação de vídeo, as imagens que seguem um ponto de acesso aleatório em ordem de saída (por exemplo, incluindo essas imagens que estão na mesma unidade de acesso como a imagem que fornece o ponto de acesso aleatório) podem ser decodificadas corretamente sem usar quaisquer imagens que precedem o ponto de acesso aleatório.

Por exemplo, até mesmo caso uma porção do fluxo de bits seja perdida durante a transmissão ou durante a decodificação, um decodificador pode reiniciar a decodificação do fluxo de bits com início a partir do próximo ponto de acesso aleatório. Suporte para acesso aleatório pode facilitar, por exemplo, serviços de transmissão contínua dinâmica, operações de pesquisa, comutação de canal, etc. Em alguns casos, as imagens que seguem um ponto de acesso aleatório em ordem de saída podem usar imagens dianteiras para decodificação. As imagens dianteiras podem se referir às imagens que seguem o ponto de acesso aleatório no fluxo de bits e são emitidas e exibidas antes do ponto de acesso aleatório.

[0138] Em alguns esquemas de codificação, tais pontos de acesso aleatórios podem ser fornecidos pelas imagens que são denominadas de imagens de IRAP. Por exemplo, um ponto de acesso aleatório (por exemplo, fornecido por uma camada de aprimoramento imagem de IRAP) em uma camada de aprimoramento ("layerA") contida em uma unidade de acesso ("auA") pode fornecer acesso aleatório de camada específica de modo que para cada camada de referência ("layerB") da layerA (por exemplo, uma camada de referência que é uma camada que é usada para prever a layerA) que tem um ponto de acesso aleatório contido em uma unidade de acesso ("auB") que está na layerB e precede a auA em ordem de decodificação (ou um ponto de acesso aleatório contido na auA) as imagens na layerA, que seguem a auB em ordem de saída (incluindo as imagens localizadas na auB), possam ser corretamente passíveis de decodificação sem precisar decodificar quaisquer imagens na layerA que precede a auB. Por exemplo, quando a decodificação começa a partir de auB, e uma imagem na layerB e na auB é decodificável, então, a imagem em layerA que está na auA e as imagens em layerA que seguem auA são

decodificadas.

[0139] As imagens IRAP podem ser codificadas com o uso de intrapredição (por exemplo, codificadas sem se referir a outras imagens na mesma camada) e podem incluir, por exemplo, imagens de atualização de decodificador instantânea (IDR), imagens de acesso aleatório limpo (CRA) e imagens de acesso de enlace rompido (BLA). Quando há uma imagem de IDR no fluxo de bits, as imagens que precedem a imagem de IDR em ordem de decodificação não são usadas para predição pelas imagens que seguem a imagem de IDR em ordem de decodificação. Quando há uma imagem de CRA no fluxo de bits, as imagens que seguem a imagem de CRA podem ou não usar as imagens que precedem a imagem de CRA em ordem de decodificação para a predição. Essas imagens que seguem a imagem de CRA na ordem de decodificação, porém, usam imagens que precedem a imagem de CRA em ordem de decodificação, podem ser denominadas de imagens dianteiras omitidas de acesso aleatório (RASL). Outro tipo de imagem que segue uma imagem de IRAP em ordem de decodificação e que precede a imagem de IRAP em ordem de saída é uma imagem dianteira decodificável de acesso aleatório, que pode não conter referências a quaisquer imagens que precedem a imagem de IRAP em ordem de decodificação. As imagens de RASL podem ser descartadas pelo decodificador caso as imagens que precedem a imagem de CRA não estejam disponíveis. As imagens de RASL podem estar ou não estar presentes após uma imagem de BLA. Quando as imagens de RASL estão presentes após imagens de BLA, as mesmas devem ser ignoradas e/ou não decodificadas visto que suas imagens de referência podem não estar disponíveis. Uma unidade de acesso (por exemplo, um grupo de imagens que inclui todas as imagens codificadas associadas ao mesmo tempo de saída ao longo de camadas) que contém uma imagem de camada-base (por exemplo, a imagem que tem um valor de ID de camada igual a

0) que é uma imagem de IRAP pode ser denominada de uma unidade de acesso de IRAP.

ALINHAMENTO DE CAMADA CRUZADA DE IMAGENS DE IRAP

[0140] Em extensões de HEVC como SHVC e MV-HEVC, as imagens de IRAP podem não estar alinhadas (por exemplo, contidas na mesma unidade de acesso) através de diferentes camadas. Por exemplo, caso tenha sido exigido que as imagens de IRAP estejam alinhadas, qualquer unidade de acesso que contém pelo menos uma imagem de IRAP contém apenas as imagens de IRAP. Por outro lado, caso não tenha sido exigido que as imagens de IRAP estejam alinhadas, em uma única unidade de acesso, uma imagem (por exemplo, em uma primeira camada) pode ser uma imagem de IRAP, e outra imagem (por exemplo, em uma segunda camada) pode ser uma imagem que não seja de IRAP. Ter tais imagens de IRAP não alinhadas em um fluxo de bits pode fornecer algumas vantagens. Por exemplo, em um fluxo de bits de duas camadas, caso haja mais imagens de IRAP na camada-base do que na camada de aprimoramento, em aplicações de difusão e de difusão seletiva, o baixo atraso de sintonização e a alta eficiência de codificação podem ser alcançados.

[0141] Em alguns esquemas de codificação de vídeo, uma POC pode ser usada para acompanhar a ordem relativa na qual as imagens decodificadas são exibidas. Alguns dentre tais esquemas de codificação pode fazer com que os valores de POC sejam redefinidos (por exemplo, definidos como zero ou definidos como algum valor sinalizado no fluxo de bits) toda vez que determinados tipos de imagens aparecerem no fluxo de bits. Por exemplo, os valores de POC de determinadas imagens de IRAP podem ser redefinidos, o que faz com que os valores de POC de outras imagens que precedem essas imagens de IRAP em ordem de decodificação também sejam redefinidos. Isso pode ser problemático quando não é exigido

que as imagens de IRAP não estejam alinhadas através de diferentes camadas. Por exemplo, quando uma imagem ("picA") for uma imagem de IRAP, e outra imagem ("picB") na mesma unidade de acesso não for uma imagem de IRAP, o valor de POC de uma imagem ("picC"), que é redefinido devido à picA ser uma imagem de IRAP na camada que contém a picA, pode ser diferente do valor de POC de uma imagem ("picD"), que não é redefinido, na camada que contém a picB, em que a picC e a picD estão na mesma unidade de acesso. Isso faz com que a picC e a picD tenham diferentes valores de POC embora pertençam à mesma unidade de acesso (por exemplo, mesmo tempo de saída). Desse modo, nesse exemplo, o processo de derivação para derivar os valores de POC da picC e da picD podem ser modificados para produzir os valores de POC que são consistentes com a definição de valores de POC e das unidades acesso.

REDEFINIÇÃO DE POC PARA CODIFICAÇÃO DE MÚLTIPLAS CAMADAS

[0142] O elemento de sintaxe poc_reset_idc pode indicar se a POC deveria ser redefinida para uma imagem. O elemento de sintaxe poc_reset_idc pode indicar se o MSB da POC deve ser redefinido, ou tanto o MSB quanto o LSB da POC devem ser redefinidos, ou nenhum deve ser redefinido. Por exemplo, o valor de 0 para o poc_reset_idc indica que a POC não é redefinida. O valor de 1 para poc_reset_idc indica que o MSB de POC deve ser redefinido. O valor de 2 para poc_reset_idc indica que tanto o MSB de POC quanto o LSB devem ser redefinidos. O valor de 3 para o poc_reset_idc indica que a redefinição foi indicada para uma imagem anterior. Por exemplo, o valor de poc_reset_idc para a imagem anterior foi tanto 1 quanto 2. O valor de 3 para poc_reset_idc pode ser usado de modo que, quando a imagem em que a POC deve ser redefinida for perdida (por exemplo,

durante o processo de decodificação), a POC possa ser adequadamente redefinida em imagens subsequentes.

[0143] O `full_poc_reset_flag` pode indicar se a redefinição para a imagem anterior foi apenas para o MSB de POC, ou tanto para o MSB de POC quanto o LSB. Por exemplo, o valor de 0 para `full_poc_reset_flag` indica que apenas o MSB deve ser redefinido. O valor de 1 para o `full_poc_reset_flag` indica que tanto o MSB quanto o LSB devem ser redefinidos. O sinalizador `full_poc_reset_flag` pode ser usado em conexão com `poc_reset_idc`. Por exemplo, quando o valor de `poc_reset_idc` for 3, o `full_poc_reset_flag` pode indicar se a redefinição de POC para a imagem anterior for apenas para o MSB ou tanto para o MSB quanto o LSB.

[0144] Em versões anteriores de SHVC e MV-HEVC (por exemplo, SHVC Working Draft 6, MV-HEVC Working Draft 8, etc.), determinadas restrições ou restrições se aplicam, por exemplo, em relação ao `poc_reset_idc`. Entretanto, essas restrições não redefinem adequadamente a POC quando uma imagem não está presente ou quando uma imagem é descartável. Determinados detalhes relacionados a essas restrições e problemas associados a tais restrições são discutidos em maiores detalhes abaixo, por exemplo, em sessões que descrevem Imagens em Falta ou Ausentes, Imagens Descartáveis e Mistura de Imagens de BLA e CRA.

[0145] Além disso, nas versões anteriores de SHVC e MV-HEVC, não há restrições quanto ao valor de `full_poc_reset_flag` de uma imagem com base em `poc_reset_idc` da AU de redefinição de POC no mesmo período de redefinição de POC. Um valor incorreto de `full_poc_reset_flag` pode resultar em o mecanismo de redefinição de POC não funcionar adequadamente. Determinados detalhes relacionados ao `full_poc_reset_flag` são discutidos em maiores detalhes abaixo, por exemplo, na sessão que descreve o

full_poc_reset_flag.

[0146] A fim de solucionar esses e outros desafios, as técnicas de acordo com determinados aspectos redefinem a POC quando uma imagem não está presente (por exemplo, em falta ou ausente) ou quando uma imagem é descartável. As técnicas também impõem uma restrição quanto ao valor de full_poc_reset_flag de uma imagem com base no valor de poc_reset_idc. Dessa forma, as técnicas podem assegurar que a POC seja redefinida corretamente. Por exemplo, a POC pode ser redefinida corretamente quando faltar a imagem com poc_reset_idc igual tanto a 1 quanto a 2. Determinados detalhes relacionados à redefinição de POC são explicados adicionalmente abaixo.

IMAGENS EM FALTA OU AUSENTES

[0147] Algumas restrições relacionadas ao valor de poc_reset_idc podem ser da seguinte forma. Pode ser uma exigência de conformidade de fluxo de bits que as seguintes restrições se apliquem:

[0148] • O valor de poc_reset_idc não deve ser igual a 1 ou 2 para um acesso aleatório skipped leading (RASL) imagem, a imagem dianteira decodificável de acesso aleatório (RADL), uma imagem não de referência de subcamada, ou uma imagem que tem TemporalId maior do que 0, ou uma imagem que tem sinalizador descartável igual a 1. TemporalId pode indicar a ID de uma subcamada temporal de uma camada.

[0149] • O valor de poc_reset_idc de todas as imagens em uma AU deve ser igual.

[0150] • Quando a imagem em uma AU com nuh_layer_id igual a 0 for uma imagem de ponto de acesso intra-aleatório (IRAP) com um valor particular de nal_unit_type e houver pelo menos uma outra imagem na mesma AU com um valor diferente de nal_unit_type, o valor de poc_reset_idc deve ser igual a 1 ou 2 para todas as imagens

na AU.

[0151] • quando houver pelo menos uma imagem (i) que tem `nuh_layer_id` maior do que 0 e (ii) que for uma imagem de atualização de decodificador instantânea (IDR) com um valor particular de `nal_unit_type` em uma AU e (iii) houver pelo menos uma outra imagem na mesma AU com um valor diferente de `nal_unit_type`, então, o valor de `poc_reset_idc` deve ser igual a 1 ou 2 para todas as imagens na AU.

[0152] • O valor de `poc_reset_idc` de uma imagem de acesso aleatório desobstruído (CRA) ou de acesso de enlace rompido (BLA) deve ser menor do que 3.

[0153] • Quando a imagem com `nuh_layer_id` igual a 0 em uma AU for uma imagem de IDR e houver pelo menos uma imagem de não IDR na mesma AU, o valor de `poc_reset_idc` deve ser igual a 2 para todas as imagens na AU.

[0154] • Quando a imagem com `nuh_layer_id` igual a 0 em uma AU não for uma imagem de IDR, o valor de `poc_reset_idc` não deve ser igual a 2 para nenhuma imagem na AU.

[0155] Dentre as restrições listadas acima, o caso em que uma AU que tem imagens não presentes para algumas das camadas e que tem uma imagem de IRAP em pelo menos uma camada não é abrangido. Por exemplo, uma AU (por exemplo, AU D) contém uma imagem de IDR na camada-base (BL), porém, não contém nenhuma imagem na camada de aprimoramento (EL). Um caso em que uma camada não tem uma imagem em uma AU pode ser descrito como a AU que tem uma imagem em falta, uma imagem ausente, uma imagem não presente, etc. A camada pode não incluir uma imagem nessa AU particular, ou a imagem pode ter sido perdida (por exemplo, durante o processo de decodificação). Em tal situação, a POC deve ser redefinida visto que a POC da imagem de IDR na BL deve ser igual a 0. Entretanto, a cadeia de POC na camada de aprimoramento

continua. A falha na redefinição da POC das imagens de camada de aprimoramento na AU D poderia resultar em uma POC derivada que não é alinhada com a camada transversal para imagens na AU subsequentes. Entretanto, nenhuma dentre as restrições listadas acima para o valor de `poc_reset_idc` é aplicável para esta situação e, portanto, a redefinição de POC não é obrigatória para AU D.

[0156] Em outro exemplo, uma AU (por exemplo, AU D) contém uma imagem de IDR na EL, porém, não contém nenhuma imagem na BL. De modo similar, em tal situação, a POC pode ser redefinida visto que o MSB de POC da imagem de IDR na EL será redefinido para 0. Entretanto, com as abordagens existentes, a redefinição de POC não é obrigatória para a AU D.

[0157] Para assegurar um comportamento correto, a redefinição de POC pode ser exigida na AU D para ambos os casos descritos acima. Ademais, uma ou mais imagens e/ou AU que seguem imediatamente a AU D devem ter `poc_reset_idc` igual a 3. Com essas alterações, a redefinição de POC pode ser conduzida para a camada em que a imagem estava ausente assim que houver uma imagem em tal camada.

[0158] Pode haver casos em que a POC não precise ser redefinida para uma AU que contém uma imagem de IRAP e também imagens em falta. A Figura 4 ilustra exemplos de tais casos. A AU A não precisa redefinir a POC, visto que as imagens em falta são da Camada 2 e não há imagens em tal camada até a AU B que realizem a redefinição da POC. A AU B precisa redefinir a POC visto que a segunda AU após a AU B tem uma imagem na Camada 2, exceto se a segunda AU em si realizar redefinição de POC (no presente documento, presume-se que a segunda AU não realize). A AU C precisa redefinir a POC visto que tem uma IDR na Camada 0, porém, não IRAP(s) em outras camadas. A AU D não precisa redefinir a POC visto

que a AU tem uma IDR alinhada com camada transversal e nenhuma imagem em falta. Por fim, a AU E precisa redefinir a POC visto que a segunda AU após a AU E tem uma imagem na Camada 2, exceto se a segunda AU em si realizar a redefinição de POC (novamente, é presumido que a segunda AU não realize).

IMAGENS DESCARTÁVEIS

[0159] O problema descrito acima para as imagens em falta pode se aplicar de modo similar às imagens descartáveis, que, após serem descartadas se tornariam imagens ausentes.

[0160] Quando as imagens de IRAP são alinhadas com a camada transversal, a redefinição de POC não é exigida com as abordagens existentes. Entretanto, não há nenhuma restrição que proíba que uma imagem de IRAP seja marcada como uma imagem descartável. Quando uma AU ou uma sub-AU contém imagens de IRAP alinhadas com a camada transversal e pelo menos uma dentre as imagens de IRAP é descartável, a derivação de POC para as imagens nas AUs subsequentes, em ordem de decodificação, pode ser incorreta se a imagem descartável de IRAP na sub-AU tiver sido removida conforme ilustrado no exemplo da Figura 5. Quando a imagem de IDR de EL na AU E é descartada, o valor do bit mais significativo (MSB) da POC da imagem de EL na AU F pode ser derivado para que seja igual ao valor de MSB de POC da imagem de âncora anterior (por exemplo, 256). Isso pode resultar em valores de POC diferentes derivados para imagens na BL e na EL para a AU F.

MISTURA DE IMAGENS DE BLA E CRA

[0161] Com as abordagens existentes, quando uma imagem em uma AU tem `nuh_layer_id` igual a 0 e é uma imagem de IRAP com um valor particular de `nal_unit_type`, e há pelo menos uma outra imagem na mesma unidade de acesso com um valor diferente de `nal_unit_type`, o valor de `poc_reset_idc` é definido igual a 1 ou 2 para todas as imagens na AU. Isso

significa que se uma AU em um fluxo de bits de 2 camadas contiver uma imagem de CRA na BL e uma imagem de BLA na camada de aprimoramento, então, a redefinição de POC, tanto que redefine apenas o MSB de POC quanto que redefine tanto o MSB de POC quanto o bit menos significativo (LSB), é exigida. Tal constrição não é ideal, particularmente para determinados casos de uso.

[0162] A Figura 6 ilustra um exemplo de comutação descendente e comutação ascendente de camada, em que a comutação ascendente de camada é conduzida na AU que contém imagens de CRA alinhadas com a camada transversal, auA. Visto que não é exigido realizar a redefinição de POC quando os valores de `nal_unit_type` de IRAP são alinhados com a camada transversal, um codificador pode não sinalizar nenhuma informação de redefinição de POC (por exemplo, `poc_reset_idc`, `poc_reset_period_id`, etc.) na extensão de cabeçalho de segmento de fatia nas imagens de CRA da auA. Entretanto, para a comutação ascendente de camada, o sistema pode alterar o tipo de unidade de NAL da imagem de EL da auA de CRA para BLA. Em tal caso, a construção implica que, além de alterar o tipo de unidade de NAL, uma entidade de sistema que realiza a operação de comutação ascendente também precisa modificar a extensão de cabeçalho de segmento de fatia das imagens na auA para inserir informações de redefinição de POC. Tal peso adicional imposto à entidade de sistema é desnecessário, visto que a redefinição de POC não é, de fato, necessária na situação ilustrada. Isso se aplica de modo similar para divisão de fluxos de bits em imagens de CRA alinhadas com a camada transversal em que apenas a imagem de CRA de camada-base é alterada para ser uma imagem de BLA.

`full_poc_reset_flag`

[0163] Quando `poc_reset_idc` é igual a 3, `full_poc_reset_flag` é sinalizado para indicar se tanto o MSB

de POC quanto o LSB devem ser redefinidos ou se apenas o MSB de POC deve ser redefinido. O valor de `full_poc_reset_flag` pode indicar o valor de `poc_reset_idc` da primeira unidade de acesso no mesmo período de redefinição de POC. Em algumas modalidades, um período de redefinição de POC pode se referir a uma sequência de unidades de acesso em ordem de decodificação, a começar por uma unidade de acesso com `poc_reset_idc` igual a 1 ou 2 e um valor particular de `poc_reset_period_id` e incluindo todas as unidades de acesso tanto que têm o mesmo valor de `poc_reset_period_id` quanto que têm `poc_reset_idc` igual a 0. Se o valor errado de `full_poc_reset_flag` for sinalizado, o decodificador poderia decrementar a POC de imagens anteriores pelo valor errado. A Figura 7, fornecida abaixo, ilustra um exemplo em que o codificador atribui um valor incorreto para `full_poc_reset_flag`. Visto que o valor de `poc_reset_idc` da imagem em AU E é 2, o que significa que tanto o MSB de POC quanto o LSB são redefinidos, o valor de `full_poc_reset_flag` (quando presente, na imagem sucessora, na ordem de decodificação, AU E e que pertence ao mesmo período de redefinição de POC que a AU E) deve ser 1 para garantir que o valor correto de `DeltaPocVal` seja derivado para decrementar o valor de POC de imagens anteriores. Visto que o valor de `full_poc_reset_flag` é sinalizado incorretamente como 0 para cada imagem de Camada 1 em AU G, o valor usado para decrementar a POC de imagens da Camada 1 (que precedem, na ordem de decodificação, AU E) é 64 em vez de 102. Os valores de POC incorretos de imagens anteriores podem afetar o processo de derivação de seleção de imagem de referência (RPS) visto que o decodificador não pode encontrar as imagens de referência corretas para a imagem atual. No exemplo ilustrado da Figura 7, durante a derivação de RPS da imagem de Camada 1 de AU G, o decodificador não pode encontrar

imagens de referência com POC igual a - 1 e - 2 e, portanto, essas imagens de referência são marcadas "indisponíveis para referência".

REDEFINIÇÃO DE POC E VALOR DE DEFINIÇÃO DE full_poc_reset_flag

[0164] Para superar os problemas descritos acima com abordagens existentes para redefinição de POC em codecs de múltiplas camadas, esta revelação descreve os aprimoramentos abaixo. Na presente revelação, as seguintes técnicas e abordagens descritas podem ser usadas sozinhas ou em qualquer combinação.

[0165] De acordo com um ou mais aspectos das técnicas de redefinição de POC descritas no presente documento, é fornecida uma atualização de restrições relacionadas ao valor de poc_reset_idc. De tal modo, se houver pelo menos uma imagem em falta na AU auA que contém uma imagem de IRAP na BL ou uma imagem de IDR em uma EL, então, o valor de poc_reset_idc deve ser igual a 1 ou 2 para todas as imagens na AU.

[0166] Por exemplo, uma forma de detectar uma imagem em falta para a AU auA é a seguinte: quando o número de imagens na AU auA for menor do que o número de imagens no fluxo de bits, então, a AU auA pode ser indicada ou é possível pressupor que tenha as imagens em falta.

[0167] Em outro exemplo, o valor de poc_reset_idc é restringido a ser igual a 1 ou 2 quando há pelo menos uma imagem em falta na camada LayerIdA da AU auA que contém uma imagem de IRAP em uma BL ou uma imagem de IDR em uma EL e há outra AU auB que satisfaz pelo menos uma dentre as seguintes:

[0168] • pertence a um período de redefinição de POC diferente da unidade de acesso auA;

[0169] • tem uma imagem de IRAP com nuh_layer_id

igual a 0 e NoClrasOutputFlag igual a 1; ou

[0170] • é a última unidade de acesso no fluxo de bits em ordem de decodificação

[0171] e há uma imagem com nuh_layer_id igual a LayerIdA que segue auA em ordem de decodificação e precede a AU auB em ordem de decodificação.

[0172] Em outro exemplo, quando não há nenhuma imagem com nuh_layer igual a nenhum valor particular de nuh_layer_id, LayerIdA na unidade de acesso auA, e há uma imagem picA com nuh_layer_id igual a LayerIdA que segue auA em ordem de decodificação e a seguinte condição é verdadeira, então, a AU auA pode ser indicada ou é possível pressupor que a mesma tem imagens em falta:

[0173] • Deixar que auB seja a primeira AU que segue auA em ordem de decodificação e pertence a um período de redefinição de POC diferente da unidade de acesso auA, quando presente. Deixar auC ser a primeira unidade de acesso que segue auA em ordem de decodificação e tem uma imagem de IRAP com nuh_layer_id igual a 0 e NoClrasOutputFlag igual a 1, quando presente. Deixar auD ser a primeira unidade de acesso que segue auA em ordem de decodificação e tem uma imagem de IRAP em cada uma das camadas no fluxo de bits, quando presente. Nenhuma dentre auB, auC e auD existe no fluxo de bits, ou a imagem picA precede, em ordem de decodificação, a primeira dentre auB (quando presente), auC (quando presente) e auD (quando presente) em ordem de decodificação.

[0174] De acordo com um ou mais aspectos das técnicas de redefinição de POC descritas no presente documento, é fornecida uma atualização de restrições relacionadas ao valor de poc_reset_idc. De tal modo, quando há pelo menos uma imagem descartável na AU que contém a imagem de IRAP na BL ou uma imagem de IDR em uma EL, então,

o valor de poc_reset_idc deve ser igual a 1 ou 2 para todas as imagens na AU.

[0175] Em outro exemplo, uma constrição pode ser adicionada de modo que as imagens de IRAP não possam ter discardable_flag igual a 1.

[0176] Ainda em outro exemplo, uma constrição pode ser adicionada de modo que as imagens de IDR não possam ter discardable_flag igual a 1, mas as imagens de CRA e de BLA podem ter discardable_flag igual a 1.

[0177] De acordo com um ou mais aspectos das técnicas de redefinição de POC descritos no presente documento, uma constrição pode ser adicionada para poc_reset_idc para assegurar que, quando houver uma imagem em falta ou imagem descartável de uma camada layerA em uma unidade de acesso auA que contém uma imagem de IRAP na BL ou uma imagem de IDR em uma EL, a começar pela primeira AU após auA, em ordem de decodificação, que tem a imagem com nuh_layer_id igual à layerA até a primeira AU que segue auA, em ordem de decodificação, que contém imagem com nuh_layer_id igual a layerA que tem TemporalId igual a 0 e tem discardable_flag igual a 0, inclusive, o valor de poc_reset_idc deve ser igual a 3.

[0178] Em outro exemplo, poc_reset_idc deve ser igual a 3 para imagens da camada layerA contida em unidades de acesso que seguem, em ordem de decodificação, auA e precedem, em ordem de decodificação, a primeira AU após auA, em ordem de decodificação, que contém uma imagem com nuh_layer_id igual a layerA que tem TemporalId igual a 0, tem discardable_flag igual a 0 e não é uma imagem de RASL, RADL ou de não referência de subcamada, inclusive. Uma imagem de não referência de subcamada (SLNR) pode se referir a uma imagem que não é usada como uma imagem de referência por uma subcamada temporal. Por exemplo, uma camada pode consistir

em uma ou mais subcamadas temporais que são indicadas por uma ID temporal (por exemplo, TemporalId). Se uma imagem que está na camada mais alta e que está na subcamada temporal mais alta não for usada como uma referência por outras imagens na mesma camada, a mesma pode ser removida sem fazer com que o fluxo de bits se torne indecodificável. Nesse caso, a imagem pode ser similar a uma imagem descartável.

[0179] De acordo com um ou mais aspectos das técnicas de redefinição de POC descritas no presente documento, uma constrição pode ser adicionada para assegurar que o valor de full_poc_reset_flag de uma imagem indica o valor correto de poc_reset_idc da primeira AU no mesmo período de redefinição de POC que a imagem atual.

[0180] De acordo com um ou mais aspectos das técnicas de redefinição de POC descritas no presente documento, uma constrição existente para poc_reset_idc pode ser modificada para assegurar que quando uma AU contiver apenas imagens de CRA ou de BLA (porém, com qualquer tipo de imagem de BLA), a redefinição de POC não será exigida (por exemplo, poc_reset_idc não precisa ser igual a 1 ou 2).

MODALIDADES EXEMPLIFICATIVAS

[0181] As técnicas mencionadas acima podem ser implantadas conforme mostrado nos exemplos a seguir. Os exemplos são fornecidos no contexto de versões anteriores de SHVC e MV-HEVC (por exemplo, WD 6 de SHVC e WD 8 de MV-HEVC). As adições às versões anteriores de SHVC e MV-HEVC são indicadas em *itálico*, e eliminações das versões anteriores de SHVC e MV-HEVC são indicadas em ~~tachado~~.

MODALIDADE EXEMPLIFICATIVA 1

[0182] A modalidade 1 redefine a POC quando há uma imagem ausente ou uma imagem descartável em uma AU. Por exemplo, uma variável chamada de absentPictureInAuFlag pode indicar se uma AU contém uma imagem ausente ou uma imagem

descartável. Se a primeira AU no mesmo período de redefinição de POC tiver uma imagem ausente ou uma imagem descartável em uma camada particular, o valor de `poc_reset_idc` é definido como 3 para AUs subsequentes em ordem de decodificação, a começar por uma AU que tem uma imagem na mesma camada e a terminar em uma AU que tem uma imagem que está na mesma camada, tem uma `temporalID` igual a 0 e não é descartável, inclusive. Além disso, o valor de `full_poc_reset_flag` pode ser definido como o valor de `poc_reset_idc` menos 1 quando o valor de `poc_reset_idc` de imagens na primeira AU no mesmo período de redefinição de POC for igual a 1 ou 2.

[0183] Por exemplo, codificadores devem ser cautelosos ao definir o valor de `poc_reset_idc` de imagens quando houver uma imagem de uma camada não presente em uma AU ou houver uma imagem com `discardable_flag` igual a 1 presente em uma AU, para assegurar que os valores de POC derivados de imagens dentro de uma AU sejam alinhados com a camada transversal. Em algumas modalidades, tais casos devem ser tratados de modo similar às AUs com imagens de IRAP não alinhadas com camada transversal em termos da possibilidade de POC ser redefinida.

A variável `absentPictureInAuFlag` é derivada da seguinte forma:

- Se qualquer uma dentre as duas condições a seguir for verdadeira, o valor de `absentPictureInAuFlag` é definido como 1:

- Há pelo menos uma imagem na mesma unidade de acesso que tem `discardable_flag` igual a 1.

- O número de imagens na unidade de acesso é menor do que o número de camadas no fluxo de bits.

- De outro modo, o valor de `absentPictureInAuFlag` é definido como 0.

É uma exigência de conformidade de fluxo de bits que as seguintes restrições se apliquem:

- O valor de poc_reset_idc não deve ser igual a 1 ou 2 para uma imagem de RASL, uma imagem de RADL, uma imagem de não referência de subcamada ou uma imagem que tem TemporalId maior do que 0, ou uma imagem que tem discardable_flag igual a 1.

- O valor de poc_reset_idc de todas as imagens em uma unidade de acesso deve ser igual.

- Quando a imagem em uma unidade de acesso com nuh_layer_id igual a 0 for uma imagem de IRAP e houver pelo menos uma outra imagem na mesma unidade de acesso *que não é uma imagem de IRAP ou absentPictureInAuFlag for igual a 1*, o valor de poc_reset_idc deve ser igual a 1 ou 2 para todas as imagens na unidade de acesso.

- Quando houver pelo menos uma imagem que tem nuh_layer_id maior do que 0 e que é uma imagem de IDR com um valor particular de nal_unit_type em uma unidade de acesso e há pelo menos uma outra imagem na mesma unidade de acesso com um valor diferente de nal_unit_type ou *absentPictureInAuFlag for igual a 1*, o valor de poc_reset_idc deve ser igual a 1 ou 2 para todas as imagens na unidade de acesso.

- O valor de poc_reset_idc de uma imagem de CRA ou de BLA deve ser menor do que 3.

- Quando a imagem com nuh_layer_id for igual a 0 em uma unidade de acesso em uma imagem de IDR e houver pelo menos uma imagem de não IDR na mesma unidade de acesso ou *absentPictureInAuFlag for igual a 1*, o valor de poc_reset_idc deve ser igual a 2 para todas as imagens na unidade de acesso.

- Quando a imagem com nuh_layer_id for igual a 0

em uma unidade de acesso não for uma imagem de IDR, o valor de poc_reset_idc não deve ser igual a 2 para nenhuma imagem na unidade de acesso.

- Quando a primeira unidade de acesso auA no período de redefinição de poc não tiver nenhuma imagem em uma camada particular layerA ou tiver uma imagem que tem discardable_flag igual a 1 em uma camada particular layerA, o valor de poc_reset_idc de imagens nas unidades de acesso subsequentes após auA, em ordem de decodificação, a começar pela primeira unidade de acesso que tem imagem com nuh_layer_id igual à layerA até a unidade de acesso que tem imagem com nuh_layer_id igual à layerA, TemporalId igual a 0, e discardable_flag igual a 0, inclusive, deve ser igual a 3.

O valor de poc_reset_idc de uma unidade de acesso é o valor de poc_reset_idc das imagens na unidade de acesso.

poc_reset_period_id identifica um período de redefinição de POC. Não deve haver duas imagens consecutivas em ordem de decodificação na mesma camada que tenham o mesmo valor de poc_reset_period_id e poc_reset_idc igual a 1 ou 2. Quando não estiver presente, o valor de poc_reset_period_id é inferido da seguinte forma:

- Se a imagem anterior picA tiver poc_reset_period_id presente no cabeçalho de segmento de fatia presente na mesma camada do fluxo de bits que a imagem atual, o valor de poc_reset_period_id é inferido para ser igual ao valor do poc_reset_period_id de picA.

- Do contrário, o valor de poc_reset_period_id é inferido como igual a 0.

OBSERVAÇÃO - Não é proibido que múltiplas imagens em uma camada tenham o mesmo valor de poc_reset_period_id

e tenham poc_reset_idc igual a 1 ou 2, exceto se tais imagens ocorrerem em duas unidades de acesso consecutivas em ordem de decodificação. Para minimizar a probabilidade de tais duas imagens aparecerem no fluxo de bits devido às perdas de imagem, operações de extração, busca ou divisão de fluxo de bits, os codificadores devem definir o valor de poc_reset_period_id como um valor aleatório para cada período de redefinição de POC (sujeito às restrições especificadas acima).

É uma exigência de conformidade de fluxo de bits que as seguintes restrições se apliquem:

- Um período de redefinição de POC não deve incluir mais do que uma unidade de acesso com poc_reset_idc igual a 1 ou 2.

- Uma unidade de acesso com poc_reset_idc igual a 1 ou 2 deve ser a primeira unidade de acesso em um período de redefinição de POC.

- Uma imagem que segue, em ordem de decodificação, a primeira imagem de redefinição de POC dentre todas as camadas de um período de redefinição de POC em ordem de decodificação não deve preceder, em ordem de saída, outra imagem em nenhuma camada que preceda a primeira imagem de redefinição de POC em ordem de decodificação.

full_poc_reset_flag igual a 1 especifica que tanto os bits mais significativos quanto os bits menos significativos do valor de contagem de ordem de imagem para a imagem atual são redefinidos quando a imagem anterior em ordem de decodificação na mesma camada não pertence ao mesmo período de redefinição de POC. full_poc_reset_flag igual a 0 especifica que apenas os bits mais significativos do valor de contagem de ordem de imagem para a imagem atual

são redefinidos quando a imagem anterior em ordem de decodificação na mesma camada não pertence ao mesmo período de redefinição de POC.

É uma exigência de conformidade de fluxo de bits que, quando o valor de poc_reset_idc de imagens na primeira unidade de acesso no mesmo período de redefinição de POC for igual a 1 ou 2, o valor de full_poc_reset_flag, quando presente, deve ser igual a poc_reset_idc-1.

TABELA 1 - MODALIDADE EXEMPLIFICATIVA 1

MODALIDADE EXEMPLIFICATIVA 2

[0184] A presente modalidade tem por base a Modalidade 1, com a diferença de que a derivação do valor da variável `absentPictureInAuFlag` é da seguinte forma:

Quando uma unidade de acesso auA tem uma imagem de IRAP com nuh_layer_id igual a 0 ou uma imagem de IDR com nuh_layer_id maior do que 0, a variável absentPictureInAuFlag é derivada da seguinte forma para a unidade de acesso auA:

- Se qualquer uma dentre as seguintes duas condições for verdadeira, o valor de absentPictureInAuFlag é definido igual a 1:

- Há pelo menos uma imagem com discardable_flag igual a 1 na unidade de acesso auA.

- Há uma unidade de acesso auB que segue a unidade de acesso auA em ordem de decodificação e a unidade de acesso auB satisfaz uma ou mais dentre as seguintes condições:

- A unidade de acesso pertence a um período de redefinição de POC diferente da unidade de acesso auA.

- A unidade de acesso tem uma imagem de IRAP com nuh_layer_id igual a 0 e NoClrasOutputFlag igual a 1.

- A unidade de acesso é a última unidade de acesso

no fluxo de bits.

e, além disso, não há nenhuma imagem com um valor particular de nuh_layer_id LayerIdA na unidade de acesso auA, e há uma imagem com nuh_layer_id igual a LayerIdA que segue a unidade de acesso auA em ordem de decodificação e precede a unidade de acesso auB em ordem de decodificação.

- De outro modo, o valor de absentPictureInAuFlag é definido como 0.

TABELA 2 - MODALIDADE EXEMPLIFICATIVA 2

MODALIDADE EXEMPLIFICATIVA 3

[0185] A presente modalidade tem por base as Modalidades 1 e 2, com a diferença de que o valor de discardable_flag não pode ser 1 para imagens de IRAP, e a derivação da variável absentPictureInAuFlag não considera o valor do sinalizador descartável de imagens. As alterações na semântica de cabeçalho de fatia em relação à Modalidade 1 são mostradas abaixo:

discardable_flag igual a 1 especifica que a imagem codificada não é usada como uma imagem de referência para interpriedição e não é usada como uma imagem de referência intercamada no processo de decodificação de imagens subsequentes na ordem de decodificação, discardable_flag igual a 0 especifica que a imagem codificada pode ser usada como uma imagem de referência para interpriedição e pode ser usada como uma imagem de referência intercamada no processo de decodificação de imagens subsequentes na ordem de decodificação. Quando não estiver presente, o valor de discardable_flag é inferido como igual a 0.

Quando nal_unit_type for igual a TRAIL_R, TSA_R, STSA_R, RASL_R, RADL_R, BLA_W_LP, BLA_W_RADL, BLA_N_LP, IDR_W_LP, IDR_N_LP e CRA_NUT, o valor de discardable_flag

deve ser igual a 0.

A alteração na derivação da variável `absentPictureInAuFlag` pode ser da seguinte forma:

A variável `absentPictureInAuFlag` é derivada da seguinte forma:

- *Se o número de imagens na unidade de acesso é menor do que o número de camadas no fluxo de bits, o valor de `absentPictureInAuFlag` é definido como 1.*

- *De outro modo, o valor de `absentPictureInAuFlag` é definido como 0.*

Ou,

Quando uma unidade de acesso `auA` tem uma imagem de IRAP com `nuh_layer_id` igual a 0 ou uma imagem de IDR com `nuh_layer_id` maior do que 0, a variável `absentPictureInAuFlag` é derivada da seguinte forma para a unidade de acesso `auA`:

- *Se há uma unidade de acesso `auB` que segue a unidade de acesso `auA` em ordem de decodificação e a unidade de acesso `auB` satisfaz uma ou mais dentre as seguintes condições:*

- *A unidade de acesso pertence a um período de redefinição de POC diferente da unidade de acesso `auA`.*

- *A unidade de acesso tem uma imagem de IRAP com `nuh_layer_id` igual a 0 e `NoClrasOutputFlag` igual a 1.*

- *A unidade de acesso é a última unidade de acesso no fluxo de bits.*

e, além disso, não há nenhuma imagem com um valor particular de `nuh_layer_id` `LayerIdA` na unidade de acesso `auA`, e há uma imagem com `nuh_layer_id` igual a `LayerIdA` que segue a unidade de acesso `auA` e precede a unidade de acesso `auB` em ordem de decodificação.

- *De outro modo, o valor de `absentPictureInAuFlag`*

é definido como 0.

TABELA 3 - MODALIDADE EXEMPLIFICATIVA 3

MODALIDADE EXEMPLIFICATIVA 4

[0186] A presente modalidade tem por base a Modalidade 1, com a diferença de que a derivação do valor da variável `absentPictureInAuFlag` é da seguinte forma:

Quando uma unidade de acesso auA tem uma imagem de IRAP com `nuh_layer_id` igual a 0 ou uma imagem de IDR com `nuh_layer_id` maior do que 0, a variável `absentPictureInAuFlag` é derivada da seguinte forma para a unidade de acesso auA:

- Se qualquer uma dentre as seguintes duas condições for verdadeira, o valor de `absentPictureInAuFlag` é definido igual a 1:

- Há pelo menos uma imagem com `discardable_flag` igual a 1 na unidade de acesso auA.

- Há pelo menos uma imagem em falta na unidade de acesso auA. Em outras palavras, a seguinte condição é verdadeira. Deixar que auB seja a primeira unidade de acesso que segue auA em ordem de decodificação e pertence a um período de redefinição de POC diferente da unidade de acesso auA, quando presente. Deixar auC ser a primeira unidade de acesso que segue auA em ordem de decodificação e tem uma imagem de IRAP com `nuh_layer_id` igual a 0 e `NoClrasOutputFlag` igual a 1, quando presente. Deixar auD ser a primeira unidade de acesso que segue auA em ordem de decodificação e tem uma imagem de imagem de IRAP em cada uma das camadas no fluxo de bits, quando presente. Não há nenhuma imagem com `nuh_layer` igual a nenhum valor particular de `nuh_layer_id` `LayerIdA` na unidade de acesso auA, há uma imagem `picA` com `nuh_layer_id` igual a `LayerIdA`, que segue auA em ordem de decodificação e nenhuma dentre

auB, auC e auD existe no fluxo de bits, ou a imagem pica precede, em ordem de decodificação, a primeira dentre auB (quando presente), auC (quando presente) e auD (quando presente) em ordem de decodificação.

- De outro modo, o valor de absentPictureInAuFlag é definido como 0.

TABELA 4 - MODALIDADE EXEMPLIFICATIVA 4

MÉTODO DE REDEFINIÇÃO DE POC

[0187] A Figura 8 é um fluxograma que ilustra um método para codificar informações de vídeo de acordo com um ou mais aspectos da presente revelação. O método se refere à redefinição da POC. O processo 800 pode ser realizado por um codificador (por exemplo, o codificador conforme mostrado na Figura 2A, 2B, etc.), um decodificador (por exemplo, o decodificador conforme mostrado na Figura 3A, 3B, etc.), ou qualquer outro componente, dependendo da modalidade. Os blocos do processo 800 são descritos em relação ao decodificador 33 na Figura 3B, mas o processo 800 pode ser realizado por outros componentes, como um codificador, conforme mencionado acima. O decodificador de vídeo da camada 1 30B do decodificador 33 e/ou o decodificador da camada 0 30A do decodificador 33 pode realizar o processo 800, dependendo da modalidade. Todas as modalidades descritas em relação à Figura 8 podem ser implantadas separadamente, ou em combinação entre si. Certos detalhes em relação ao processo 800 são explicados acima, por exemplo, em relação à Figura 4 e 7.

[0188] O processo 800 começa no bloco 801. O decodificador 33 pode incluir uma memória (por exemplo, armazenamento temporário de imagem decodificada 160) para armazenar informações de vídeo associadas a uma pluralidade de camadas. O decodificador 33 pode obter informações associadas a uma AU atual a ser codificada. A AU atual pode

conter imagens de uma ou mais camadas dentre a pluralidade de camadas.

[0189] No bloco 802, o decodificador 33 determina se a AU atual inclui uma primeira camada que contém uma imagem de IRAP.

[0190] No bloco 803, o decodificador 33 determina se a AU atual inclui uma segunda camada que não contém nenhuma imagem ou que contém uma imagem descartável.

[0191] No bloco 804, o decodificador 33 redefine a POC da segunda camada na AU atual, em resposta à determinação de que a AU atual inclui (1) uma primeira camada que contém uma imagem de IRAP e (2) uma segunda camada que não contém nenhuma imagem ou que contém uma imagem descartável. Em algumas modalidades, o decodificador 33 redefine a POC da primeira camada na AU atual, em resposta à determinação de que a AU atual inclui (1) uma primeira camada que contém uma imagem de IRAP e (2) uma segunda camada que não contém nenhuma imagem ou que contém uma imagem descartável. Em uma modalidade, a primeira camada é uma camada-base. Em outra modalidade, a primeira camada é uma camada de aprimoramento e a imagem de IRAP é uma imagem de IDR. O decodificador 33 pode redefinir a POC de uma camada definindo-se um valor de um primeiro elemento de sintaxe indicativo da possibilidade de redefinir a POC. Em uma modalidade, o primeiro elemento de sintaxe é `poc_reset_idc`.

[0192] Em determinadas modalidades, o decodificador 33, em resposta à redefinição da POC da segunda camada na AU atual, para uma AU que é subsequente à AU atual em ordem de decodificação, define o valor do primeiro elemento de sintaxe para indicar que a POC da segunda camada foi redefinida em uma AU anterior.

[0193] Em algumas modalidades, o decodificador 33 define o valor do primeiro elemento de sintaxe para

indicar que a POC da segunda camada foi redefinida em uma AU anterior para uma ou mais AUs subsequentes à AU atual em ordem de decodificação. As uma ou mais AUs subsequentes podem começar com uma primeira AU que contém uma primeira imagem que tem a mesma ID de camada que a segunda camada e terminar com uma segunda AU subsequente à primeira AU em ordem de decodificação que contém uma segunda imagem que tem a mesma ID de camada que a segunda camada, inclusive. Em uma modalidade, a segunda imagem tem uma ID temporal igual a 0 e não é uma imagem descartável. Em determinadas modalidades, adicionalmente, a segunda imagem não é uma imagem de RASL, não é uma imagem de RADL, ou não é uma SLNR da segunda camada.

[0194] Em determinadas modalidades, o decodificador 33 é um dispositivo de comunicação sem fio, que compreende adicionalmente um receptor configurado para receber dados de vídeo de acordo com pelo menos uma tecnologia de acesso via rádio (RAT), em que os dados de vídeo compreendem as informações de vídeo associadas à pluralidade de camadas; e um transmissor configurado para operar de acordo com a pelo menos uma RAT. O dispositivo de comunicação sem fio pode ser um telefone celular, e os dados de vídeo recebidos podem ser recebidos pelo receptor e podem ser modulados de acordo com o padrão de comunicação celular.

[0195] Em algumas modalidades, o processo 800 pode ser executável em um dispositivo de comunicação sem fio que compreende: um receptor configurado para receber dados de vídeo de acordo com pelo menos uma tecnologia de acesso via rádio (RAT), em que os dados de vídeo compreendem as informações de vídeo associadas à pluralidade de camadas; um transmissor configurado para operar de acordo com a pelo menos uma RAT; uma memória configurada para armazenar os dados de vídeo; e um processador configurado para executar

instruções para processar os dados de vídeo armazenados na memória. O dispositivo de comunicação sem fio pode ser um telefone celular, e os dados de vídeo recebidos podem ser recebidos pelo receptor e podem ser modulados de acordo com o padrão de comunicação celular.

[0196] O processo 800 termina no bloco 805. Os blocos podem ser adicionados e/ou omitidos no processo 800, dependendo da modalidade, os blocos do processo 800 podem ser realizados em ordens diferentes, dependendo da modalidade.

[0197] Quaisquer recursos e/ou modalidades descritos em relação à redefinição da POC na presente revelação particular podem ser implantados separadamente ou em qualquer combinação dos mesmos. Por exemplo, quaisquer recursos e/ou modalidades descritas em conexão com as Figuras 1 a 7 e outras partes da revelação podem ser implantados em qualquer combinação com quaisquer recursos e/ou modalidades descritas em conexão com a Figura 8, e vice-versa.

MÉTODO PARA DEFINIR `full_poc_reset_flag`

[0198] A Figura 9 é um fluxograma que ilustra um método para codificar informações de vídeo de acordo com um ou mais aspectos da presente revelação. O método se refere à definição do valor de `full_poc_reset_flag`. O processo 900 pode ser realizado por um codificador (por exemplo, o codificador conforme mostrado na Figura 2A, 2B, etc.), um decodificador (por exemplo, o decodificador conforme mostrado na Figura 3A, 3B, etc.), ou qualquer outro componente, dependendo da modalidade. Os blocos do processo 900 são descritos em relação ao decodificador 33 na Figura 3B, mas o processo 900 pode ser realizado por outros componentes, como um codificador, conforme mencionado acima. O decodificador de vídeo da camada 1 30B do decodificador 33 e/ou o decodificador da camada 0 30A do decodificador 33

pode realizar o processo 900, dependendo da modalidade. Todas as modalidades descritas em relação à Figura 9 podem ser implantadas separadamente, ou em combinação entre si. Certos detalhes em relação ao processo 900 são explicados acima, por exemplo, em relação à Figura 4 e 7.

[0199] O processo 900 começa no bloco 901. O decodificador 33 pode incluir uma memória (por exemplo, armazenamento temporário de imagem decodificada 160) para armazenar informações de vídeo associadas a uma pluralidade de camadas. O decodificador 33 pode obter informações associadas a uma AU atual a ser codificada. A AU atual pode conter imagens de uma ou mais camadas dentre a pluralidade de camadas.

[0200] No bloco 902, o decodificador 33 redefine a POC de uma camada incluída na AU atual por meio de (1) redefinição apenas do MSB da POC ou (2) redefinição tanto do MSB da POC quanto do LSB da POC. Por exemplo, a POC da camada é redefinida na AU atual.

[0201] No bloco 903, para imagens em uma ou mais AUs subsequentes à AU atual em ordem de decodificação, o decodificador 33 define o valor de um primeiro sinalizador indicativo da possibilidade de uma redefinição da POC ser uma redefinição total. Em uma modalidade, o valor do primeiro sinalizador é definido igual a 0 quando a POC é redefinida através de (1) redefinição apenas do MSB da POC e o valor do primeiro sinalizador é definido igual a 1 quando a POC é redefinida através de (2) redefinição tanto do MSB da POC quanto do LSB da POC. As imagens das uma ou mais camadas na AU atual podem ter o mesmo período de redefinição de POC. Em uma modalidade, o primeiro sinalizador é `full_poc_reset_flag`. O valor do primeiro sinalizador pode corresponder ao valor de um segundo sinalizador indicativo da possibilidade de redefinir a POC. Em uma modalidade, o

segundo sinalizador é poc_reset_idc.

[0202] Em algumas modalidades, o valor do primeiro sinalizador é definido igual a 0 quando o segundo sinalizador indica que o valor do segundo sinalizador indica que apenas o MSB da POC deve ser redefinido, e em que o valor do segundo sinalizador é definido igual a 1 quando o segundo sinalizador indica que tanto o MSB quanto o LSB da POC devem ser redefinidos.

[0203] Em determinadas modalidades, o decodificador 33 é um dispositivo de comunicação sem fio, que compreende adicionalmente um receptor configurado para receber dados de vídeo de acordo com pelo menos uma tecnologia de acesso via rádio (RAT), em que os dados de vídeo compreendem as informações de vídeo associadas à pluralidade de camadas; e um transmissor configurado para operar de acordo com a pelo menos uma RAT. O dispositivo de comunicação sem fio pode ser um telefone celular, e os dados de vídeo recebidos podem ser recebidos pelo receptor e podem ser modulados de acordo com o padrão de comunicação celular.

[0204] Em algumas modalidades, o processo 900 pode ser executável em um dispositivo de comunicação sem fio que compreende: um receptor configurado para receber dados de vídeo de acordo com pelo menos uma tecnologia de acesso via rádio (RAT), em que os dados de vídeo compreendem as informações de vídeo associadas à pluralidade de camadas; um transmissor configurado para operar de acordo com a pelo menos uma RAT; uma memória configurada para armazenar os dados de vídeo; e um processador configurado para executar instruções para processar os dados de vídeo armazenados na memória. O dispositivo de comunicação sem fio pode ser um telefone celular, e os dados de vídeo recebidos podem ser recebidos pelo receptor e podem ser modulados de acordo com o padrão de comunicação celular.

[0205] O processo 900 termina no bloco 904. Os blocos podem ser adicionados e/ou omitidos no processo 900, dependendo da modalidade, os blocos do processo 900 podem ser realizados em ordens diferentes, dependendo da modalidade.

[0206] Quaisquer recursos e/ou modalidades descritos em relação à definição do valor de `full_poc_reset_flag` na presente revelação podem ser implantados separadamente ou em qualquer combinação dos mesmos. Por exemplo, quaisquer recursos e/ou modalidades descritas em conexão com as Figuras 1 a 7 e outras partes da revelação podem ser implantados em qualquer combinação com quaisquer recursos e/ou modalidades descritas em conexão com a Figura 9, e vice-versa.

[0207] As informações e os sinais revelados no presente documento podem ser representados com o 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.

[0208] As várias etapas de blocos lógicos ilustrativos, circuitos e algoritmos descritos em conjunto com as modalidades reveladas no presente documento podem ser implantadas como hardware eletrônico, software de computador ou combinações de ambos. A fim de ilustrar claramente essa possibilidade de intercâmbio de hardware e software, vários componentes, blocos, módulos, circuitos e etapas ilustrativos foram descritos acima geralmente em termos de funcionalidade dos mesmos. A possibilita de tal

funcionalidade ser implantada como hardware ou software depende da aplicação particular e das restrições de projeto impostas no sistema geral. As pessoas versadas na técnica podem implantar a funcionalidade descrita de várias maneiras para cada aplicação em particular, porém, tais decisões de implantação não devem ser interpretadas como causadoras de um afastamento do escopo da presente revelação.

[0209] As técnicas descritas no presente documento podem ser implantadas em hardware (por exemplo, hardware de computador), software, firmware ou em qualquer combinação dos mesmos. Tais técnicas podem ser implantadas em qualquer um dentre uma variedade de dispositivos, tais como computadores de propósito geral, aparelhos de telefone de dispositivo de comunicação sem fio ou dispositivos de circuito integrado que têm múltiplos usos incluindo a aplicação em aparelhos de telefone de dispositivo de comunicação sem fio e outros dispositivos. Quaisquer recursos descritos como módulos ou componentes podem ser implantados juntos em um dispositivo de lógica integrada ou separadamente como dispositivos de lógica distinta, porém, interoperável. Caso sejam implantadas em software, as técnicas podem ser realizadas pelo menos em parte por uma mídia de armazenamento de dados legível por computador que compreende código de programa incluindo instruções que, quando executadas, realizam um ou mais dos métodos descritos acima. O meio legível por computador pode ser um meio legível por computador não transitório. Uma mídia legível por computador pode formar parte de um produto de programa de computador, que pode incluir materiais de empacotamento. A mídia legível por computador pode compreender mídias de armazenamento de dados ou memória, tais como memória de acesso aleatório (RAM), tal como memória de acesso aleatório dinâmica síncrona (SDRAM), memória somente de leitura (ROM),

memória de acesso aleatório não volátil (NVRAM), memória somente de leitura programável eletricamente apagável (EEPROM), memória Flash, mídias de armazenamento de dados magnéticos ou ópticos e similares. As técnicas, adicional ou alternativamente, podem ser realizadas, pelo menos em parte, por uma mídia 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, tais como ondas ou sinais propagados.

[0210] 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 de aplicação específica (ASICs), matrizes lógicas programáveis em campo (FPGAs) ou outro conjunto de circuitos lógico discreto ou integrado equivalente. Tal processador pode ser configurado para realizar qualquer uma das técnicas descritas nesta revelação. Um processador de propósito geral pode ser um microprocessador, porém, na alternativa, o processador pode ser qualquer processador, controlador, microcontrolador ou máquina em 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 combinação com um núcleo de DSP ou qualquer outra tal conimagem). Consequentemente, o termo "processador", conforme usado no presente documento, pode se referir a qualquer uma das estruturas supracitadas, qualquer combinação das estruturas supracitadas ou qualquer outra estrutura ou aparelho adequado para implantação das técnicas descritas no presente documento. Além disso, em

alguns aspectos, a funcionalidade descrita no presente documento pode ser fornecida dentro de módulos de software ou módulos de hardware dedicados configurados para codificar e decodificar ou incorporados em um decodificador-codificador de vídeo combinado (CODEC). Também, as técnicas poderiam ser plenamente implantadas em um ou mais circuitos ou elementos lógicos.

[0211] As técnicas desta revelação podem ser implantadas em uma ampla variedade de dispositivos ou aparelhos, incluindo um aparelho sem fio, um circuito integrado (IC) ou um conjunto de ICs (por exemplo, um conjunto de chips). Vários componentes, módulos ou unidades são descritos nesta revelação 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.

[0212] Várias modalidades da revelação foram descritas. Essas e outras modalidades estão dentro do escopo das reivindicações a seguir.

REIVINDICAÇÕES

1. Método de codificação de informações de vídeo por Codificação de Vídeo de Alta Eficiência, HEVC, compreendendo:

armazenar informações de vídeo HEVC associadas a uma pluralidade de camadas;

obter informações associadas a uma unidade de acesso, AU, atual a ser codificada, em que a AU atual contém imagens de uma ou mais camadas dentre a pluralidade de camadas,

o método caracterizado pelo fato de que compreende adicionalmente, se a pluralidade de camadas compreender pelo menos uma primeira camada que contém uma imagem de IRAP e uma segunda camada que não contém nenhuma imagem ou que contém uma imagem descartável:

indicar, por definição apropriada de um valor de um sinalizador poc_reset_idc, uma redefinição (902) de uma contagem de ordem de imagem, POC, na primeira camada incluída na AU atual e realizar a redefinição da POC da camada na AU atual, a POC compreendendo bits mais significativos, MSBs, e bits menos significativos, LSBs, a redefinição sendo:

- 1) redefinição de apenas os MSBs da POC; ou
- 2) redefinição tanto dos MSBs da POC quanto dos LSBs da POC; e

em imagens na segunda camada de uma ou mais AUs subsequentes à AU atual em ordem de decodificação, definir (903) um valor de um sinalizador full_poc_reset indicativo de se a redefinição da POC da primeira camada da AU atual é uma redefinição total,

em que o valor do sinalizador full_poc_reset é definido igual a 0 quando a POC é redefinida por redefinição somente dos MSBs da POC,

em que o valor do sinalizador full_poc_reset é

definido igual a 1 quando a POC é redefinida por redefinição tanto dos MSBs da POC quanto dos LSBs da POC,

em que as imagens a partir das uma ou mais camadas na AU atual têm o mesmo período de redefinição de POC,

em que o valor do sinalizador `full_poc_reset` corresponde ao valor de um sinalizador `poc_reset_idc` indicativo da possibilidade de redefinir a POC, de modo que o valor do sinalizador `full_poc_reset` é definido igual a 0 quando o valor do sinalizador `poc_reset_idc` indica que apenas os MSBs da POC devem ser redefinidos, e o valor do sinalizador `full_poc_reset` é definido igual a 1 quando o valor do sinalizador `poc_reset_idc` indicar que tanto os MSBs quanto os LSBs da POC devem ser redefinidos.

2. Memória legível por computador, caracterizada pelo fato de que compreende instruções que, quando executadas em um processador fazem com que o processador realize o método conforme definido na reivindicação 1.

3. Aparelho para codificar informações de vídeo por Codificação de Vídeo de Alta Eficiência, HEVC, caracterizado pelo fato de que compreende:

uma unidade de memória para armazenar informações de vídeo HEVC associadas a uma pluralidade de camadas; e

um processador operacionalmente acoplado à unidade de memória e configurado para realizar as etapas de método conforme definido na reivindicação 1.

4. Aparelho, de acordo com a reivindicação 3, caracterizado pelo fato de que o aparelho é um dispositivo de comunicação sem fio que compreende adicionalmente:

um receptor configurado para receber dados de vídeo de acordo com pelo menos uma tecnologia de acesso via rádio, RAT, em que os dados de vídeo compreendem as informações de vídeo associadas à pluralidade de camadas; e

um transmissor configurado para operar de acordo

com a pelo menos uma RAT.

5. Aparelho, de acordo com a reivindicação 3, caracterizado pelo fato de que o dispositivo de comunicação sem fio é um telefone celular e os dados de vídeo recebidos são recebidos pelo receptor e são modulados de acordo com um padrão de comunicação celular.

10

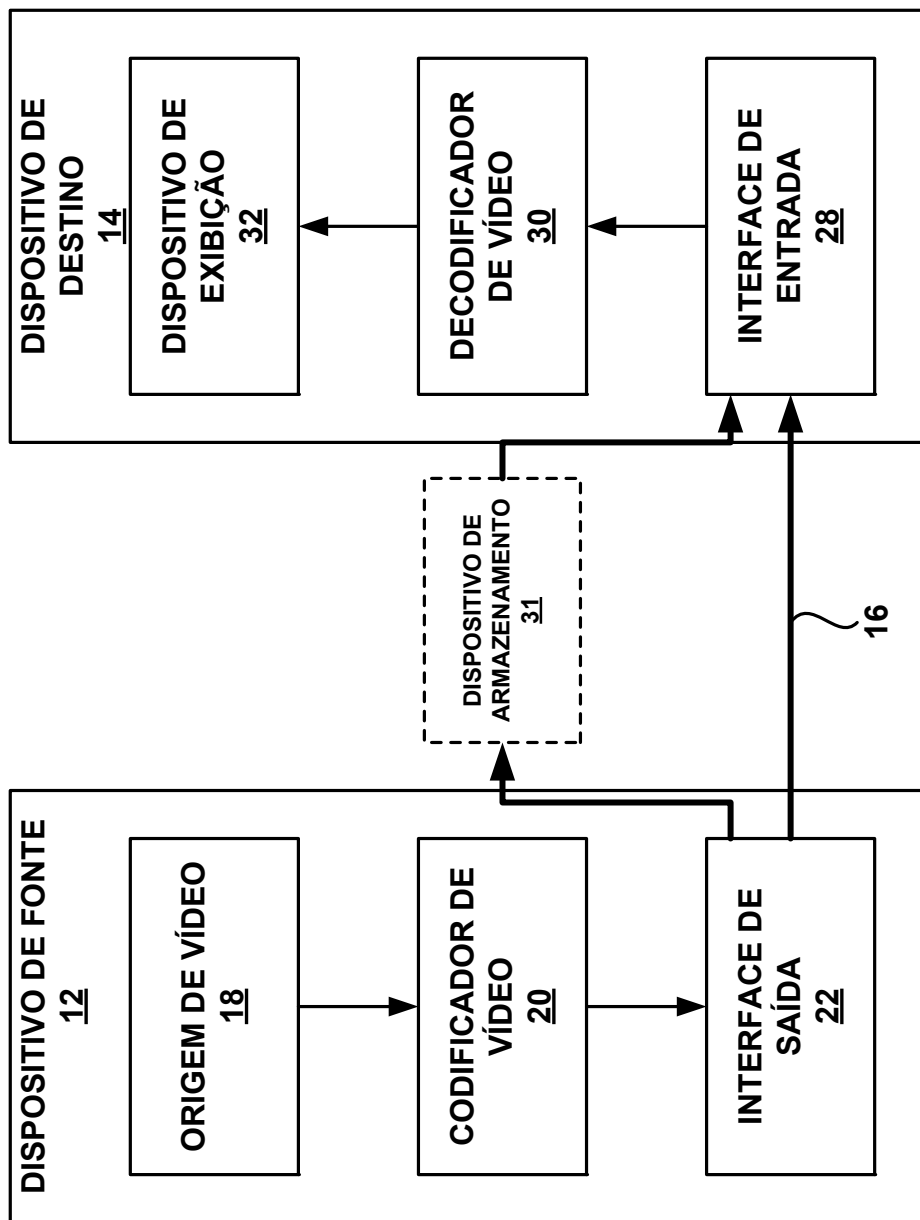


FIG. 1A

10'

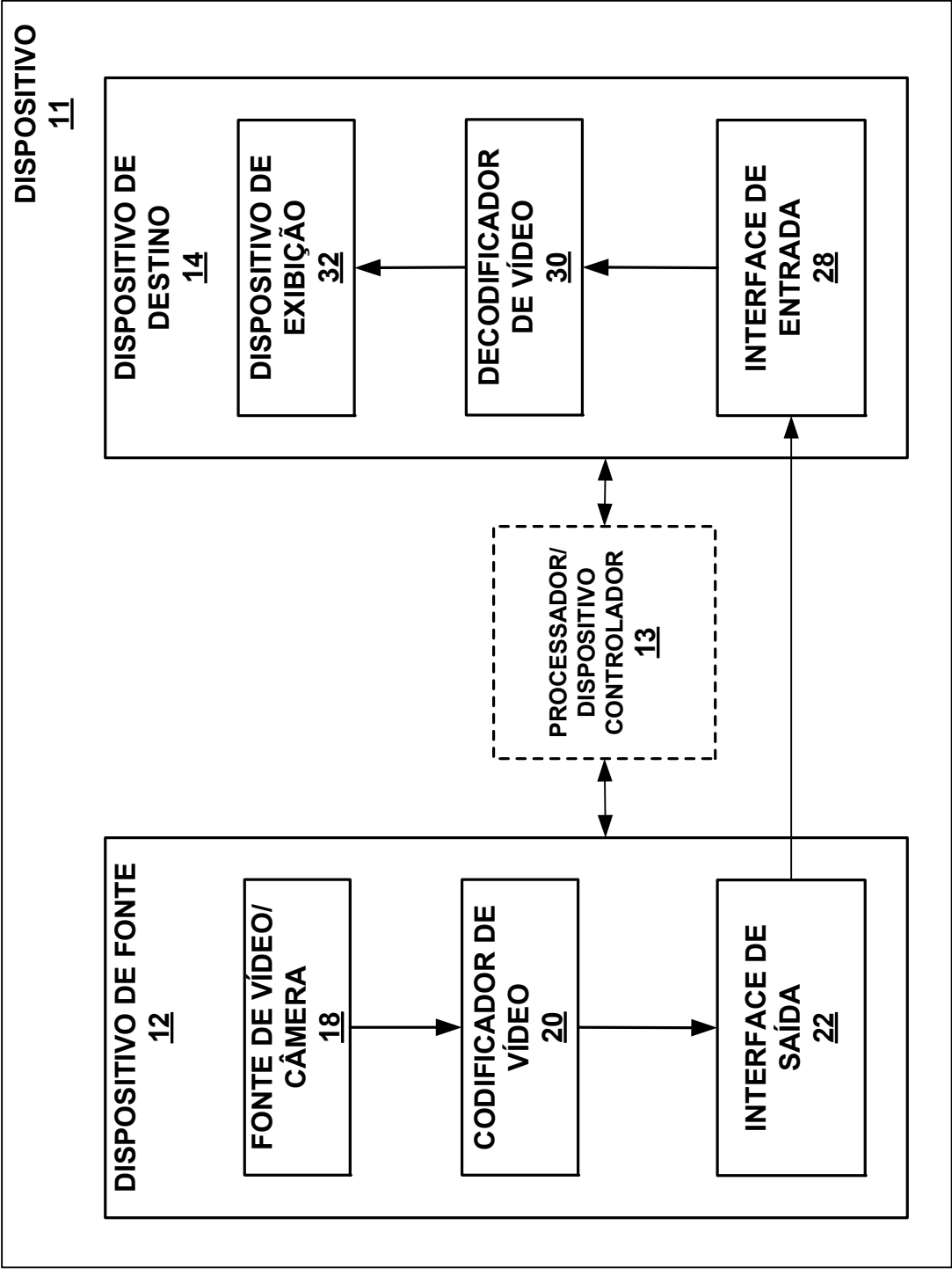
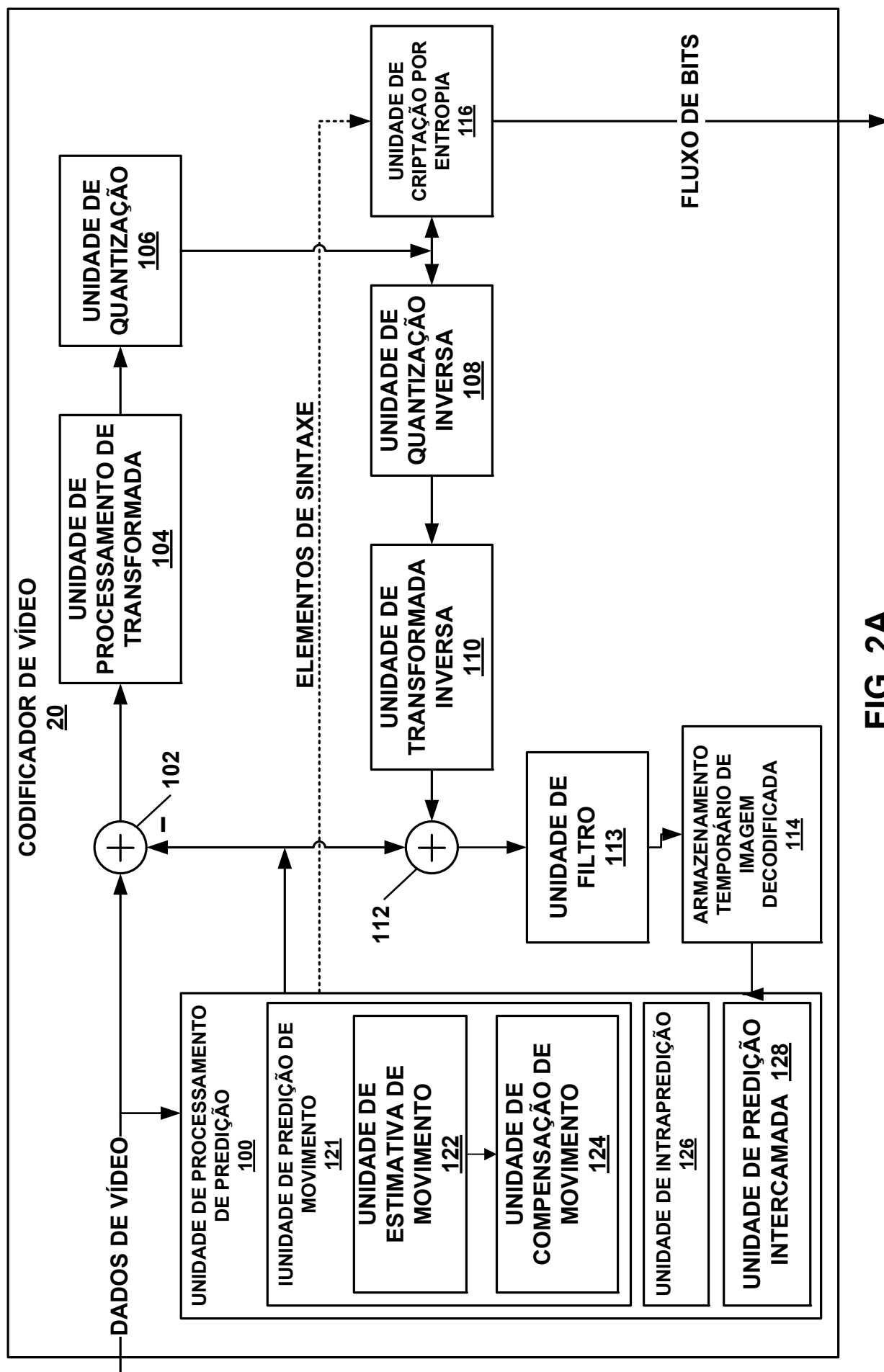


FIG. 1B



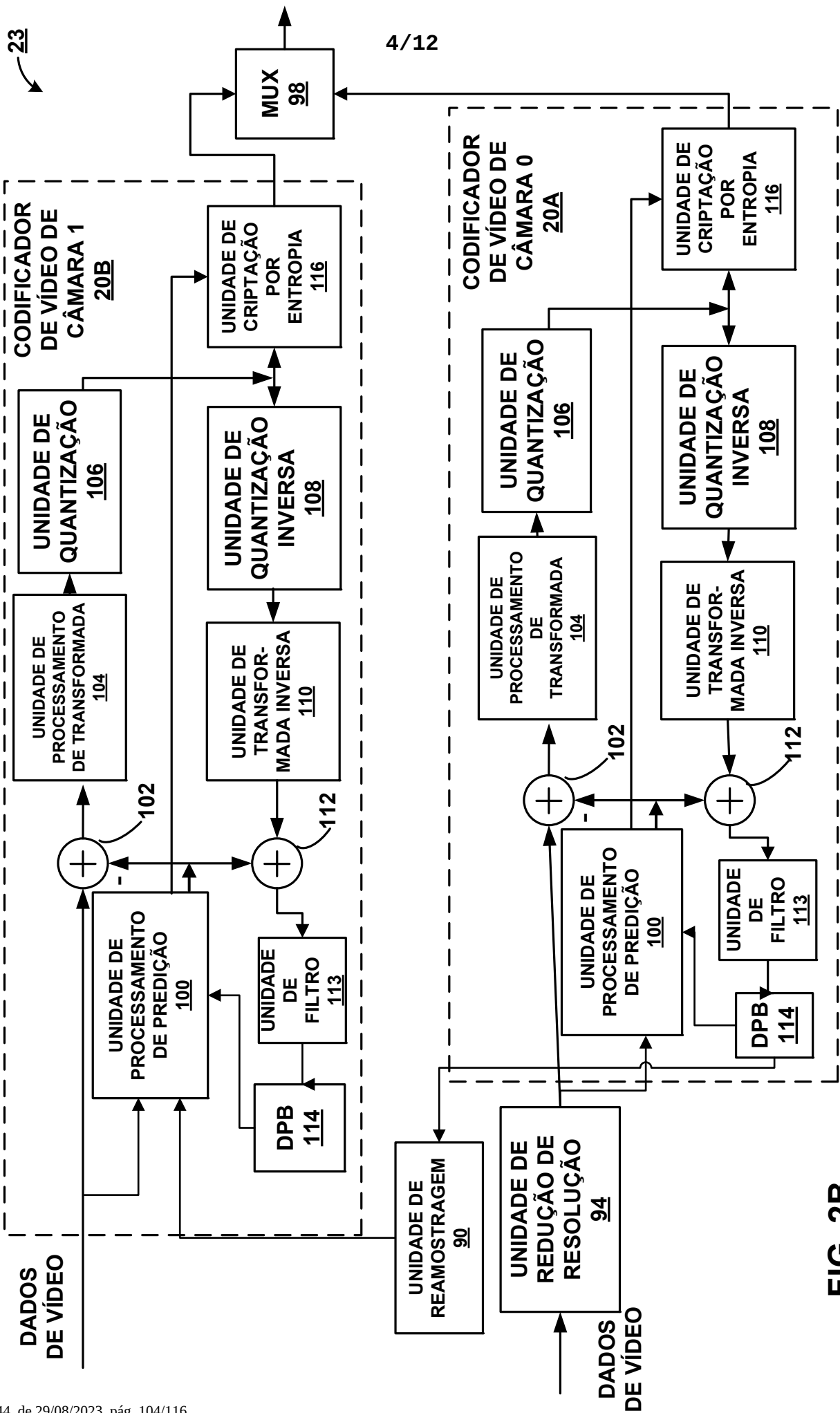


FIG. 2B

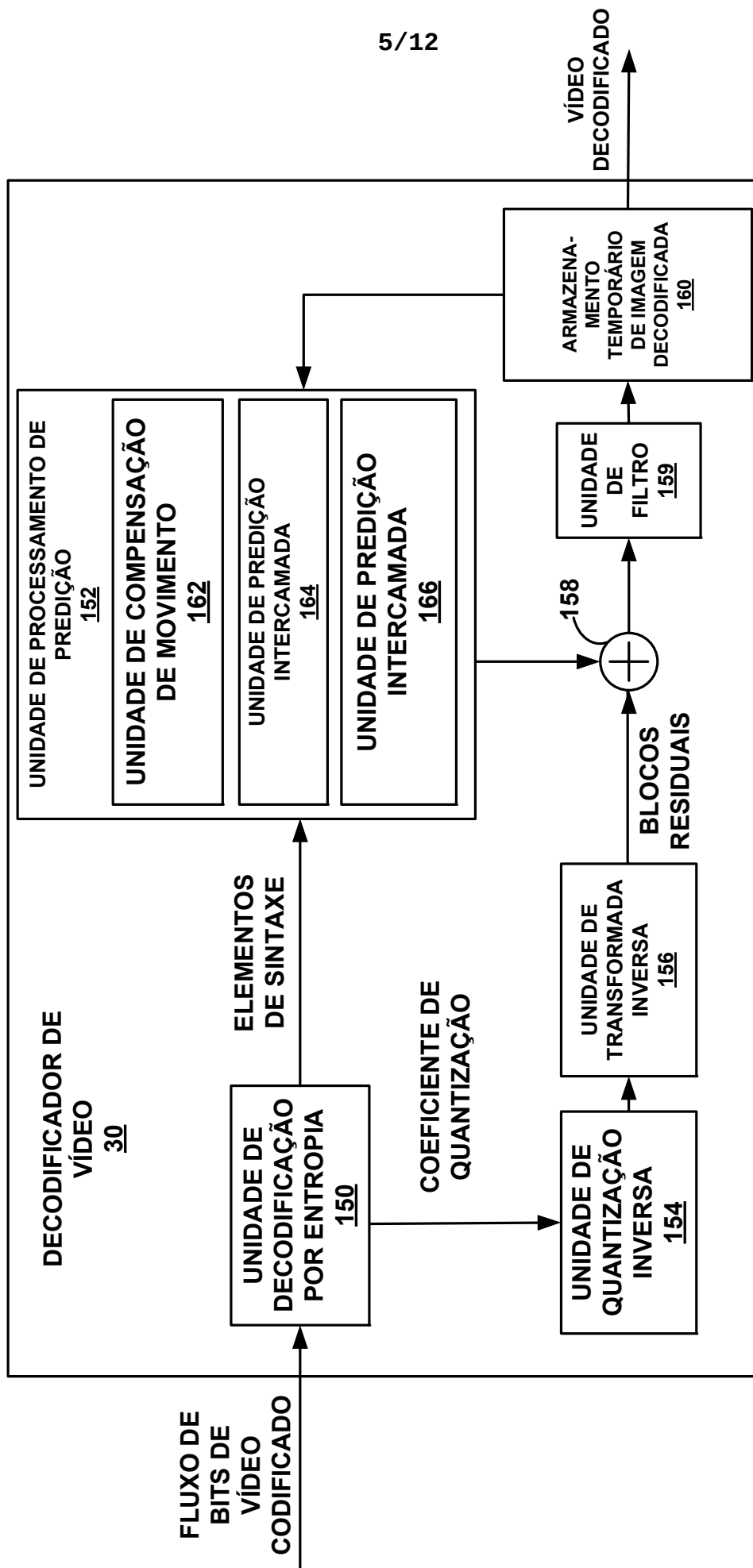


FIG. 3A

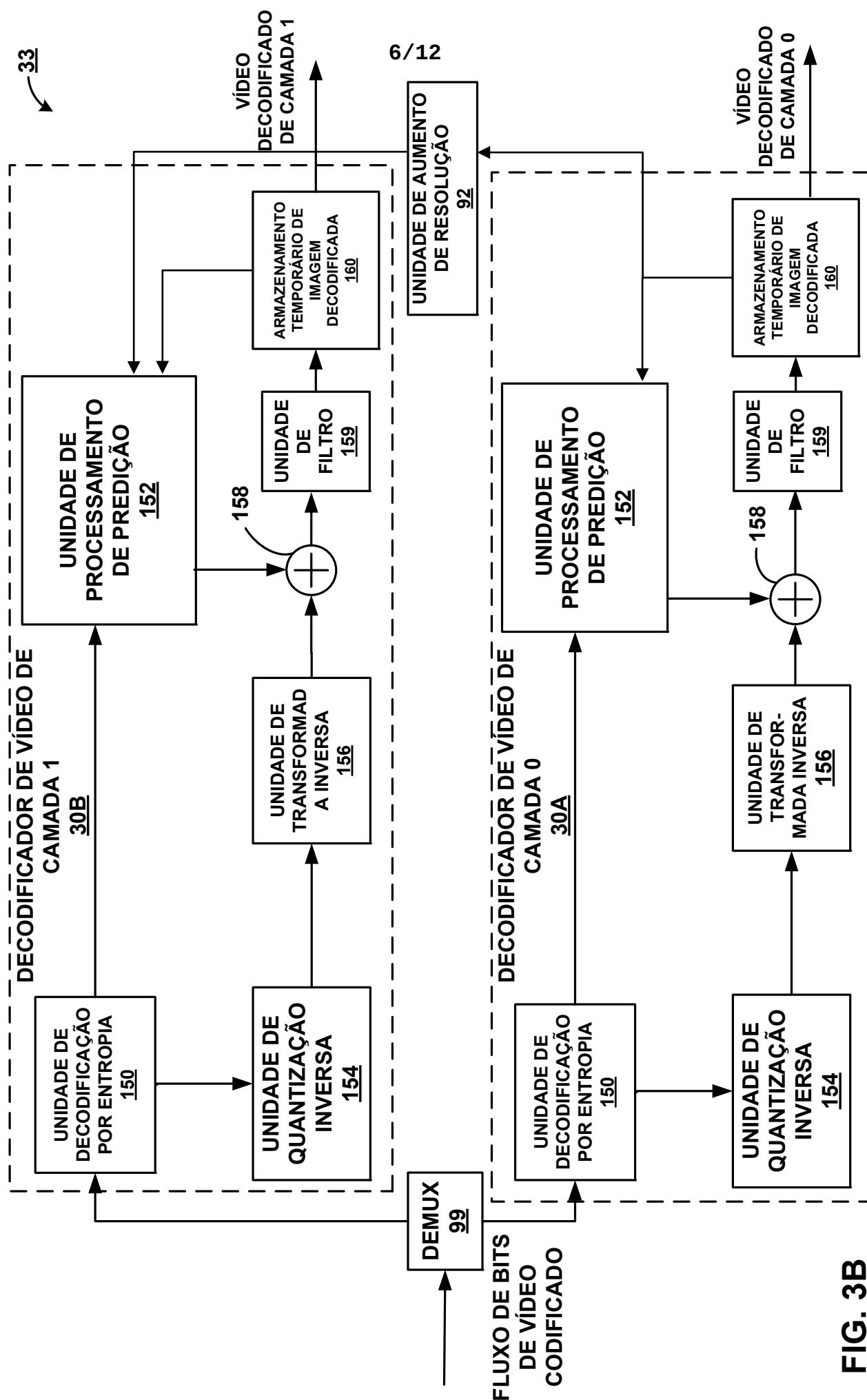


FIG. 3B

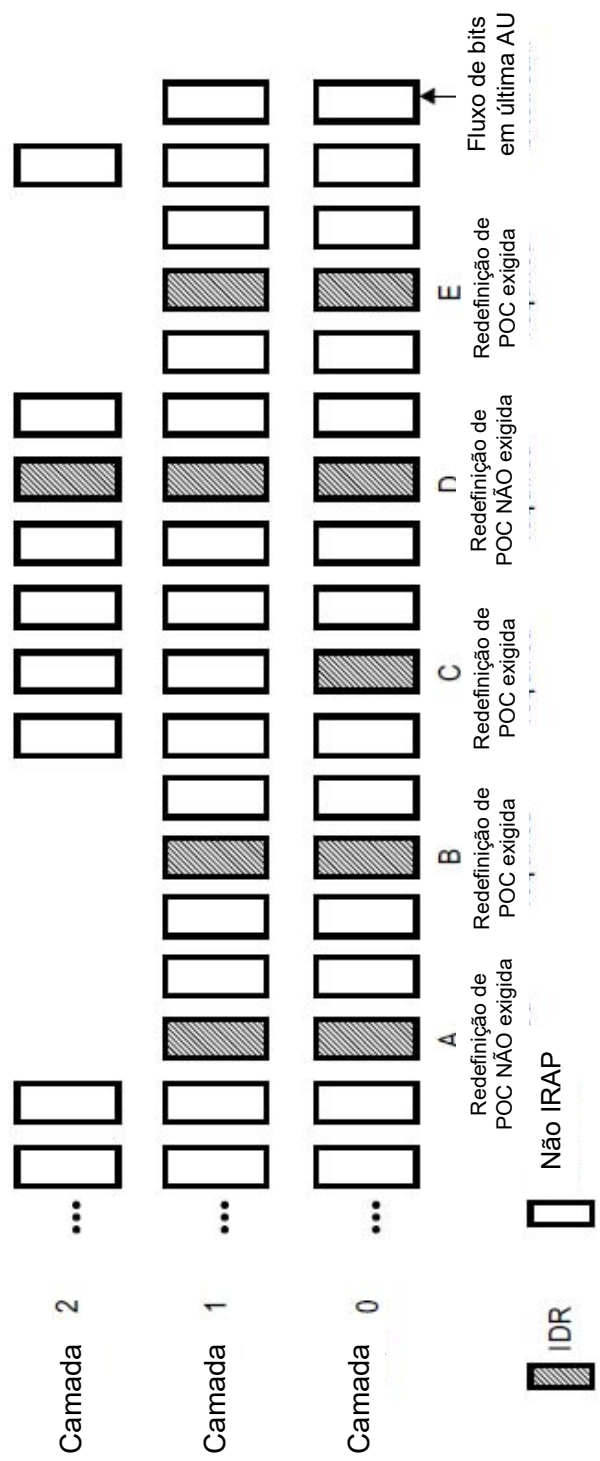


FIG. 4

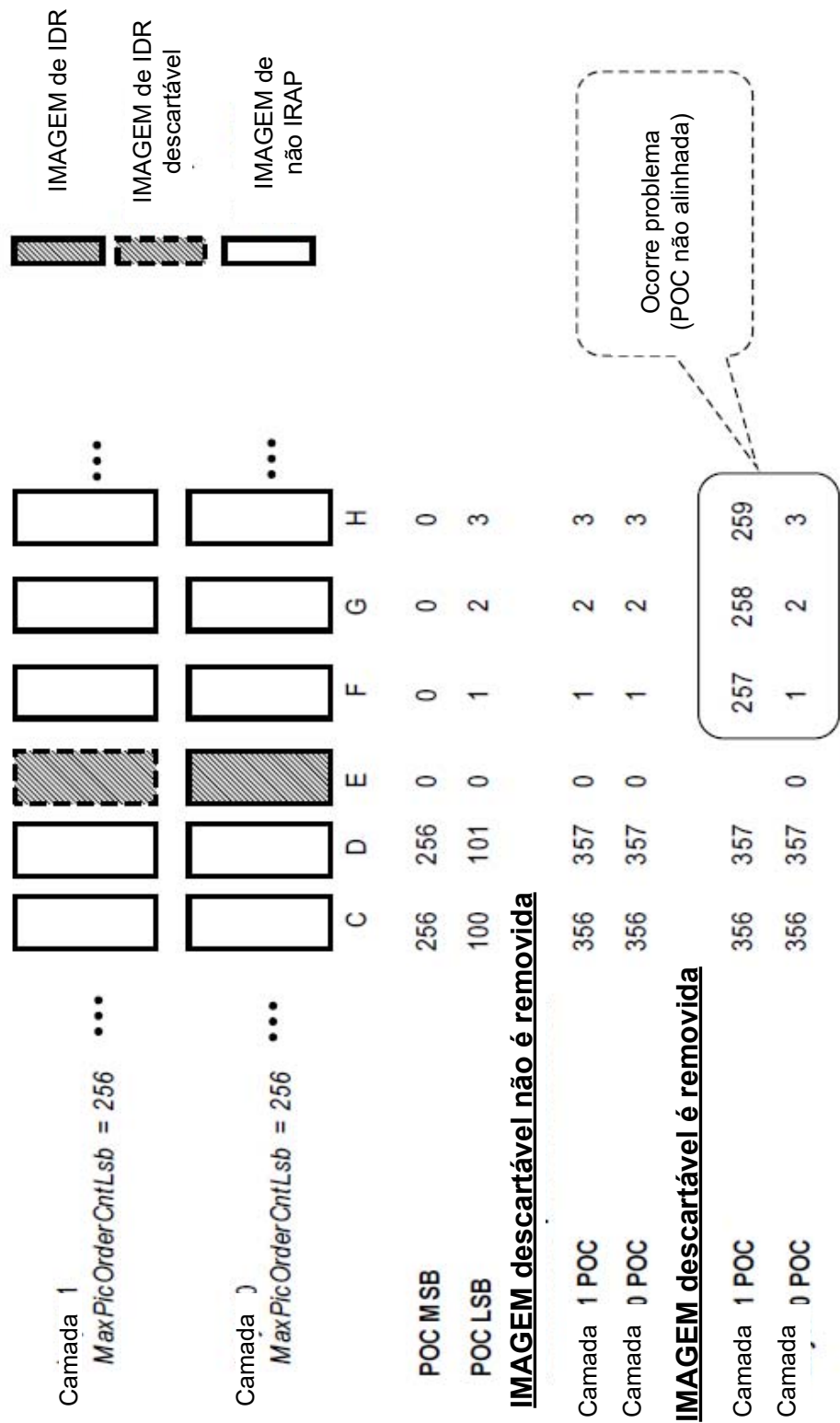


FIG. 5

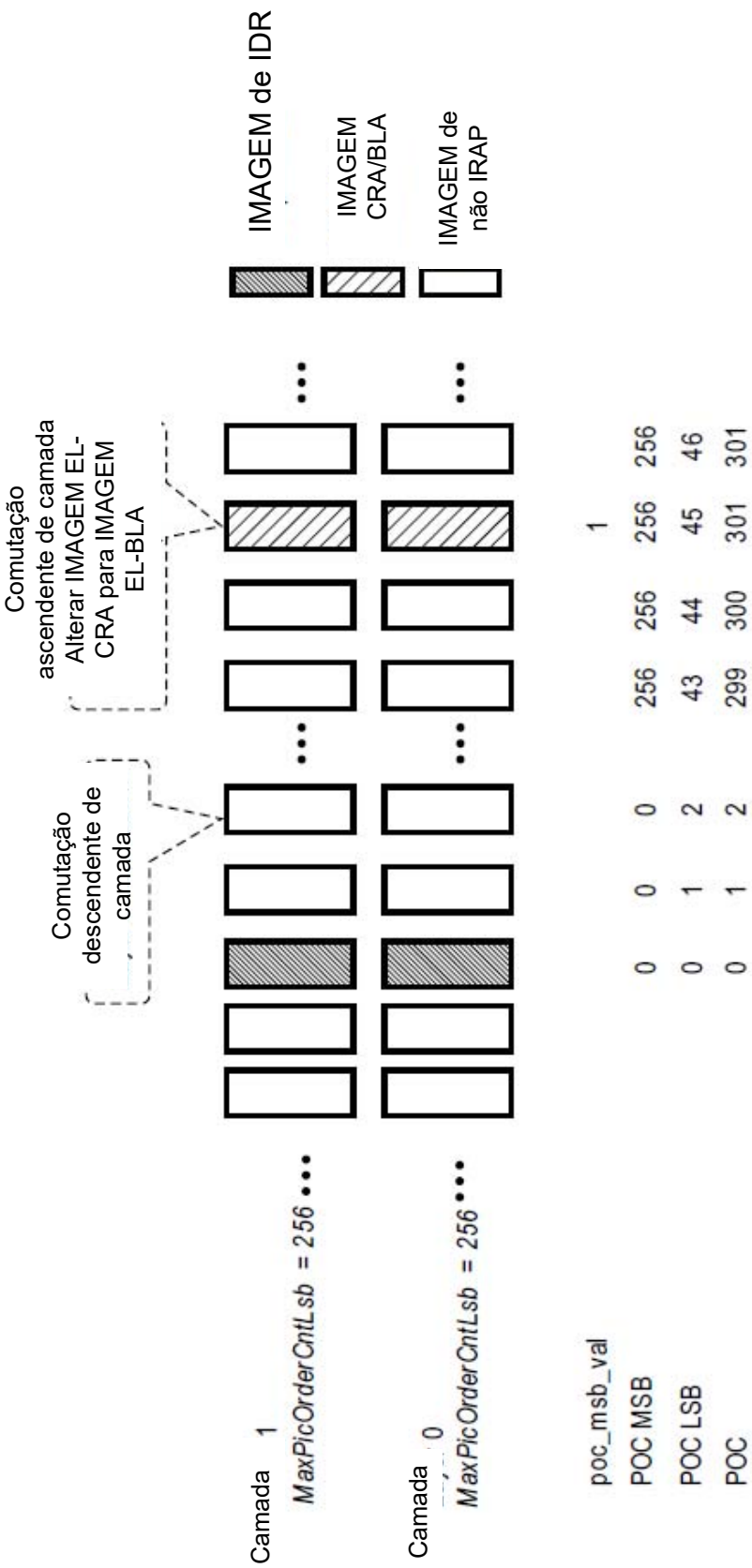
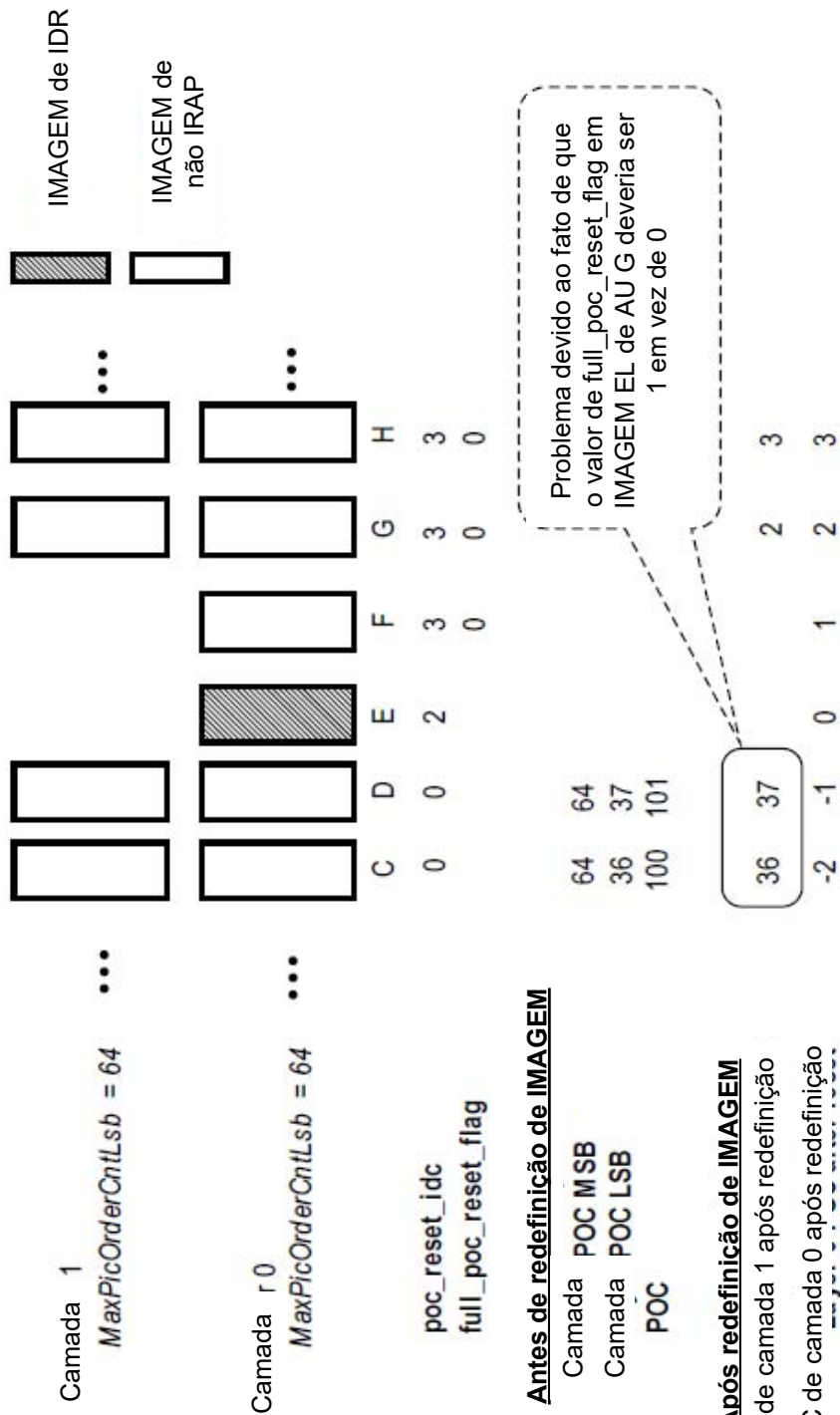


FIG. 6



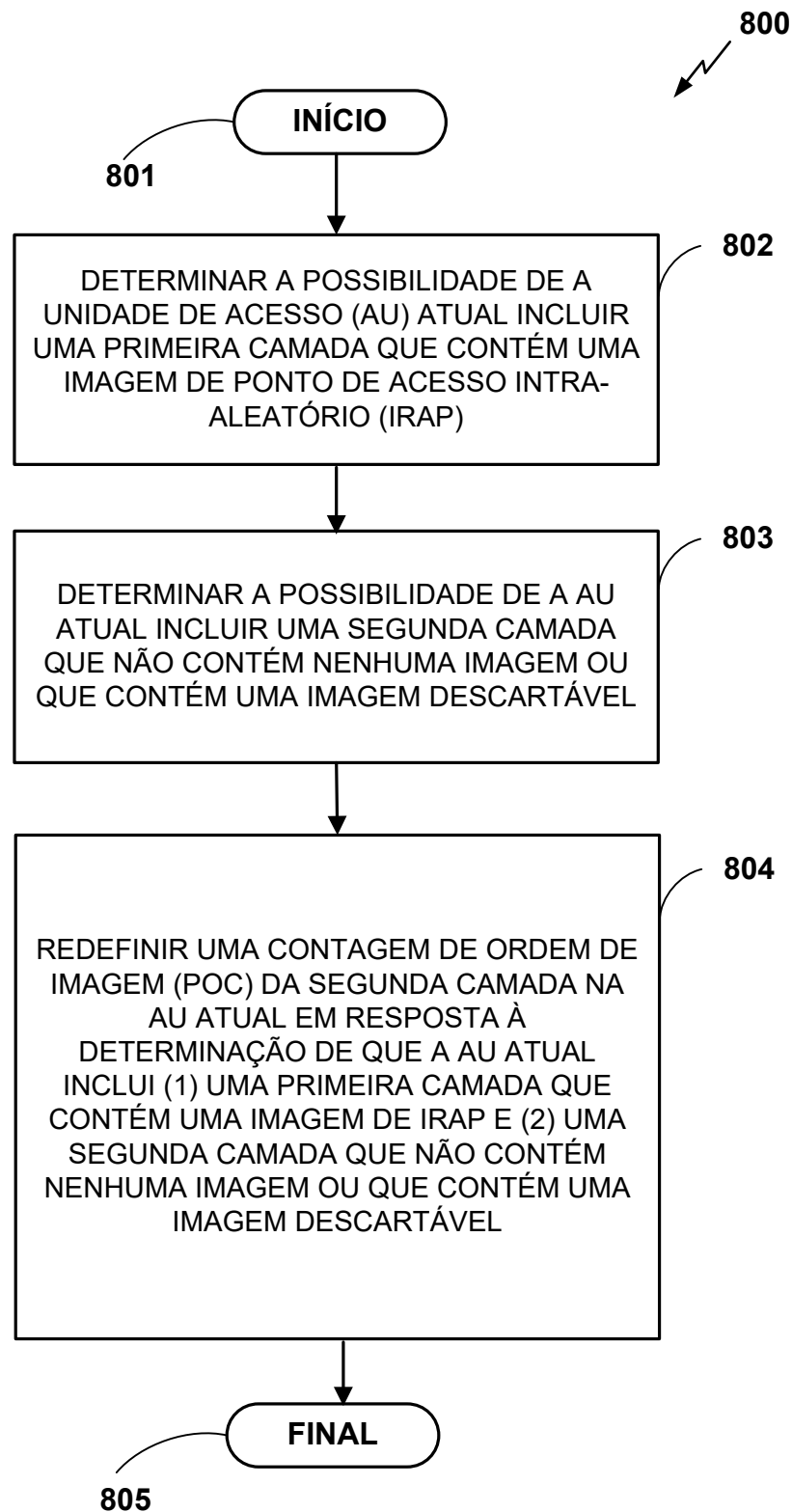


FIG. 8

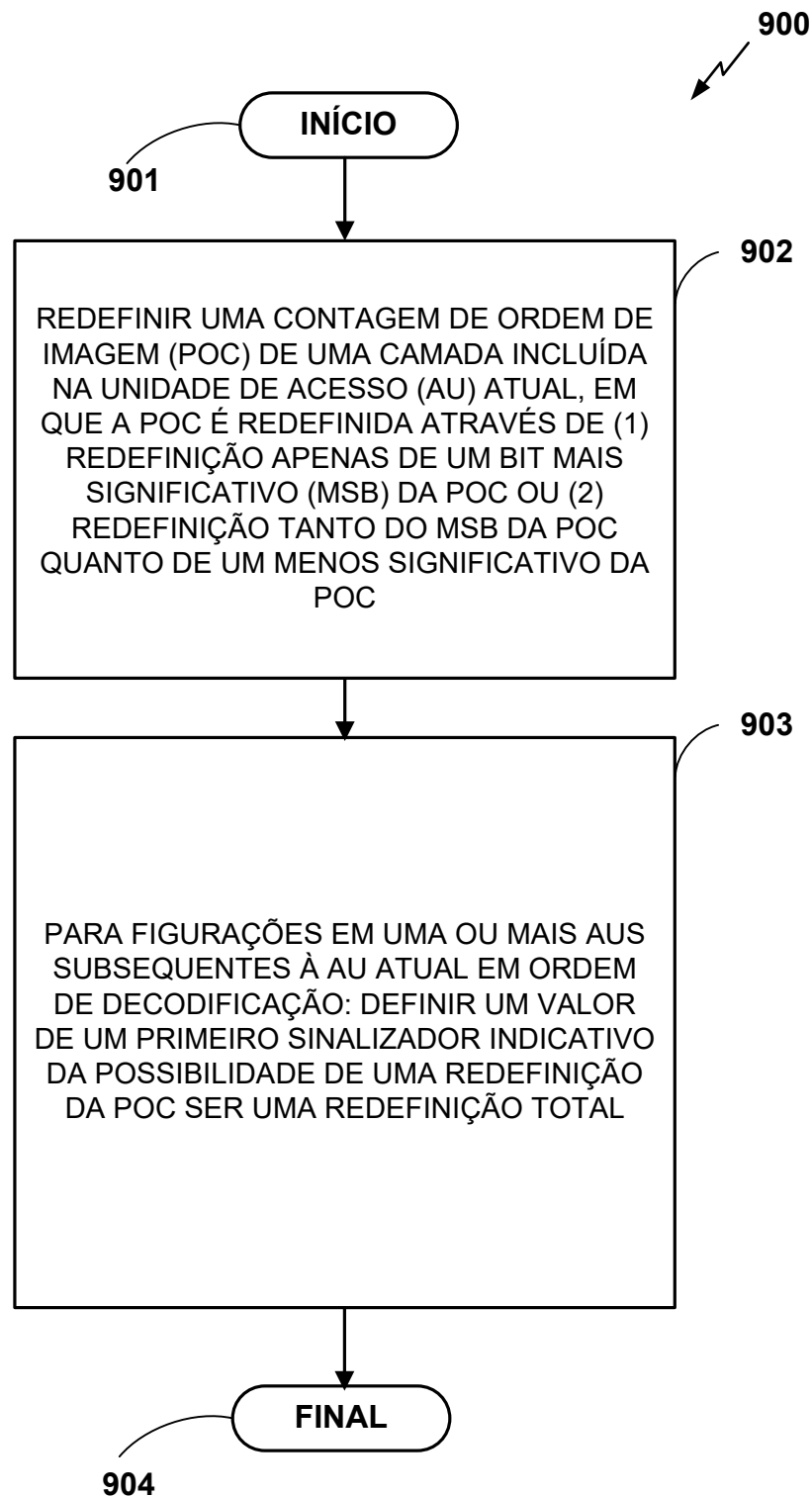


FIG. 9