



* B R 1 1 2 0 1 6 0 1 1 8 1 8 B 1 *

República Federativa do Brasil

Ministério do Desenvolvimento, Indústria,
Comércio e Serviços

Instituto Nacional da Propriedade Industrial

(11) BR 112016011818-9 B1

(22) Data do Depósito: 25/11/2014

(45) Data de Concessão: 16/05/2023

(54) Título: PROJETO DE VALOR DE POC PARA CODIFICAÇÃO DE VÍDEO DE MULTICAMADA

(51) Int.Cl.: H04N 19/70; H04N 19/46; H04N 19/30; H04N 19/68.

(30) Prioridade Unionista: 24/11/2014 US 14/552,129; 25/11/2013 US 61/908,671; 26/11/2013 US 61/909,347.

(73) Titular(es): QUALCOMM INCORPORATED.

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

(86) Pedido PCT: PCT US2014067363 de 25/11/2014

(87) Publicação PCT: WO 2015/077764 de 28/05/2015

(85) Data do Início da Fase Nacional: 24/05/2016

(57) Resumo: PROJETO DE VALOR DE POC PARA CODIFICAÇÃO DE VÍDEO DE MULTICAMADA. Em um exemplo, um método de decodificação de dados de vídeo inclui decodificação de dados que indicam uma redefinição de contagem de ordem de imagem (POC) para um valor de POC de uma primeira imagem de uma primeira camada de dados de vídeo de multicamada, em que a primeira imagem é incluída em uma unidade de acesso. O método exemplar também inclui, com base nos dados que indicam a redefinição de POC para o valor de POC da primeira imagem e antes da decodificação da primeira imagem, diminuir os valores de POC de todas as imagens armazenadas em um armazenador de imagem decodificada (DPB) que precedem a primeira imagem em ordem de codificação, incluindo pelo menos uma imagem de uma segunda camada de dados de vídeo de multicamada.

"PROJETO DE VALOR DE POC PARA CODIFICAÇÃO DE VÍDEO DE
MULTICAMADA"

[0001] Este pedido reivindica o benefício do Pedido Provisório US No. 61/908.671, depositado em 25 de novembro de 2013, e Pedido Provisório U.S. No. 61/909.347, depositado em 26 de novembro de 2013, que são aqui incorporados por referência na sua totalidade.

CAMPO TÉCNICO

[0002] Esta divulgação refere-se à codificação de vídeo.

FUNDAMENTOS

[0003] Capacidades de vídeo digitais podem ser incorporadas em uma ampla gama de dispositivos, incluindo televisores digitais, sistemas de broadcast diretos digitais, sistemas de transmissão sem fio, assistentes digitais pessoais (PDAs), computadores laptop ou desktop, tablets, e-books, câmeras digitais, dispositivos de gravação digital, reprodutores de mídia digital, dispositivos de jogos de vídeo, consoles de jogos de vídeo, telefones celulares de rádio ou satélite, os chamados "smartphones", aparelhos de videoteleconferência, dispositivos de streaming de vídeo, e similares. Dispositivos de vídeo digital implementam as técnicas de codificação de vídeo, tais como as descritas nos padrões definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264 / MPEG-4, Parte 10, Codificação de Vídeo Avançada (AVC), padrão codificação de vídeo de alta eficiência (HEVC) atualmente em desenvolvimento, e a extensão de tais normas. Os dispositivos de vídeo podem transmitir, receber, codificar, decodificar e/ou armazenar informações de vídeo digital de forma mais eficiente através da aplicação de tais técnicas de codificação de vídeo.

[0004] Técnicas de codificação de vídeo incluem predição espacial (intraimagem) e/ou predição temporal (interimagem) para reduzir ou eliminar a redundância inerente em sequências de vídeo. Para a codificação de vídeo baseada em blocos, uma fatia de vídeo (por exemplo, um quadro de vídeo ou uma porção de um quadro de vídeo) pode ser dividida em blocos de vídeo, que podem igualmente ser referidos como blocos em árvore, unidades de codificação (CUs) e/ou nós de codificação. Blocos de vídeo em uma fatia intracodificada (I) de uma imagem são codificados usando a predição espacial em relação às amostras de referência em blocos vizinhos na mesma imagem. Blocos de vídeo em uma fatia intercodificada (P ou B) de uma imagem espacial podem utilizar a predição no que diz respeito às amostras de referência em blocos vizinhos na mesma imagem, ou a predição temporal com respeito às amostras de referência em outras imagens de referência. As imagens podem ser referidas como quadros, e imagens de referência podem ser referidas como uma imagem de referência.

[0005] Predição espacial ou temporal resulta em um bloco preditivo para um bloco de ser codificado. 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 formam o bloco de predição, e os dados residuais que 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 a compressão adicional, os dados residuais podem ser transformados do domínio de pixel para um domínio da transformada, resultando em coeficientes de transformada

residuais, que, em seguida, podem ser quantificados. Os coeficientes de transformada quantificados, inicialmente dispostos em uma matriz bidimensional podem ser verificados, a fim de produzir um vetor unidimensional de coeficientes de transformada, e codificação de entropia pode ser aplicada para conseguir ainda mais compressão.

SUMÁRIO

[0006] De um modo geral, esta divulgação descreve técnicas para melhorar a resiliência de erro para codificar dados de vídeo relacionados com imagens de referência. Em particular, esta divulgação descreve um projeto de valor de contagem de ordem de imagem (POC) que pode melhorar essa resiliência de erros. Estas técnicas podem ser usadas quando da codificação de dados de vídeo de camada única ou de dados de vídeo de multicamada. Em geral, estas técnicas incluem diminuir valores de POC para melhorar a resiliência de erro em um processo de saída de imagem. Por exemplo, um decodificador de vídeo pode ser configurado para diminuir os valores de POC de todas as imagens (de todas as camadas) armazenadas em um armazenador de imagem decodificada (DPB) antes de decodificar uma imagem inicial de uma unidade de acesso em que a redefinição de POC é realizada. Desta maneira, os valores de POC podem permanecer alinhados no DPB, permitindo, assim, que as imagens da unidade de acesso sejam emitidas na ordem adequada.

[0007] Em um exemplo, um método de decodificação de dados de vídeo inclui decodificar dados que indicam uma redefinição de contagem de ordem de imagem (POC) para um valor de POC de uma primeira imagem de uma primeira camada de dados de vídeo de multicamada, em que a primeira imagem é incluída em uma unidade de acesso, e, com base nos dados que indicam a redefinição de POC para o valor de POC da primeira imagem e antes de decodificar a primeira imagem,

diminuir os valores de POC de todas as imagens armazenadas para um armazenador de imagem decodificada (DPB) que precede a primeira imagem, em ordem de codificação, incluindo pelo menos uma imagem de uma segunda camada de dados de vídeo de multicamada.

[0008] Em um outro exemplo, um método de codificação de dados de vídeo inclui codificar dados que indicam uma redefinição de contagem de ordem de imagem (POC) para um valor de POC de uma primeira imagem de uma primeira camada de dados de vídeo de multicamada, em que a primeira imagem é incluída em uma unidade de acesso, e, com base nos dados que indicam a redefinição de POC para o valor de POC da primeira imagem e antes de decodificar a primeira imagem, diminuir os valores de POC de todas as imagens armazenadas em um armazenador de imagem decodificada (DPB) que precede a primeira imagem, em ordem de codificação, incluindo pelo menos uma imagem de uma segunda camada de dados de vídeo de multicamada.

[0009] Em um outro exemplo, um dispositivo para a codificação de dados de vídeo inclui uma memória que compreende um armazenador de imagem decodificada (DPB) configurada para armazenar dados de vídeo de multicamada. O dispositivo também inclui um codificador de vídeo configurado para codificar dados que indicam uma redefinição de contagem de ordem de imagem (POC) para um valor de POC de uma primeira imagem de uma primeira camada dos dados de vídeo de multicamada, em que a primeira imagem é incluída em uma unidade de acesso e, com base nos dados que indicam a redefinição de POC para o valor de POC da primeira imagem e antes de decodificar a primeira imagem, diminuir valores de POC de todas as imagens armazenadas no DPB que precedem a primeira imagem, em ordem de

decodificação incluindo pelo menos uma imagem de uma segunda camada de dados de vídeo de multicamada.

[0010] Em um outro exemplo, um dispositivo para a codificação de dados de vídeo inclui meios para codificação de dados que indicam uma redefinição de contagem de ordem de imagem (POC) para um valor de POC de uma primeira imagem de uma primeira camada dos dados de vídeo de multicamada, em que a primeira imagem está incluída em uma unidade de acesso, e, com base nos dados que indicam a redefinição de POC para o valor de POC da primeira imagem e antes de decodificar a primeira imagem, meios para diminuir os valores de POC de todas as imagens armazenadas no DPB que precedem a primeira imagem, em ordem de codificação, incluindo pelo menos uma imagem de uma segunda camada de dados de vídeo de multicamada.

[0011] Em um outro exemplo, um meio de armazenamento legível por computador não transitório tem armazenadas nele instruções que, quando executadas, fazem com que um processador de um dispositivo para decodificação de dados de vídeo codifique dados que indicam uma redefinição de contagem de ordem de imagem (POC) para um valor de POC de uma primeira imagem de uma primeira camada de dados multicamada de vídeo, em que a primeira imagem é incluída numa unidade de acesso, e, com base nos dados que indicam a redefinição de POC para o valor de POC da primeira imagem e prévia para decodificar a primeira imagem, valores decréscimo POC de todas as imagens armazenadas no DPB que precedem a primeira imagem, em ordem de codificação, incluindo pelo menos uma imagem de uma segunda camada de dados de vídeo de multicamada.

[0012] Os detalhes de um ou mais exemplos são apresentados nos desenhos acompanhantes e na descrição abaixo. Outras características, objetos e vantagens serão

evidentes a partir da descrição e desenhos, e a partir das reivindicações.

BREVE DESCRIÇÃO DOS DESENHOS

[0013] A figura 1 é um diagrama de blocos que ilustra um exemplo de sistema de decodificação e codificação de vídeo que pode utilizar técnicas de codificação de informação de valor de POC de acordo com as técnicas desta divulgação.

[0014] A figura 2 é um diagrama de blocos que ilustra um exemplo de um codificador de vídeo que pode implementar as técnicas para codificar informação para valor de POC de acordo com as técnicas desta divulgação.

[0015] A figura 3 é um diagrama de blocos que ilustra um exemplo de um decodificador de vídeo que pode implementar as técnicas para codificar informação para valor de POC de acordo com as técnicas desta divulgação.

[0016] A figura 4 é um fluxograma que ilustra um exemplo de método para codificar dados incluindo redefinição de valores de POC, de acordo com as técnicas desta divulgação.

[0017] A figura 5 é um fluxograma que ilustra um exemplo de método para a decodificação de dados, incluindo a redefinição de valores de POC, de acordo com as técnicas desta divulgação.

DESCRIÇÃO DETALHADA

[0018] De um modo geral, esta divulgação descreve um projeto de valor de contagem de ordem de imagem (POC) para codificação de vídeo, que pode ser benéfico para codificação de vídeo de multicamada. Esta divulgação descreve várias melhorias no projeto para sinalização e derivação de valores de POC em codificação de vídeo de multicamada. No entanto, enquanto as técnicas são geralmente descritas em relação à codificação de vídeo de

multicamada, as técnicas da presente divulgação podem também ser aplicadas a codificação de vídeo de camada única.

[0019] Os valores de POC podem indicar uma ordem de exibição de imagens e podem geralmente ser usados para identificar a imagem. Por exemplo, uma imagem possuindo um valor de POC de 0 (isto é, uma imagem que tem um valor de POC igual a 0) é apresentada antes de uma imagem com um valor de POC de 1. Quando um bloco de imagem atual é interpretado em relação a uma imagem de referência, a imagem de referência pode ser identificada utilizando um valor de POC para a imagem de referência. Mais particularmente, os valores de POC para imagens de referência a serem incluídos em uma lista de imagem de referência podem ser sinalizados em um conjunto de parâmetros, tais como conjunto de parâmetros de sequência (SPS), conjunto de parâmetros de imagem (PPS), e/ou um cabeçalho de fatia correspondente ao bloco. Deste modo, um codificador de vídeo pode identificar uma imagem de referência sinalizando um índice para a lista de imagem de referência correspondente à posição da imagem de referência na lista de imagem de referência, e um decodificador de vídeo pode identificar a imagem de referência através da construção de uma lista de imagem de referência (com base nos valores de POC) e usar o índice de referência para determinar a posição de uma imagem de referência na lista de imagem de referência.

[0020] Os valores de POC podem ser compostos de bits menos significativos (LSB) e bits mais significativos (MSB). Em vez de aumentar os valores de POC de imagens indefinidamente durante a codificação, um valor de POC de uma imagem pode ser periodicamente redefinido para ser igual a zero. Valores de POC são tipicamente redefinidos para uma imagem Renovação de Decodificador Instantânea

(IDR). Além disso, o codificador de vídeo pode diminuir os valores de POC de imagens que foram tratadas e armazenadas em um armazenador de imagem decodificada (DPB) anterior para a imagem atual relacionada com a redefinição de POC. Os valores de POC de imagens de referência, que foram decrementados podem ter um valor negativo, a fim de manter a ordem de saída apropriada de tais imagens.

[0021] Em codificação de vídeo de multicamada (por exemplo, codificação de vídeo multivista, codificação de vídeo escalonável, ou semelhantes), um DPB pode ser dividido em "sub-DPBS", com cada camada de dados de vídeo atribuída a um sub-DPB gerenciado separadamente do DPB. Um componente de vista pode incluir uma imagem codificada para uma determinada camada ou vista em um determinado momento (em termos de ordem de exibição, ou ordem de saída). Uma unidade de acesso é, em geral, uma unidade de dados, incluindo todos os componentes de vista (por exemplo, todas as unidades de camada de abstração de rede (NAL)) para uma instância temporal comum. Assim, uma unidade de acesso para uma instância temporal particular pode incluir respectivos componentes de vista de respectivos sub-DPBS. Os componentes da vista de uma unidade de acesso são normalmente destinados para serem emitidos em conjunto (ou seja, emissão praticamente simultânea), onde emitir uma imagem geralmente envolve transferência de imagens do DPB (por exemplo, o armazenamento de imagens do DPB em uma memória externa, enviando as imagens do DPB para um display, retirando as imagens do DPB, ou semelhante).

[0022] Um codificador de vídeo pode usar um processo de emissão e bombeamento para emitir de imagens do DPB que são marcadas como "necessárias para emissão" e "sem uso para referência". Em alguns casos, o processo de emissão da imagem pode ser invocado duas vezes para cada

imagem, incluindo (1) antes da decodificação da imagem, mas depois de analisar o cabeçalho de fatia da primeira fatia da imagem, e (2) quando a última unidade de decodificação da imagem é removida do armazenador de imagem codificada. Quando o processo de bombear é invocado, o processo gera todas as imagens que pertencem à unidade de acesso selecionada.

[0023] Projetos de POC existentes (por exemplo, como observado abaixo) podem não ser otimizados. Por exemplo, em alguns casos, diminuir os valores de POC pode ser um processo específico de camada. Assim, os valores de POC de imagens anteriores em um sub-DPB (por exemplo, imagens que tenham sido processadas e armazenadas no sub-DPB anteriormente à imagem a ser codificada) para uma camada não podem ser alinhados em camada cruzada com os valores de POC de imagens anteriores em outros sub-DPBs quando os valores de POC ou bits mais significativos (MSB) de POC de imagens de uma camada são redefinidos durante a decodificação de uma imagem dessa camada, conforme descrito no exemplo abaixo. Alinhamento de camada cruzada, por exemplo, ocorre quando cada imagem de uma unidade de acesso tem o mesmo valor de POC, de modo que as imagens da unidade de acesso são emitidas simultaneamente ou quase simultaneamente. Em projetos de POC existentes, os valores de POC das imagens em camadas mais elevadas (por exemplo, as camadas que têm identificadores de camada relativamente maiores) podem não ser atualizados até depois de decodificar uma imagem de uma camada de base. Atualizar o POC ou MSB POC desta forma pode criar um problema para a emissão da imagem através do processo de colisão, o que requer um alinhamento de camada cruzada de valores de POC para cada unidade de acesso que contém imagens decodificadas no DPB.

[0024] Em um exemplo para fins de ilustração, assume-se que uma sequência multicamada tem três camadas (por exemplo, as camadas A, B e C). Além disso, assume-se que uma unidade de acesso particular inclui imagens para as quais redefinição de POC é realizada. Neste exemplo, um decodificador de vídeo pode redefinir o valor de POC de uma imagem de camada A da unidade de acesso (por exemplo, ao receber uma indicação de um fluxo de bits para executar uma reposição de POC). O decodificador de vídeo também pode diminuir imagens de camada A que foram previamente armazenadas no sub-DPB para a camada A para manter a ordem de saída apropriada. No entanto, ao decodificar a primeira imagem de camada A e antes de decodificar a imagem de camada B da unidade de acesso, os valores de POC de imagens para as camadas B e C que são armazenados nos respectivos sub-DPBs para camadas B e C ficam desalinhados (não alinhados em camada cruzada) com os valores de POC das imagens armazenadas no sub-DPB da camada A. Isto é, porque a imagem de camada B e a imagem da camada C da unidade de acesso ainda não foram redefinidas, os valores de POC das respectivas subcamadas de cada uma das camadas B e C também ainda não foram decrementados. Tal desalinhamento pode fazer com que as imagens sejam impressas a partir do DPB na ordem incorreta, porque as imagens de uma unidade de acesso devem ser alinhadas em multicamada antes de serem emitidas.

[0025] O projeto de valor de POC descrito nesta divulgação inclui diversos aspectos, qualquer um ou todos dos quais podem ser implementados separadamente ou em qualquer combinação. As técnicas da presente divulgação podem superar o problema discutido acima. Em um exemplo, codificadores de vídeo (por exemplo, codificadores de vídeo e decodificadores de vídeo) podem ser configurados para diminuir valores de POC de todas as imagens (de todas as

camadas) armazenadas em um DPB antes de decodificar uma imagem inicial de uma unidade de acesso em que redefinição de POC é realizada. Por exemplo, os codificadores de vídeo podem aplicar um processo diminuição de POC em imagens anteriores, em ordem de codificação em um DPB (ou seja, todas as imagens que antecedem a imagem atual, em ordem de decodificação para todos os sub-DPBs) uma vez, no início de cada unidade de acesso que requer redefinição de POC. Em outras palavras, os codificadores de vídeo podem aplicar o processo de diminuição de POC de uma única vez, no início de cada unidade de acesso que requer redefinição de POC, de tal modo que os valores de POC de todas as imagens que precedem a imagem atual, em ordem de codificação dentro de todos os sub-DPBs são diminuídas com base na única invocação do processo de diminuição de POC. O processo de diminuição de POC pode ser invocada perante a decodificação da primeira imagem recebida para a unidade de acesso, mas após a análise e decodificação da informação de cabeçalho de fatia de primeira fatia dessa imagem. Desta forma, os valores de POC de imagens armazenadas no DPB (ou seja, para todos os sub-DPBs) podem permanecer alinhados ao codificar as imagens da unidade de acesso, permitindo assim que as imagens do DPB sejam emitidas na ordem correta.

[0026] Outras técnicas desta divulgação incluem técnicas de sinalização de valores de POC. Por exemplo, um outro problema potencial com certos modelos de POC é com relação à sinalização de informações adicionais de bit menos significativo (LSB) de POC. Em alguns casos, como descrito em maior detalhe abaixo, o POC LSB de uma imagem de POC-âncora não é sinalizado quando a imagem de POC-âncora de uma camada de base é uma imagem IDR. Não sinalizar o POC LSB em tais casos pode criar um problema, porque uma imagem IDR em uma camada de base não contém

informações para derivar o valor usado para diminuir valores de POC de imagens anteriores (por exemplo, imagens que precedem uma imagem em código, em ordem de codificação) no DPB.

[0027] De acordo com aspectos da presente divulgação, uma condição para a sinalização de POC LSB da imagem de POC-âncora pode ser atualizado. Por exemplo, de acordo com aspectos da presente divulgação, um codificador de vídeo pode ser um sinal (e um decodificador de vídeo pode decodificar) os dados que indicam o POC adicional LSB para imagens IDR de camada de base possuindo um elemento de sintaxe poc_reset_idc que é igual a 2 (como descrito em maior detalhe abaixo).

[0028] A figura 1 é um diagrama de blocos que ilustra um sistema de codificação e decodificação de vídeo exemplar 10 que pode utilizar as técnicas de gerenciamento de valores de POC de acordo com as técnicas desta divulgação. Como mostrado na figura 1, o sistema 10 inclui um dispositivo de origem 12, que provê dados de vídeo codificados para serem decodificados em um momento posterior por um dispositivo de destino 14. Em particular, o dispositivo de origem 12 provê os dados de vídeo para o dispositivo de destino 14 através de um meio legível por computador 16. Dispositivo de origem 12 e dispositivo de destino 14 podem compreender qualquer um de uma ampla gama de dispositivos, incluindo computadores desktop, computadores notebook (ou seja, laptop), tablets, set-top boxes, aparelhos de telefone, tais como os chamados "smartphones", as assim chamadas "smartpads", televisores, câmeras, dispositivos de exibição, reprodutores de mídia digital, consoles de jogos de vídeo, dispositivos de streaming de vídeo, ou algo semelhante. Em alguns casos, o

dispositivo de origem 12 e o dispositivo de destino 14 pode ser equipado para comunicações sem fio.

[0029] Dispositivo de destino 14 pode receber os dados de vídeo codificados para serem decodificados via o meio legível por computador 16. O meio legível por computador 16 pode compreender qualquer tipo de meio ou dispositivo capaz de mover os dados de vídeo codificados do dispositivo de origem 12 para o dispositivo de destino 14. Em um exemplo, meio legível por computador 16 pode compreender um meio de comunicação para permitir que dispositivo de origem 12 transmita dados de vídeo codificados diretamente para o dispositivo de destino 14 em tempo real. Os dados de vídeo codificados podem ser modulados de acordo com um padrão de comunicação, tal como um protocolo de comunicação sem fio, e transmitidos para o dispositivo de destino 14. O meio de comunicação pode compreender qualquer meio de comunicação com ou sem fio, tal como um espectro de rádio frequência (RF) ou uma ou mais linhas de transmissão física. O meio de comunicação pode formar parte de uma rede baseada em pacotes, tal como uma rede de área local, uma rede de área ampla, ou uma rede global tal como a Internet. O meio de comunicação pode incluir roteadores, comutadores, estações base, ou qualquer outro equipamento que possa ser útil para facilitar a comunicação de dispositivo de origem 12 para o dispositivo de destino 14.

[0030] Em alguns exemplos, os dados codificados podem ser produzidos a partir da interface de saída 22 para um dispositivo de armazenamento. Da mesma forma, os dados codificados podem ser acessados a partir do dispositivo de armazenamento, interface de entrada. O dispositivo de armazenamento pode incluir qualquer de uma variedade de meios de armazenamento de dados distribuídos ou localmente

acessados, como um disco rígido, os discos Blu-ray, DVDs, CD-ROMs, memória flash, memória volátil ou não volátil, ou qualquer outro meio de armazenamento digital adequado para armazenar dados de vídeo codificados. Em um outro exemplo, o dispositivo de armazenamento pode corresponder a um servidor de arquivos ou outro dispositivo de armazenamento intermediário que pode armazenar o vídeo codificado gerado pelo dispositivo de origem 12. O dispositivo de destino 14 pode acessar os dados de vídeo armazenados a partir do dispositivo de armazenamento através de fluxo contínuo ou transferência. O servidor de arquivos pode ser qualquer tipo de servidor capaz de armazenar dados de vídeo e transmitir aqueles dados de vídeo codificados para o dispositivo de destino 14. Servidores de arquivos exemplares incluem um servidor de web (por exemplo, para um site), um servidor de FTP, dispositivos de armazenamento anexados à rede (NAS), ou uma unidade de disco local. O dispositivo de destino 14 pode acessar os dados de vídeo codificados através de qualquer conexão de dados padrão, incluindo acesso à Internet. Isto pode incluir um canal sem fio (por exemplo, uma conexão Wi-Fi), uma conexão cabeada (por exemplo, DSL, um modem por cabo, etc.), ou uma combinação de ambos, que é adequada para acessar os dados de vídeo codificados armazenados em um servidor de arquivos. A transmissão de dados de vídeo codificados a partir do dispositivo de armazenamento pode ser uma transmissão ao vivo, uma transmissão de download, ou uma combinação dos mesmos.

[0031] As técnicas da presente divulgação não são necessariamente limitadas a aplicações sem fio ou configurações. As técnicas podem ser aplicadas a codificação de vídeo em apoio a qualquer uma de uma variedade de aplicações multimídia, como transmissões de

televisão através do ar, transmissões de televisão por cabo, transmissão de televisão por satélite, transmissões de streaming de vídeo por Internet, como streaming dinâmico adaptativo através de HTTP (DASH), vídeo digital que é codificado sobre um meio de armazenamento de dados, a decodificação de vídeo digital armazenado em um meio de armazenamento de dados, ou outras aplicações. Em alguns exemplos, o sistema 10 pode ser configurado para suportar transmissão de vídeo de uma via ou duas vias para suportar aplicativos como streaming de vídeo, reprodução de vídeo, transmissão de vídeo, e/ou vídeo telefonia.

[0032] No exemplo da figura 1, dispositivo de origem 12 inclui fonte de vídeo 18, codificador de vídeo 20, e interface de saída 22. O dispositivo de destino 14 inclui interface de entrada 28, decodificador de vídeo 30 e dispositivo de exibição 32. De acordo com esta divulgação, codificador de vídeo 20 do dispositivo de origem 12 pode ser configurado para aplicar as técnicas de codificação de informação de valor de POC de acordo com as técnicas desta divulgação. Em outros exemplos, um dispositivo de origem e um dispositivo de destino podem incluir outros componentes ou arranjos. Por exemplo, dispositivo de origem 12 pode receber dados de vídeo de uma fonte de vídeo externa 18, como uma câmara externa. Da mesma forma, o dispositivo de destino 14 pode interagir com um dispositivo externo, em vez de incluir um dispositivo de visualização integrada.

[0033] O sistema ilustrado 10 da figura 1 é meramente um exemplo. As técnicas para o gerenciamento de valores de POC de acordo com as técnicas da presente divulgação podem ser realizadas por qualquer dispositivo de codificação e/ou decodificação de vídeo digital. Embora geralmente as técnicas da presente invenção sejam realizadas por um dispositivo de codificação de vídeo, as

técnicas também podem ser realizadas por um codificador de vídeo / decodificador, tipicamente referidas como um "codec". Além disso, as técnicas da presente divulgação podem também ser realizadas por um pré-processador de vídeo. Dispositivo de origem 12 e o dispositivo de destino 14 são meramente exemplos de tais dispositivos de codificação nos quais o dispositivo de origem 12 gera dados de vídeo codificados para a transmissão para o dispositivo de destino 14. Em alguns exemplos, os dispositivos 12, 14 podem operar de um modo substancialmente simétrico de tal modo que cada um dos dispositivos 12, 14 incluem codificação de vídeo e componentes de decodificação. Assim, o sistema 10 pode suportar transmissão de vídeo unidirecional ou bidirecional entre dispositivos de vídeo 12, 14, por exemplo, para streaming de vídeo, reprodução de vídeo, transmissão de vídeo, ou telefonia de vídeo.

[0034] A fonte de vídeo 18 do dispositivo de origem 12 pode incluir um dispositivo de captura de vídeo, tal como uma câmera de vídeo, um arquivo de vídeo contendo vídeo capturado anteriormente, e/ou uma interface de alimentação de vídeo para receber vídeo a partir de um provedor de conteúdo de vídeo. Como uma outra alternativa, a fonte de vídeo 18 pode gerar dados baseados em computação gráfica como o vídeo de origem, ou uma combinação de vídeo ao vivo, vídeo arquivado e vídeo gerado por computador. Em alguns casos, se a fonte de vídeo 18 é uma câmera de vídeo, dispositivo de origem 12 e dispositivo de destino 14 podem formar os chamados telefones com câmera ou telefones de vídeo. Como mencionado acima, no entanto, as técnicas descritas nesta divulgação podem ser aplicáveis à codificação de vídeo, em geral, e podem ser aplicadas para aplicações sem fio e/ou com fio. Em cada caso, o vídeo capturado, pré-capturado, ou gerado pelo computador pode

ser codificado por um codificador de vídeo 20. A informação de vídeo codificado pode então ser emitida pela interface de saída 22 para um meio legível por computador 16.

[0035] Meio legível por computador 16 pode incluir meios transientes, tais como um broadcast sem fio ou transmissão de rede com fio ou mídia de armazenamento (isto é, mídia de armazenamento não transitória), tais como um disco rígido, unidade flash, disco compacto, disco de vídeo digital, disco Blu-ray, ou outra mídia legível por computador. Em alguns exemplos, um servidor de rede (não mostrado) pode receber dados de vídeo codificado a partir do dispositivo de origem 12 e prover os dados de vídeo codificados para o dispositivo de destino 14, por exemplo, por meio de transmissão de rede. Da mesma forma, um dispositivo de computação de uma instalação de produção de meios, tal como uma instalação de disco de estampagem, pode receber dados de vídeo codificado a partir do dispositivo de origem 12 e produzir um disco contendo os dados de vídeo codificado. Portanto, o meio legível por computador 16 pode ser entendido como incluindo um ou mais meios legíveis por computador de várias formas, em vários exemplos.

[0036] Interface de entrada 28 do dispositivo de destino 14 recebe informações de meio legível por computador 16. As informações de meio legível por computador 16 podem incluir informações de sintaxe definida pelo codificador de vídeo 20, que também é usada pelo decodificador de vídeo 30, que inclui elementos de sintaxe que descrevem características e/ou processamento de blocos e outras unidades codificadas, por exemplo, GOPs. O dispositivo de exibição 32 exibe os dados de vídeo decodificados a um usuário, e pode compreender qualquer um de uma variedade de dispositivos de exibição, tais como um tubo de raios catódicos (CRT), uma tela de cristal líquido

(LCD), uma tela de plasma, uma tela de diodo emissor de luz orgânica (OLED) ou outro tipo de dispositivo de exibição.

[0037] O codificador de vídeo 20 e decodificador de vídeo 30 cada um pode ser implementado como qualquer um de uma variedade de conjunto de circuitos de codificador adequado, tal como um ou mais microprocessadores, processadores de sinais digitais (DSPs), circuitos integrados de aplicação específica (ASIC), arranjos de porta programáveis em campo (FPGAs), lógica discreta, software, hardware, firmware ou quaisquer suas combinações. Quando as técnicas são implementadas parcialmente em software, um dispositivo pode armazenar instruções para o software em um meio legível por computador adequado, não transitório e executar as instruções em hardware utilizando um ou mais processadores para executar as técnicas desta divulgação. Cada um do codificador de vídeo 20 e do decodificador de vídeo 30 pode ser incluído em um ou mais codificadores ou decodificadores, cada um dos quais pode ser integrado como parte de um codificador / decodificador combinado (codec) de um respectivo dispositivo. Um dispositivo, incluindo um codificador de vídeo 20 e/ou decodificador de vídeo 30 pode compreender um circuito integrado, um microprocessador, e/ou um dispositivo de comunicação sem fio, tal como um telefone celular.

[0038] Embora não mostrado na figura 1, em alguns aspectos, codificador de vídeo 20 e decodificador de vídeo 30 pode cada um ser integrado com um codificador de áudio e decodificador, e pode incluir unidades apropriadas MUX-DEMUX, ou outro hardware e software, para lidar com a codificação de áudio e vídeo em um fluxo de dados comum ou fluxos de dados separados. Se aplicável, as unidades MUX-DEMUX podem estar de acordo com o protocolo multiplexador

ITU H.223, ou outros protocolos, tais como o protocolo de Datagrama de Usuário (UDP).

[0039] Esta divulgação pode geralmente referir-se a um codificador de vídeo 20 "sinalizando" certa informação para outro dispositivo, como decodificador de vídeo 30. Deve ser entendido, no entanto, que o codificador de vídeo 20 pode sinalizar informação associando certos elementos de sintaxe com várias porções codificadas de dados de vídeo. Ou seja, codificador de vídeo 20 pode "sinalizar" dados ao armazenar certos elementos de sintaxe em cabeçalhos de várias porções codificadas de dados de vídeo. Em alguns casos, tais elementos de sintaxe podem ser codificados e armazenados antes de ser recebidos e decodificados pelo decodificador de vídeo 30. Deste modo, o termo "sinalização" pode geralmente se referir à comunicação de sintaxe ou outros dados para decodificar os dados de vídeo comprimidos, se tal comunicação ocorre em tempo quase real ou real ou ao longo de um período de tempo, tal como pode ocorrer quando o armazenamento de elementos de sintaxe de um meio no momento da codificação, que pode então ser recuperado por um dispositivo de decodificação, em qualquer momento após ter sido armazenado aneste meio.

[0040] Codificador de vídeo 20 e decodificador de vídeo 30 podem operar de acordo com um padrão de codificação de vídeo. Padrões de codificação de vídeo exemplar incluem ITU-T H.261, ISO / IEC MPEG-1 Visual, ITU-T H.262 ou ISO / IEC MPEG-2 Visual, ITU-T H.263, ISO / IEC MPEG-4 Visual e ITU-T H.264 (também conhecido como ISO / IEC MPEG-4 AVC), incluindo as suas extensões Codificação de Vídeo Escalonável (SVC) e Codificação de Vídeo Multivista (MVC). Recentemente, o projeto de um novo padrão de codificação de vídeo, ou seja, Codificação de Vídeo de Alta Eficiência (HEVC), foi finalizado pela Joint Collaboration

Team on Video Coding (JCT-VC) de ITU-T Video Coding Experts Group (VCEG) e ISO/IEC Motion Picture Experts Group (MPEG). A última especificação do projeto HEVC, e referida como HEVC WD doravante, está disponível em http://phenix.int-evry.fr/jct/doc_end_user/documents/15_Geneva/wg11/JCTVC-01003-v1.zip. A extensão multivista para HEVC, ou seja, MV-HEVC, também está sendo desenvolvida pelo JCT-3V. Um recente Projeto de Trabalho (WD) de MV-HEVC, conhecido como MV-HEVC WD6 adiante, está disponível em http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/6_Geneva/wg11/JCT3V-F1004-v3.zip. A extensão escalável para HEVC, chamada SHVC, também está sendo desenvolvida por JCT-VC. Um recente Projeto de Trabalho (WD) de SHVC e referido como SHVC WD4 doravante, está disponível em http://phenix.int-evry.fr/jct/doc_end_user/documents/15_Geneva/wg11/JCTVC-01008-v1.zip. As técnicas da presente divulgação, no entanto, não estão limitadas a qualquer padrão de codificação particular. Outros exemplos de padrões de codificação de vídeo incluem MPEG-2 e ITU-T H.263.

[0041] Os esforços de normalização de HEVC baseiam-se em um modelo de um dispositivo de codificação de vídeo referido como o modelo de teste HEVC (HM). O HM presume vários recursos adicionais de dispositivos de codificação de vídeo em relação aos dispositivos existentes de acordo com, por exemplo, ITU-T H.264 / AVC. Por exemplo, enquanto H.264 provê nove modos de codificação intrapredição, o HM pode prover até trinta e três modos de codificação intrapredição.

[0042] De um modo geral, o modelo de trabalho do HM descreve que um quadro de vídeo ou a imagem pode ser dividido em uma sequência de blocos em árvore ou unidades de codificação maiores (LCU) que incluem ambas as amostras de luminância e crominância. Sintaxe de dados dentro de um

fluxo de dados pode definir um tamanho para a LCU, que é uma unidade de codificação maior em termos de número de pixels. Uma fatia inclui um número de blocos em árvore consecutivos na ordem de codificação. Um quadro de vídeo ou imagem pode ser dividido em uma ou mais fatias. Cada bloco em árvore pode ser dividido em unidades de codificação (CUs) de acordo com um quadtree. Em geral, uma estrutura de dados inclui um nó quadtree por CU, com um nó raiz correspondente ao bloco em árvore. Se uma CU é dividida em quatro sub-CUs, o nó correspondente a CU inclui quatro nós de folhas, cada um dos quais corresponde a um dos sub-CUs.

[0043] Cada nó da estrutura de dados quadtree pode prover dados de sintaxe para a CU correspondente. Por exemplo, um nó no quadtree pode incluir um flag de divisão, que indica se a CU correspondente ao nó é dividida em sub-CUs. Elementos de sintaxe para uma CU podem ser definidos recursivamente, e podem depender do fato de a CU ser dividida em sub-CUs. Se uma CU não é adicionalmente dividida, ela é referida como uma CU folha. Nesta divulgação, quatro sub-CUs de uma CU folha também serão referidas como CUs folha mesmo se não houver divisão explícita da CU folha original. Por exemplo, se uma CU no tamanho de 16x16 não é dividida ainda mais, as quatro sub-CUs 8x8 irão também ser referidas como CUs folha embora a CU 16x16 nunca tenha sido dividida.

[0044] Uma CU tem um efeito semelhante ao de um macrobloco da norma H.264, exceto que uma CU não tem uma distinção de tamanho. Por exemplo, um bloco em árvore pode ser dividido em quatro nós filho (também referidos como sub-CUs), e cada nó filho por sua vez pode ser um nó pai e ser dividido em mais quatro nós filho. Um nó filho definitivo, indivisível, referido como um nó de folha do quadtree, compreende um nó de codificação, também referido

como uma CU folha. Sintaxe de dados associada com um fluxo de bits codificado pode definir um número máximo de vezes que um bloco em árvore pode ser dividido, referido como uma profundidade máxima de CU, e também pode definir um tamanho mínimo dos nós de codificação. Por conseguinte, um fluxo de bits pode também definir uma unidade de codificação menor (SCU). Esta descrição usa o termo "bloco" para se referir a qualquer um de uma CU, PU, ou TU, no contexto de HEVC, ou estruturas de dados semelhantes no contexto de outras normas (por exemplo, macroblocos e os sub-blocos do mesmo em H.264 / AVC).

[0045] Uma CU inclui um nó de codificação e as unidades de predição (PUs) e unidades de transformada (TUs) associadas com o nó de codificação. Um tamanho da CU corresponde a um tamanho do nó de codificação e deve ser de forma quadrada. O tamanho da CU pode variar de 8x8 pixels até o tamanho do bloco em árvore com um máximo de 64x64 pixels ou maior. Cada CU pode conter uma ou mais PUs e uma ou mais TUs. Dados de sintaxe associados a uma CU podem descrever, por exemplo, o particionamento da CU em uma ou mais PUs. Modos de particionamento podem diferir entre se a CU é ignorada ou codificada em modo direto, codificada em intrapredição, ou codificada em modo interpredição. PUs podem ser particionadas para serem não quadradas. Sintaxe de dados associada com uma CU também pode descrever, por exemplo, particionamento da CU em uma ou mais TUs de acordo com um quadtree. A TU pode ser quadrada ou não quadrada (por exemplo, retangular).

[0046] O padrão HEVC permite transformações de acordo com TUs, que podem ser diferentes para diferentes CUs. As TUs são tipicamente dimensionadas com base no tamanho de PUs dentro de uma determinada CU definida para uma LCU particionada, embora isso possa não ser sempre o

caso. As TUs são tipicamente do mesmo tamanho ou menores do que as PUs. Em alguns exemplos, as amostras residuais correspondentes a uma CU podem ser subdivididas em unidades menores que utilizam uma estrutura quadtree conhecida como "quad tree residual" (RQT). Os nós folha do RQT podem ser referidos como unidades de transformada (TUs). Os valores de diferença de pixel associados às TUs podem ser transformados para produzir coeficientes de transformada, que podem ser quantificados.

[0047] Uma CU folha pode incluir uma ou mais unidades de predição (PUs). Em geral, uma PU representa uma área espacial correspondente a toda ou a uma porção da CU correspondente, e pode incluir dados para obter uma amostra de referência para a PU. Além disso, uma PU inclui dados relacionados com a predição. Por exemplo, quando a PU é codificada em modo intra, os dados para a PU podem ser incluídos em um quadtree residual (RQT), que pode incluir dados que descrevem um modo de intrapredição para uma TU correspondente à PU. Como outro exemplo, quando a PU é codificada em modo inter, a PU pode incluir dados que definem um ou mais vetores de movimento para a PU. Os dados que definem o vetor de movimento para uma PU podem descrever, por exemplo, um componente horizontal do vetor de movimento, um componente vertical do vetor de movimento, uma resolução para o vetor de movimento (por exemplo, precisão de um quarto de pixel ou precisão de um oitavo de pixel), uma imagem de referência a qual o vetor de movimento aponta, e/ou uma lista de imagem de referência (por exemplo, lista 0, lista 1, ou lista C) para o vetor de movimento.

[0048] A informação de movimento para uma PU interpretada (mais particularmente, uma porção interpretada de uma CU a qual uma PU corresponde) pode

incluir um identificador de lista de imagem de referência e um índice de referência que corresponde a uma posição de uma imagem de referência na lista de imagens de referência. O codificador de vídeo 20, por exemplo, pode construir a lista de imagem de referência (incluindo quaisquer modificações feitas em uma lista de imagem de referência originalmente construída), e realizar uma pesquisa de movimento entre as imagens de referência na lista de quadro de referência para identificar um bloco que segue de perto (por exemplo, com base na soma das métricas de diferenças absolutas (SAD), ou métricas semelhantes), também referidas como um bloco de referência. De modo a sinalizar a localização do bloco de referência, o codificador de vídeo pode codificar 20 a lista de identificadores de imagem de referência e o índice de referência para a PU. Além disso, o codificador de vídeo 20 pode codificar os dados representativos de imagens de referência para ser incluídos na lista de imagem de referência, por exemplo, através da sinalização de dados representativa de valores de POC para as imagens de referência em um cabeçalho de fatia e/ou em um conjunto de parâmetros, tais como um PPS ou um SPS.

[0049] Uma CU folha com uma ou mais PUs também pode incluir uma ou mais unidades de transformada (TUs). As unidades de transformada podem ser especificadas utilizando uma RQT (também referidas como uma estrutura quadtree TU), como discutido acima. Por exemplo, um flag de divisão pode indicar se uma CU folha é dividida em quatro unidades de transformada. Em seguida, cada unidade de transformada pode ser adicionalmente dividida em outras sub-TUs. Quando uma TU não é dividida ainda mais, ela pode ser referida como uma TU folha. Geralmente, para intracodificação, todas as TUs folhas pertencentes a uma CU folha compartilham o mesmo modo de intrapredição. Isto é, o mesmo modo de

intrapredição é geralmente aplicado para calcular os valores previstos para todas as TUs de uma CU folha. Para a intracodificação, um codificador de vídeo pode calcular um valor residual para cada TU folha usando o modo de intrapredição, tal como uma diferença entre a parte da CU correspondente à TU e o bloco original. Uma TU não está necessariamente limitada ao tamanho de uma PU. Assim, a TU pode ser maior ou menor do que uma PU. Para a intracodificação, uma PU pode ser colocada com uma TU folha correspondente para a mesma CU. Em alguns exemplos, a dimensão máxima de uma TU folha pode corresponder ao tamanho da CU folha correspondente.

[0050] Além disso, as TUs de CUs folha podem também ser associadas com as respectivas estruturas de dados quadtree, referidas como quadrees residuais (RQTs). Isto é, uma CU folha pode incluir um quadtree indicando a forma como a CU folha é dividida em TUs. O nó raiz de um quadtree TU geralmente corresponde a uma CU folha, enquanto o nó raiz de um quadtree CU corresponde geralmente a um bloco em árvore (ou LCU). TUs de RQT que não são divididas são referidas como TUs folha. Em geral, a divulgação utiliza os termos CU e TU para se referir a CU folha e TU folha, respectivamente, salvo indicação em contrário.

[0051] Uma sequência de vídeo inclui tipicamente uma série de quadros de vídeo ou imagens. Um grupo de imagens (GOP) compreende geralmente uma série de uma ou mais das imagens de vídeo. Um GOP pode incluir dados de sintaxe em um cabeçalho de GOP, um cabeçalho de uma ou mais das imagens, ou em outro local, que descreve uma série de quadros incluídos no GOP. Cada fatia de uma imagem pode incluir dados de sintaxe fatia que descrevem um modo de codificação para a respectiva fatia. O codificador de vídeo 20 opera tipicamente em blocos de vídeo dentro de fatias de

vídeo individuais de modo a codificar os dados de vídeo. Um bloco de vídeo pode corresponder a um nó de codificação dentro de uma CU. Os blocos de vídeo podem ter fixos ou de tamanhos variados, e podem ser diferentes em tamanho de acordo com um padrão de codificação especificado.

[0052] A título de exemplo, o HM suporta predição em vários tamanhos de PU. Partindo do princípio de que o tamanho de uma CU particular é $2N \times 2N$, o HM suporta intrapredição em tamanhos de PU de $2N \times 2N$ ou $N \times N$, e interpredição em tamanhos simétricos de PU de $2N \times 2N$, $2N \times N$, $N \times 2N$, ou $N \times N$. O HM também suporta particionamento assimétrico para interpredição em tamanhos de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ e $nR \times 2N$. Em particionamento assimétrico, uma direção de uma CU não é repartida, enquanto que a outra direção é particionada em 25% e 75%. A porção da CU correspondente à partição 25% é indicada por um "n" seguido por uma indicação de "cima", "baixo", "esquerda", ou "direita". Assim, por exemplo, " $2N \times nU$ " refere-se a uma CU $2N \times 2N$ que é dividida horizontalmente com uma PU $2N \times 0.5N$ no topo e uma PU $2N \times 1.5N$ na parte inferior.

[0053] Nesta divulgação, " $N \times N$ " e "N por N" podem ser utilizados indiferentemente para se referir as dimensões de pixels de um bloco de vídeo em termos de dimensões vertical e horizontal, por exemplo, 16×16 pixels ou 16 por 16 pixels. Em geral, um bloco de 16×16 terá 16 pixels na direção vertical ($Y = 16$) e 16 elementos de imagem na direção horizontal ($X = 16$). Do mesmo modo, um bloco de $N \times N$ geralmente tem N pixels na direção vertical e N pixels na direção horizontal, em que n representa um valor inteiro positivo. Os pixels de um bloco podem ser dispostos em filas e colunas. Além disso, os blocos não têm necessariamente de ter o mesmo número de pixels na direção horizontal como na direção vertical. Por exemplo, os blocos

podem compreender, $N \times M$ pixels em que M não é necessariamente igual a N .

[0054] Na sequência de codificação intrapreditiva ou interpredictiva utilizando a PU de uma CU, codificador de vídeo 20 pode calcular dados residuais para as TUs da CU. A PU pode compreender dados de sintaxe que descrevem um método ou modo de gerar dados de pixel preditivos no domínio espacial (também referido como o domínio do pixel) e as TUs podem compreender coeficientes no domínio da transformada após a aplicação de uma transformada, por exemplo, uma transformada de cosseno discreta (DCT), uma transformada de número inteiro, uma transformada de wavelet, ou uma transformada conceitualmente similar aos dados de vídeo residuais. Os dados residuais podem corresponder a diferenças de pixel entre os pixels da imagem não codificada e os valores de predição correspondendo as PUs. O codificador de vídeo 20 pode formar as TUs incluindo os dados residuais na CU, em seguida, transformar as TUs para produzir coeficientes de transformada para a CU.

[0055] Na sequência de eventuais transformadas para produzir coeficientes de transformada, codificador de vídeo 20 pode realizar quantização dos coeficientes de transformada. Quantização geralmente refere-se a um processo em que os coeficientes de transformada são quantizados para possivelmente reduzir a quantidade de dados utilizada para representar os coeficientes, provendo a compressão adicional. O processo de quantização pode reduzir a profundidade do bit associada com alguns ou todos os coeficientes. Por exemplo, um valor de n bits pode ser arredondado para baixo para um valor de m bits durante quantização, em que n é maior do que m .

[0056] Após a quantização, o codificador vídeo pode digitalizar os coeficientes de transformada, produzindo um vetor unidimensional a partir da matriz bidimensional incluindo os coeficientes de transformada quantificados. A varredura pode ser concebida para colocar coeficientes de energia mais elevada (e, portanto, menor frequência) na frente da matriz e para colocar coeficientes de energia mais baixa (e, portanto, maior frequência) na parte de trás da matriz. Em alguns exemplos, o codificador de vídeo 20 pode utilizar uma ordem de pesquisa predefinida para varrer os coeficientes de transformada quantificados para produzir um vetor em série que pode ser codificado por entropia. Em outros exemplos, codificador de vídeo 20 pode executar uma verificação adaptativa. Após a varredura de coeficientes de transformada quantificados, para formar um vetor unidimensional, um codificador de vídeo 20 pode codificar por entropia o vetor unidimensional, por exemplo, de acordo com codificação de comprimento variável adaptativa de contexto (CAVLC), codificação aritmética binária adaptativa de contexto (CABAC), codificação aritmética binária adaptativa de contexto baseada em sintaxe (SBAC), codificação por entropia de particionamento de intervalo de probabilidade (PIPE) ou outra metodologia de codificação de entropia. O codificador de vídeo 20 pode também codificar por entropia elementos de sintaxe associados com os dados de vídeo codificados para utilização pelo decodificador de vídeo 30 na codificação dos dados de vídeo.

[0057] Para executar CABAC, codificador de vídeo 20 pode atribuir um contexto dentro de um modelo de contexto para um símbolo a ser transmitido. O contexto pode referir-se, por exemplo, a se os valores vizinhos do símbolo são diferentes de zero ou não. Para realizar CAVLC,

codificador de vídeo 20 pode selecionar um código de comprimento variável para um símbolo a ser transmitido. Palavras código em VLC podem ser construídas de tal forma que códigos relativamente mais curtos correspondem a símbolos mais prováveis, enquanto códigos mais longos correspondem a símbolos menos prováveis. Desta forma, o uso do VLC pode conseguir uma economia de bits, por exemplo, através do uso das palavras código de comprimento igual para cada símbolo a ser transmitido. A determinação de probabilidade pode ser baseada em um contexto atribuído ao símbolo.

[0058] Codificador de vídeo 20 pode ainda enviar dados de sintaxe, tais como dados de sintaxe baseados em blocos, dados de sintaxe baseados em quadro, e dados de sintaxe baseados em GOP, para decodificador de vídeo 30, por exemplo, em um cabeçalho de quadro, um cabeçalho de bloco, um cabeçalho de fatia, ou um cabeçalho GOP. Os dados de sintaxe GOP podem descrever um número de quadros no respectivo GOP, e os dados de sintaxe de quadro podem indicar um modo de codificação / predição utilizado para codificar o quadro correspondente.

[0059] Segmentos de vídeo codificados podem ser organizados em unidades NAL, que proveem aplicações de endereçamento de representação de vídeo "amigáveis de rede" tais como telefonia de vídeo, armazenamento, broadcast ou streaming. Unidades NAL podem ser categorizadas como unidades NAL de Camada de Codificação de Vídeo (VCL) e unidades NAL não-VCL. Unidades VCL podem conter saída a partir do motor de compressão do núcleo e podem incluir dados do nível de bloco, macrobloco, e/ou de fatia. Outras unidades NAL podem ser unidades NAL não-VCL. Em alguns exemplos, uma imagem codificada em um instante de tempo, normalmente apresentada como uma imagem primária

codificada, pode estar contida em uma unidade de acesso, que pode incluir uma ou mais unidades NAL.

[0060] Como mencionado acima, codificador de vídeo 20 e decodificador de vídeo 30 podem geralmente ser configurados para operar de acordo com um padrão de codificação de vídeo ou uma extensão a um padrão de codificação de vídeo, tal como MV-HEVC ou SHVC. Assim, para fins de exemplo, as técnicas da presente divulgação encontram-se descritas abaixo no que diz respeito a VM-HEVC, mas deve ser entendido que estas técnicas podem ser aplicadas a outras normas de codificação de vídeo ou extensões, como SHVC.

[0061] Em HEVC, há um número de diferentes tipos de imagem que podem ser identificados pelo tipo de unidade NAL. Um tipo de imagem é um tipo de imagem de acesso aleatório, que pode geralmente incluem imagens IDR, imagens de acesso aleatório limpo (CRA), imagens de acesso de camada temporal (TLA), imagens de acesso de link interrompido (BLA), imagens STSA (Acesso de subcamada temporal por etapa), imagens dianteiras (ou seja, imagens RASL e RADL indicadas abaixo) ou uma série de outros tipos de imagem diferentes.

[0062] Codificador de vídeo 20 e decodificador de vídeo 30 podem atribuir a cada imagem de uma sequência de vídeo um valor de POC para a identificação da imagem. O codificador de vídeo 20 pode codificar (e o decodificador de vídeo 30 pode decodificar) os dados representativos de imagens de referência para ser incluídos na lista de imagem de referência, por exemplo, através da sinalização de dados representativos de valores de POC para as imagens de referência em um cabeçalho de fatia e/ou em um parâmetro definido, como um PPS ou um SPS. Um valor de POC, representada pelo elemento de sintaxe PicOrderCntVal, para

uma determinada imagem codificada denota ordem relativa da imagem no processo de saída de imagem com respeito a outras imagens na mesma sequência de vídeo codificado.

[0063] Um valor de POC inclui bits menos significativos (LSB) e os bits mais significativos (MSB). O POC pode ser obtido através da concatenação ou adicionando o MSB com o LSB. O bit menos significativo pode ser sinalizado no cabeçalho de fatia, e MSB pode ser computado pelo decodificador de vídeo 30, com base no tipo de unidade NAL da imagem atual e no MSB e LSB da imagem anterior, em ordem de decodificação que não é do tipo de unidade NAL aleatório acesso dianteiro ignorado (RASL) ou acesso aleatório decodificável dianteiro (RADL) ou nem uma imagem de subcamada nem de referência, e que tem um valor `temporal_id` que é igual a 0. Essas imagens que têm `TemporalId` igual a 0 e `nuh_layer_id` igual a `nuh_layer_id` da imagem atual e que não são imagens RASL, imagens RADL ou imagens de subcamada de não referência, são referidas como imagens POC-âncora.

[0064] Quando a imagem atual é uma imagem de ponto de acesso intra-aleatório (IRAP) com `NoRaslOutputFlag` igual a 1, ou uma imagem CRA, que é a primeira imagem do fluxo de bits, o valor de POC MSB é inferido (isto é, determinado pelo codificador) para ser igual a 0. Em um fluxo de bits de multicamada (por exemplo, um fluxo de bits SHVC ou MV-HEVC com mais do que uma camada), podem existir unidades de acesso, onde uma ou mais do que uma imagem é uma imagem IAPR e uma ou mais imagens são imagens não-IRAP. Tal AUs podem ser referidos como AUs IRAP não alinhados. Ao decodificar fluxos de bits contendo AUs IRAP não alinhados, pode ser que o POC derivado para a imagem com base nos valores POC LSB sinalizados viole a exigência de fluxo de bits que todas as imagens (por exemplo, imagens de cada

camada) em uma unidade de acesso devem ter o mesmo valor de PicOrderCntVal.

[0065] Conseqüentemente, em alguns casos, valores de POC podem ser redefinidos para uma unidade de acesso particular, a fim de manter o alinhamento de POC das imagens na unidade de acesso. Em MV-HEVC WD5, um flag de elemento de sintaxe poc_reset_flag pode ser usado para redefinir o POC das imagens de tal forma que, mesmo quando AUs IRAP não alinhados estão presentes no fluxo de bits, o valor de PicOrderCntVal da imagem atual e das imagens no DPB seria ajustado de tal modo que o POC de todas as imagens em um AU é o mesmo.

[0066] Chen et al., "CROSS-LAYER POC ALIGNMENT FOR MULTI-LAYER BITSTREAMS THAT MAY INCLUDE NON-ALIGNED IRAP PICTURES", Pedido de Patente U.S. No. de série 14/245.115, depositado em 4 de abril de 2014, descreve outro método de alcançar uma redefinição de POC, que usa dois flags: poc_msb_reset_flag e poc_lsb_reset_flag. O flag anterior redefine o MSB de PicOrderCntVal, e o último flag redefine o LSB de PicOrderCntVal. Ambos os flags são sinalizados no cabeçalho de fatia.

[0067] U.S. Provisório No. 61/890.868, depositado em 14 de outubro de 2013 propõe uma mensagem SEI contendo informações para a recuperação de valores de POC corretos quando uma imagem que contém um POC MSB ou indicação de redefinição de POC está perdida.

[0068] Hannuksela et al., "MV-HEVC/SHVC HLS: on POC value derivation", Joint Collaborative Team on Video Coding (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 15th Meeting: Geneva, CH, 23 Oct.- 1 Nov. 2013, JCTVC-00275v3 (disponível em http://phenix.int-evry.fr/jct/doc_end_user/documents/15_Geneva/wg11/JCTVC-00275-v3.zip) e Sjoberg et al., "HLS: Error robust POC

alignment," Joint Collaborative Team on Video Coding (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 15th Meeting: Geneva, CH, 23 Oct.- 1 Nov. 2013, JCTVC-00176v3 (disponível em http://phenix.int-evry.fr/jct/doc_end_user/documents/15_Geneva/wg11/JCTVC-00176-v3.zip), propõem outros métodos para sinalização e derivação de valores de POC em codificação de vídeo de multicamada.

[0069] Pedido U.S No. 14/543,470, depositado em 17 de novembro de 2014 propõe substituir os dois flags (ou seja, `poc_msb_reset_flag` e `poc_lsb_reset_flag`) por um indicador de redefinição de POC de dois bits e introduz o conceito de período de redefinição de POC em que cada período de redefinição de POC é identificado por um identificador de período de redefinição de POC, que é necessário ser diferente para dois períodos consecutivos de redefinição de POC. Com tais conceitos, a resiliência de erro de mecanismo de redefinição de POC é melhorada, de tal forma que é possível para o decodificador manter o alinhamento da camada cruzada de POC em casos de perda de uma ou mais imagens na unidade de acesso, onde a redefinição de POC é realizada.

[0070] Em geral, um período de redefinição de POC pode ser definido como segue, por exemplo, na subcláusula F.3 de MV-HEVC:

F.3.1 período de redefinição de contagem de ordem de imagem (POC): Uma sequência de imagens na ordem de decodificação em uma camada que todas têm `poc_reset_idc` maior que 0 e o mesmo valor de `poc_reset_period_id`. Alternativamente, a frase "dentro de uma camada" na definição acima pode ser omitida, por exemplo, como se segue:

F.3.2 período de redefinição de contagem de ordem de imagem (POC): Uma sequência de imagens, em ordem de decodificação que todas têm poc_reset_idc maior que 0 e o mesmo valor de poc_reset_period_id.

Assim, o período de redefinição de POC é específico de camada. Em um exemplo, MV-HEVC (ou outro padrão ou extensão padrão aplicável) pode requerer que as imagens em um período de redefinição de POC sejam contínuas, em ordem de decodificação dentro de uma camada. No entanto, permitir que estas imagens sejam intercaladas com imagens com poc_reset_idc igual a 0 na mesma camada pode adicionar flexibilidade e permitir a utilização do mesmo overhead para conseguir uma melhor resiliência de erro para as perdas de pacotes em cluster.

[0071] Codificador de vídeo 20 pode atribuir fatias para respectivos períodos de redefinição de POC. Cada período de redefinição de POC pode incluir uma ou mais fatias. Assim, quando os valores de POC são redefinidos por um período de redefinição de POC (incluindo redefinir valores de POC de imagens de referência anteriores ao período de redefinição de POC em ordem de codificação), codificador de vídeo 20 pode sinalizar valores de POC de imagens de referência para serem incluídos em uma lista de imagem de referência com base nos valores de POC redefinidos.

[0072] Codificador de vídeo 20 e decodificador de vídeo 30 podem realizar a saída de imagem e colisão de imagens armazenadas no DPB usando valores de POC. O processo de emissão e colisão de imagem é o processo de emissão de imagens que são marcadas como "necessárias para emissão" e "sem uso para referência" a partir da DPB. Após a adoção de Ramasubramonian et al., "MV-HEVC/SHVC HLS: Sub-DPB based DPB operations," Joint Collaborative Team on

Video Coding (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 15th Meeting: Geneva, CH, 23 Oct.- 1 Nov. 2013, JCTVC-00217 (disponível em http://phenix.int-evry.fr/jct/doc_end_user/documents/15_Geneva/wg11/JCTVC-00217-v1.zip) o conceito para emitir e colidir imagens do DPB para extensões HEVC foi atualizado para incorporar o seguinte:

- Especificar uma capacidade de DPB separada para cada camada chamada sub-DPB
- Não há compartilhamento da capacidade de sub-DPB ao longo das camadas
- Cada camada tem seus próprios parâmetros (imagens max, latência max, reordenamento max)
- Parâmetros distintos para cada "conjunto de camada de saída"
- Alterar a definição de um ponto de operação para ser específico para um conjunto de camada de saída em vez de um "conjunto de camadas"

Após as alterações, processos de emissão e colisão de imagem funcionam da seguinte forma: para a conformidade do decodificador, o processo de emissão da imagem é invocado duas vezes para cada imagem, incluindo (1) antes da decodificação da imagem, mas depois de analisar o cabeçalho de fatia da primeira fatia da imagem, e (2) quando a última unidade de decodificação da imagem é removida do armazenador de imagem codificada. Quando o processo de colisão é invocado, o processo gera todas as imagens que pertencem à unidade de acesso selecionada.

[0073] Uma operação exemplar da ordem de saída para um DPB, incluindo a saída de imagem e colisão, após a inclusão das adições em JCTVC-00217 e de Ramasubramonian et al., "MV-HEVC/SHVC HLS: On flushing of decoded pictures from DPB based on NoOutputOfPriorPicsFlag," Joint

Collaborative Team on Video Coding (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 15th Meeting: Geneva, CH, 23 Oct.- 1 Nov. 2013, 00266 (disponível em http://phenix.int-evry.fr/jct/doc_end_user/documents/15_Geneva/wg11/JCTVC-00266-v1.zip) para MV-HEVC WD6 é como se segue:

C.5.2 Operação da ordem de saída de DPB

C.5.2.1 Geral

O armazenador de imagem decodificada consiste em sub-DPBS, e cada sub-DPB contém armazenadores de armazenamento de imagem para o armazenamento de imagens decodificadas de apenas uma camada. Cada um dos armazenadores de armazenamento de imagem de um sub-DPB contém uma imagem decodificada que está marcada como "usada para a referência" ou é mantida para a emissão futura.

O processo para emissão e remoção de imagens do DPB, o processo de decodificação de imagem, marcação, colisão adicional e armazenamento, e do processo de "colisão" estão especificados abaixo. Estes processos são aplicados independentemente para cada camada, a partir da camada de base, em ordem crescente de valores `nuh_layer_id` das camadas no fluxo de bits crescente. Quando esses processos são aplicados a uma camada específica, apenas o sub-DPB para a camada específica é afetado, exceto para o processo de "colisão", que pode cortar e emitir imagens, marcar imagens como "não é necessário para emissão" e armazenadores de armazenamento de imagens vazias para qualquer camada.

Deixe imagem n e a imagem atual ser imagem codificada ou imagem decodificada da unidade de acesso n para um valor particular de `nuh_layer_id`, em que n é um número inteiro não negativo.

Quando esses processos são aplicados a uma camada com `nuh_layer_id` igual a `currLayerId`, as variáveis `MaxNumReorderPics`, `MaxLatencyIncreasePlus1`, `MaxLatencyPictures`, e `MaxDecPicArmacenadoringMinus1` são derivadas da seguinte forma:

- Se um CVS em conformidade com um ou mais dos perfis especificados no anexo G ou H é decodificado através da aplicação do processo de decodificação especificado nas cláusulas 2-10, anexo F, e Anexo G ou H, o seguinte se aplica:

- `MaxNumReorderPics` está definido igual a `max_vps_num_reorder_pics [TargetOptLayerSetIdx] [HighestTid]` do VPS ativo.

- `MaxLatencyIncreasePlus1` é definido igual ao valor do elemento de sintaxe `max_vps_latency_increase_plus1 [TargetOptLayerSetIdx] [HighestTid]` do VPS ativo,.

- `MaxLatencyPictures` está definido igual a `VpsMaxLatencyPictures [TargetOptLayerSetIdx] [HighestTid]` do VPS ativo.

- `MaxDecPicArmacenadoringMinus1` é definido igual ao valor do elemento de sintaxe `max_vps_dec_pic_buffering_minus1 [TargetOptLayerSetIdx] [currLayerId] [HighestTid]` do VPS ativo.

- Caso contrário (um CVS em conformidade com um ou mais dos perfis especificados no Anexo A é decodificado através da aplicação do processo de decodificação especificado nas cláusulas 2-10), se aplica o seguinte:

- `MaxNumReorderPics` está definido igual a `sps_max_num_reorder_pics [HighestTid]` do SPS ativo para a camada de base.

- `MaxLatencyIncreasePlus1` está definido igual `sps_max_latency_increase_plus1 [HighestTid]` do SPS ativo para a camada de base.

- MaxLatencyPictures está definido igual a SpsMaxLatencyPictures [HighestTid] do SPS ativo para a camada de base.

- MaxDecPicArmacenadoringMinus1 está definido igual a sps_max_dec_pic_buffering_minus1 [HighestTid] do SPS ativo para a camada de base.

C.5.2.2 Emissão e remoção de imagens do DPB

Quando a imagem atual não é a imagem 0 (ou seja, a imagem atual não tem um valor de POC de 0) na camada atual, emissão e remoção de imagens na camada atual do DPB antes da decodificação da imagem atual, ou seja, imagem n, mas depois de analisar o cabeçalho da fatia da primeira fatia da imagem atual, acontece instantaneamente quando a primeira unidade de decodificação da imagem atual é removida do CPB e prossegue como se segue:

- O processo de decodificação para RPS, conforme especificado na subcláusula 8.3.2 é invocado.

- Se a imagem atual é uma imagem IRAP com NoRaslOutputFlag igual a 1, ou a imagem de camada de base na unidade de acesso atual é uma imagem IRAP com NoRaslOutputFlag igual a 1 e NoClrasOutputFlag é igual a 1, as seguintes etapas ordenadas, são aplicadas:

1. A variável NoOutputOfPriorPicsFlag é derivada para o decodificador em teste da seguinte forma:

- Se a imagem atual é uma imagem CRA com NoRaslOutputFlag igual a 1, NoOutputOfPriorPicsFlag está definida igual a 1 (independentemente do valor da no_output_of_prior_pics_flag).

- Caso contrário, se a imagem atual é uma imagem IRAP com NoRaslOutputFlag igual a 1 e o valor de pic_width_in_luma_samples, pic_height_in_luma_samples, ou sps_max_dec_pic_buffering_minus1 [HighestTid] derivada do SPS ativo para a camada atual é diferente do valor de

pic_width_in_luma_samples, pic_height_in_luma_samples ou
sps_max_dec_pic_buffering_minus1 [HighestTid],
respectivamente, derivado do SPS que estava ativo para a
camada atual ao decodificar a imagem anterior na camada
atual, NoOutputOfPriorPicsFlag pode (mas não deve) ser
definido como 1 pelo decodificador em teste,
independentemente do valor de no_output_of_prior_pics_flag.

OBSERVAÇÃO - Embora definir
NoOutputOfPriorPicsFlag igual a
no_output_of_prior_pics_flag seja preferido sob estas
condições, o decodificador sob teste é permitido definir
NoOutputOfPriorPicsFlag para 1 neste caso.

- Caso contrário, se a imagem atual é uma imagem
IRAP com NoRaslOutputFlag igual a 1,
NoOutputOfPriorPicsFlag está definida igual
ano_output_of_prior_pics_flag.

- Caso contrário (a imagem atual não é uma imagem
IRAP com NoRaslOutputFlag igual a 1, a imagem de camada de
base na unidade de acesso atual é uma imagem de IAPR com
NoRaslOutputFlag igual a 1, e NoClrasOutputFlag é igual a
1), NoOutputOfPriorPicsFlag é definida como igual a 1.

2. O valor de NoOutputOfPriorPicsFlag derivado
para o decodificador sob teste é aplicado para a DRH como
se segue:

- Se NoOutputOfPriorPicsFlag é igual a 1, todos
os armazenadores de armazenamento de imagens no sub-DPB são
esvaziados emissão das imagens que eles contêm, e a
totalidade de sub-DPB é definida igual a 0.

- Caso contrário (NoOutputOfPriorPicsFlag é igual
a 0), todos os armazenadores de armazenamento de imagem que
contêm uma imagem que está marcada como "não é necessária
para emissão" e "não utilizada para referência" são
esvaziados (sem emissão), e todos os armazenadores de

armazenamento de imagem não vazios no sub-DPB são esvaziados invocando repetidamente o processo de "colisão" especificado na subcláusula C.5.2.4, e a totalidade de sub-DPB é definida igual a 0.

- Caso contrário, todos os armazenadores de armazenamento de imagem que contêm uma imagem na camada atual e que são marcados como "não é necessário para emissão" e "não utilizado para referência" são esvaziados (sem as imagens sendo emitidas). Para cada armazenador de armazenamento de imagem que é esvaziado, a totalidade de sub-DPB é diminuída por um. Quando uma ou mais das seguintes condições forem verdadeiras, o processo de "colisão" especificado na subcláusula C.5.2.4 é invocado repetidamente enquanto diminui ainda mais a totalidade de sub-DPB em um a cada armazenador de armazenamento de imagens adicional que é esvaziado, até que nenhuma das seguintes condições sejam verdadeiras:

- O número de unidades de acesso que contêm pelo menos uma imagem decodificada no DPB (isto é, qualquer um dos sub-DPBs) marcada como "necessária para emissão" é maior do que MaxNumReorderPics.

- MaxLatencyIncreasePlus1 não é igual a 0 e existe pelo menos uma unidade de acesso que contém pelo menos uma imagem decodificada no DPB marcado como "necessária para emissão" para a qual a variável associada PicLatencyCount é maior do que ou igual a MaxLatencyPictures.

- O número de imagens na camada atual no sub-DPB da camada atual é maior do que ou igual a MaxDecPicArmazenadoringMinus1 + 1.

C.5.2.3 Decodificação, marcação, colisão adicional e armazenamento de imagem

Os processos especificados nesta subcláusula acontecem instantaneamente quando a última unidade de decodificação de imagem n é removida da CPB.

PicOutputFlag é atualizado como segue:

- Se a unidade de acesso atual não contém uma imagem em uma camada de saída alvo e `alt_output_layer_flag` é igual a 1, as seguintes etapas ordenadas se aplicam:

- A lista `nonOutputLayerPictures` é a lista de imagens da unidade de acesso com `PicOutputFlag` igual a 1 e com valores `nuh_layer_id` que estão incluídos na `TargetDecLayerIdList` e que não estão em camadas de saída de destino.

- A imagem com o maior valor `nuh_layer_id` entre a lista `nonOutputLayerPictures` é removida da lista `nonOutputLayerPictures`.

- `PicOutputFlag` para cada imagem que está incluída na lista `nonOutputLayerPictures` é definida igual a 0.

- Caso contrário, `PicOutputFlag` para imagens que não estão incluídas em uma camada de saída de destino é definida igual a 0.

Quando a imagem atual tem `PicOutputFlag` igual a 1, para cada imagem na camada atual no sub-DPB que está marcado como "necessária para emissão" e segue a imagem atual, em ordem de emissão, a variável associada `PicLatencyCount` é definida igual a `PicLatencyCount + 1`.

A imagem atual é considerada como decodificada após a última unidade de decodificação da imagem ser decodificada. A imagem atual decodificada é armazenada em um armazenador de armazenamento de imagens vazio no sub-DPB, e o seguinte é aplicável:

- Se a imagem atual decodificada tem PicOutputFlag igual a 1, ela é marcada como "necessária para emissão" e sua variável associada PicLatencyCount está definida igual a 0.

- Caso contrário (a imagem atual decodificada tem PicOutputFlag igual a 0), ela é marcada como "não é necessária para emissão".

A imagem atual decodificada é marcada como "usada para referência de curto prazo".

Quando uma ou mais das seguintes condições forem verdadeiras, o processo de "colisão" especificado na subcláusula C.5.2.4 é chamado repetidamente até nenhuma das seguintes condições serem verdadeiras:

- O número de que contêm pelo menos uma imagem decodificada no DPB (isto é, qualquer um dos sub-DPBs) marcada como "necessária para emissão" é maior do que MaxNumReorderPics.

- MaxLatencyIncreasePlus1 não é igual a 0 e existe pelo menos uma unidade de acesso que contém pelo menos uma imagem decodificada no DPB marcada como "necessária para emissão" para a qual a variável associada PicLatencyCount é maior do que ou igual a MaxLatencyPictures.

C.5.2.4 processo de "colisão"

O processo de "colisão" consiste nas seguintes etapas ordenadas:

1. A imagem ou imagens que são primeiro para emissão são selecionadas como as que têm o menor valor de PicOrderCntVal de todas as imagens no DPB marcadas como "necessárias para emissão".

2. Cada uma destas imagens é, em ordem nuh_layer_id crescente, cortada, usando a janela de corte de conformação especificada no SPS ativo para a imagem, a

imagem cortada é emitida, e a imagem é marcada como "não é necessária para emissão".

3. Cada armazenador de armazenamento de imagem que contém uma imagem marcada como "sem uso para referência" e que foi uma das imagens cortadas e emitidas é esvaziado.

[0074] De acordo com as técnicas desta divulgação, codificador de vídeo 20 e decodificador de vídeo 30 podem ser, respectivamente configurados para diminuir valores de POC de todas as imagens (por exemplo, imagens de todas as camadas) armazenadas em um DPB antes da decodificação de uma imagem inicial de um unidade de acesso em que redefinição de POC é realizada. Por exemplo, codificador de vídeo 20 e decodificador de vídeo 30 podem aplicar um diminuição de POC em imagens que tenham sido processadas e armazenadas no DPB anteriormente à imagem sendo codificada (por exemplo, imagens que antecedem a imagem sendo codificada, em ordem de codificação, que pode, em geral ser referida como "imagens anteriores" para a imagem atual). De acordo com aspectos da presente divulgação, o codificador de vídeo 20 e o decodificador de vídeo 30 podem aplicar diminuição de POC a tais imagens anteriores em um DPB (isto é, para todos os sub-DPBs) uma vez, no início de cada unidade de acesso que requer redefinição de POC. O processo de diminuição de POC pode ser invocado antes da codificação (por exemplo, codificação ou decodificação) da imagem inicial para a unidade de acesso, mas depois (ou seja, posteriormente à) codificação de informações de cabeçalho de fatia da primeira fatia da imagem inicial. Desta forma, os valores de POC de imagens armazenadas no DPB do codificador de vídeo 20 e decodificador de vídeo 30 podem permanecer alinhados ao codificar as imagens da unidade de acesso, permitindo assim

que as imagens do DPB sejam emitidas na ordem correta e, assim, assegurar que as imagens possam ser exibidas corretamente. Por exemplo, ao manter o alinhamento de camada cruzada de imagens armazenadas no DPB (por exemplo, de modo que as imagens de cada respectiva unidade de acesso tenham os mesmos valores de POC), decodificador de vídeo 30 pode emitir todas as imagens de uma unidade de acesso antes da emissão de imagens de outra unidade de acesso. Isso pode permitir que as imagens da unidade de acesso sejam exibidas praticamente em simultâneo para prover uma experiência de visualização apropriada.

[0075] Em outro exemplo, de acordo com aspectos da presente divulgação, a temporização da diminuição de POC pode ser alterada. Por exemplo, decodificador de vídeo 30 pode invocar o processo de diminuição de POC para imagens que tenham sido processadas (por exemplo, decodificadas) e armazenadas no DPB anterior à imagem que está sendo decodificada em cada sub-DPB antes de decodificar cada imagem na unidade de acesso, mas depois da análise e decodificação da informação do cabeçalho de fatia da primeira fatia da imagem. O decodificador de vídeo 30 também pode executar um processo de diminuição de POC adicional de imagens anteriores em um sub-DPB para uma camada que tem uma imagem que não está presente ou não foi recebida pela unidade de acesso depois de decodificar as imagens que estão presentes na unidade de acesso.

[0076] Em ainda outro exemplo, de acordo com aspectos desta divulgação, decodificador de vídeo 30 pode invocar o processo de diminuição de POC de imagens anteriores em cada sub-DPB antes da decodificação de cada imagem na unidade de acesso, mas após a análise e decodificação das informações de cabeçalho de fatia da primeira fatia da respectiva imagem. Antes do processo de

diminuição de POC de imagens anteriores em um sub-DPB é executado, se o mesmo processo que ainda não foi realizado para um ou mais DPBs de subcamada inferior, o processo é realizado para os sub-DPBs.

[0077] Em ainda outro exemplo, de acordo com aspectos da presente divulgação, decodificador de vídeo 30 pode realizar o processo de diminuição de POC de imagens anteriores no DPB (isto é, para todos os sub-DPBs) é aplicado uma vez, no final de decodificação de cada unidade de acesso que necessita de redefinição de POC. Ou seja, decodificador de vídeo 30 pode decodificar todas as imagens de uma unidade de acesso para com redefinição de POC sendo realizada, seguido por diminuição dos valores de POC de todas as imagens do DPB.

[0078] De acordo com aspectos da presente divulgação, o codificador de vídeo 20 e decodificador de vídeo podem ser respectivamente configurados de acordo com restrições (por exemplo, restrições de fluxo de bits, conforme determinado e considerado aplicável, por um codificador, para um dado fluxo de bits), criadas por um padrão de codificação de vídeo aplicável, tais como MT-HEVC. Por exemplo, de acordo com aspectos desta divulgação, o valor derivado para diminuição de imagens anteriores em um DPB (por exemplo, todos os sub-DPBs) deve ser o mesmo para todas as imagens em uma unidade de acesso, onde redefinição de POC ou POC MSB é realizada.

[0079] Como mencionado acima, um período de redefinição de POC específico de camada pode ser especificado com base em um identificador de período de redefinição de POC sinalizado nos cabeçalhos de segmento de fatia. Ou seja, codificador de vídeo 20 e decodificador de vídeo 30 podem, respectivamente codificar dados representantes do identificador de período de redefinição

de POC em cabeçalhos de segmento de fatia. Cada imagem não IAPR que pertence a uma unidade de acesso que contém pelo menos uma imagem de IAPR pode ser o início de um período de redefinição de POC na camada que contém a imagem não IAPR. Ou seja, codificador de vídeo 20 pode definir um tipo de redefinição de POC para uma imagem não-IRAP de uma unidade de acesso contendo pelo menos uma imagem IRAP para indicar que a imagem não-IRAP é o início de um novo período de redefinição de POC. Em uma tal unidade de acesso, cada imagem seria o início de um período de redefinição de POC na camada que contém a imagem. Redefinição de POC, apenas POC MSB ou ambos POC MSB e POC LSB e atualização de valores de POC de imagens de mesma camada no DPB só pode ser aplicada para a primeira imagem dentro de cada período de redefinição de POC.

[0080] O codificador de vídeo 20 pode sinalizar um valor POC LSB em um cabeçalho do segmento de fatia que o decodificador de vídeo 30 pode usar para derivar valores de POC de imagens em uma camada incluindo a fatia tendo o cabeçalho do segmento de fatia. O valor de POC LSB sinalizado no cabeçalho do segmento de fatia pode ser utilizado para a derivação do valor de COP delta, que é usado para atualizar os valores de POC das imagens do DPB de mesma camada, e também para a derivação do POC MSB do valor de POC da imagem atual. Quando a imagem atual tem um tal valor de POC LSB sinalizado, e quando a imagem POC-âncora associada com a imagem atual está presente no fluxo de bits, a imagem POC-âncora tem uma indicação de qualquer redefinição de POC ou redefinição de POC MSB. Tal valor de POC LSB sinalizado para a imagem atual é igual ao valor de POC LSB da imagem POC-âncora, que é também a primeira imagem do período de redefinição de POC que a imagem atual. De acordo com aspectos da presente divulgação, o

codificador de vídeo 20, sinaliza dados que indicam o POC LSB adicional para imagens IDR de camada de base possuindo um elemento de sintaxe poc_reset_idc que é igual a dois.

[0081] Como mencionado acima, o codificador de vídeo 20 e o decodificador de vídeo 30 podem ser configurados, respectivamente, de acordo com restrições impostas por um padrão de codificação de vídeo aplicável. Por exemplo, de acordo com aspectos desta divulgação, todas as imagens na mesma unidade de acesso devem ter o mesmo valor de poc_reset_idc. Além disso, como outro exemplo, 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. Além disso, como outro exemplo, uma unidade de acesso com poc_reset_idc igual a 1 ou 2 será a primeira unidade de acesso em um período de redefinição de POC.

[0082] De acordo com aspectos da presente divulgação, o período de redefinição de POC pode ser aplicável a todas as camadas dos dados de vídeo de multicamada. Por exemplo, um período de redefinição de POC pode ser definido de tal modo que o período de redefinição de POC é aplicado através de camadas de dados multicamada. Tal como descrito em maior detalhe abaixo, o período de redefinição de POC pode incluir uma sequência de unidades de acesso, em ordem de decodificação que todos têm um poc_reset_idc que é maior do que 0. Os dados que indicam o valor de redefinição de POC discutido acima podem ser associados com um período de redefinição de POC.

[0083] O seguinte divulgação e tabelas incluem exemplo de sintaxe de acordo com as técnicas desta divulgação. A sintaxe pode ser modificada em relação, por exemplo, aquela de MV-HEVC conforme modificado por documentos JCTVC-00217 e JCTVC-00266 mencionados acima. No exemplo da descrição, tabelas de sintaxe e semântica

abaixo, adições a MV-HEVC (como alterado por documentos JCTVC-00217 e JCTVC-00266) estão representadas em *itálico* e as supressões são representadas por meio de texto entre colchetes precedido por "removido" (por exemplo, [removido: "texto removido"]). Em geral, as declarações em relação a "requisitos" devem ser entendidas de modo a formar parte do texto do padrão ou extensão padrão, e não é um requisito para os fins das técnicas desta descrição. Em alguns casos, tais "exigências" podem incluir restrições de fluxo de bits que podem ser determinadas como sendo aplicável e, em seguida, aderiram, por exemplo, um codificador de vídeo com base na determinação).

[0084] A seguinte definição de uma função matemática pode ser adicionada a MV-HEVC, por exemplo, a subcláusula 5.8 de MV-HEVC:

$$GetCurrMsb(cl, pl, pm, ml) = \begin{cases} pm + ml & ; \quad pl - cl \geq ml / 2 \\ pm - ml & ; \quad cl - pl > ml / 2 \\ pm & ; \quad \text{caso contrário} \end{cases}$$

[0085] A seguinte definição pode ser adicionada à sub-cláusula F.3:

F.3.1 período de redefinição de contagem de ordem de imagem (POC): Uma sequência de unidades de acesso, em ordem de decodificação que todos têm poc_reset_idc maior do que 0 e o mesmo valor de poc_reset_period_id, que podem ou não podem ser intercaladas com as unidades de acesso que têm poc_reset_idc igual a 0.

[0086] Em uma alternativa, imagens em um período de redefinição de POC são obrigadas a ser contínuas, em ordem de decodificação dentro de uma camada. No entanto, permitir que estas imagens a serem intercaladas com imagens com poc_reset_idc igual a 0 na mesma camada é mais flexível

e permite usar o mesmo overhead para conseguir uma melhor resiliência de erro para as perdas de pacotes em cluster.

[0087] A tabela seguinte é um exemplo de uma sintaxe de PPS de acordo com as técnicas desta divulgação:

	Descritor
pic_parameter_set_rbsp() {	
...	
pps_extension_flag	u(1)
if(pps_extension_flag) {	
poc_reset_info_present_flag	u(1)
pps_extension2_flag	u(1)
if(pps_extension2_flag)	
while(more_rbsp_data()	
pps_extension_data_flag	u(1)
}	
rbsp_trailing_bits()	
}	

[0088] No exemplo acima, pps_extension_flag igual a 0 especifica que não há elementos de sintaxe pps_extension_data_flag presentes na estrutura de sintaxe PPS Rbsp. Quando slice_segment_header_extension_present_flag é igual a 0, pps_extension_flag será igual a 0 em fluxos de bits em de acordo com a presente versão desta especificação, o valor de 1 para pps_extension_flag é reservado para uso futuro por ITU-T | ISO / IEC e decodificadores devem ignorar todos os dados que seguem o valor 1 para pps_extension_flag em uma unidade PPS NAL.

[0089] Além disso, poc_reset_info_present_flag igual a 0 especifica que o elemento de sintaxe poc_reset_idc não está presente nos cabeçalhos de segmento de fatia das fatias que se referem ao PPS. Além disso, poc_reset_info_present_flag igual a 1 especifica que o elemento de sintaxe poc_reset_idc está presente nos

cabeçalhos de segmento de fatia das fatias que se referem ao PPS.

[0090] Além disso, `pps_extension2_flag` igual a 0 especifica que nenhum elemento de sintaxe `pps_extension_data_flag` está presente na estrutura de sintaxe PPS RBSP. `pps_extension2_flag` será igual a 0 em fluxos de bits de acordo com esta versão desta especificação. O valor de 1 para `pps_extension2_flag` é reservado para uso futuro pela ITU-T | ISO / IEC. Decodificadores devem ignorar todos os dados que seguem o valor 1 para `pps_extension2_flag` em uma unidade PPS NAL.

[0091] A tabela seguinte é um exemplo da sintaxe de um cabeçalho de fatia, de acordo com as técnicas desta divulgação:

slice_segment_header() {	Descriptor
...	
if(!dependent_slice_segment_flag) {	
i = 0	
if(num_extra_slice_header_bits > i) {	
i++	
discardable_flag	u(1)
}	
if(num_extra_slice_header_bits > i) {	
i++	
cross_layer_bla_flag	u(1)
}	
...	
if(slice_segment_header_extension_present_flag) {	
slice_segment_header_extension_length	ue(v)
if(poc_reset_info_present_flag)	
poc_reset_idc	u(2)
if(poc_reset_idc > 0)	
poc_reset_period_id	u(6)
if(poc_reset_idc == 3)	
full_poc_reset_flag	u(1)
<i>if((poc_reset_idc == 3) (nuh_layer_id == 0 && poc_reset_idc == 2))</i>	
poc_lsb_val	u(v)
if((poc_reset_info_present_flag && poc_reset_idc == 0) (poc_reset_idc == 3 && (log2_max_pic_order_cnt_lsb_minus4 + 5) % 8 != 0) (nuh_layer_id == 0 && poc_reset_idc == 2 && (log2_max_pic_order_cnt_lsb_minus4 + 4) % 8 != 0))	
slice_segment_header_extension_reserved_bits	u(v)
for(i = BytesInSliceSegmtHdrExt; i < slice_segment_header_extension_length; i++)	
slice_segment_header_extension_data_byte[i]	u(8)
}	
byte_alignment()	
}	

[0092] Em alternativa, o elemento de sintaxe `poc_reset_period_id` pode ser sinalizado com um número diferente de bits, por exemplo, codificado como ?(14).

[0093] Quando presente, o valor dos elementos de sintaxe do cabeçalho do segmento de fatia `slice_pic_parameter_set_id`, `pic_output_flag`, `no_output_of_prior_pics_flag`, `slice_pic_order_cnt_lsb`, `short_term_ref_pic_set_sps_flag`, `short_term_ref_pic_set_idx`, `num_long_term_sps`, `num_long_term_pics`, `slice_temporal_mvp_enabled_flag`, `discardable_flag`, `cross_layer_bla_flag`, `inter_layer_pred_enabled_flag`, `num_inter_layer_ref_pics_minus1`, `poc_reset_idc`, `poc_reset_pic_id`, `full_poc_reset_flag` e `poc_lsb_val` deve ser o mesmo em todos os cabeçalhos do segmento de fatia de uma imagem codificada. Quando presente, o valor dos elementos de sintaxe de cabeçalho de segmento de fatia `lt_idx_sps [i]`, `poc_lsb_lt [i]`, `used_by_curr_pic_lt_flag [i]`, `delta_poc_msb_present_flag [i]`, `delta_poc_msb_cycle_lt [i]`, e `inter_layer_pred_layer_idc [i]` deve ser o mesmo em todos os cabeçalhos do segmento de fatia de uma imagem codificada para cada possível valor de `i`.

[0094] O elemento de sintaxe `slice_segment_header_extension_length` pode especificar o comprimento dos dados de extensão de cabeçalho de segmento de fatia em bytes, não incluindo os bits usados para sinalização de `slice_segment_header_extension_length` `si`. E se `poc_reset_info_present_flag` é igual a 0, o valor do elemento de sintaxe `slice_segment_header_extension_length` deve estar na faixa de 0 a 256, inclusive. Caso contrário, o valor do elemento de sintaxe `slice_segment_header_extension_length` deve estar no intervalo de 1 a 256, inclusive, quando `poc_reset_idc` é

igual a 1, ou quando poc_reset_idc é igual a 2 e nuh_layer_id é maior do que 0, e na faixa de $1 + \text{Ceil}((\log_2 \text{max_pic_order_cnt_lsb_minus4} + 5) / 8)$ a 256, inclusive, quando poc_reset_idc é igual a 2 e nuh_layer_id é igual a 0, e na faixa de $1 + ((\text{Ceil} \log_2 \text{max_pic_order_cnt_lsb_minus4} + 5) / 8)$ a 256, inclusive, quando poc_reset_idc é igual a 3.

[0095] O elemento de sintaxe poc_reset_idc igual a 0 especifica que nem os bits mais significativos nem os bits menos significativos do valor de contagem de ordem de imagem para a imagem atual são redefinidos. O elemento de sintaxe poc_reset_idc igual a 1 especifica que apenas os bits mais significativos do valor de contagem de ordem de imagem para a imagem atual pode ser redefinido. O elemento de sintaxe poc_reset_idc igual a 2 especifica que ambos os bits mais significativos e os bits menos significativos do valor de contagem de ordem de imagem para a imagem atual podem ser redefinidos. O elemento de sintaxe poc_reset_idc igual a 3 especifica que ou apenas os bits mais significativos ou ambos os bits mais significativos e os bits menos significativos do valor de contagem de ordem de imagem para a imagem atual podem ser redefinidos e informação de contagem de ordem de imagem adicional é sinalizada. Quando não está presente, o valor do elemento de sintaxe poc_reset_poc é inferido (isto é, determinado, por um codificador, sem sinalização explícita) para ser igual a 0.

[0096] É uma exigência de conformidade de fluxo de bits que as seguintes restrições se apliquem:

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

- O valor de poc_reset_idc de todas as imagens em uma unidade de acesso deve ser o mesmo. [Removidos: - Quando o valor do poc_reset_idc de uma imagem em uma unidade de acesso é igual a 1, o valor de poc_reset_idc de todas as imagens na unidade de acesso deve ser igual a 1.] [removidos: - Quando o valor de um poc_reset_idc de uma imagem em uma unidade de acesso é igual a 2, o valor de poc_reset_idc de todas as imagens na unidade de acesso deve ser igual a 2.]

- Quando uma imagem em uma unidade de acesso é uma imagem IRAP com um valor particular de nal_unit_type e há pelo menos uma outra imagem da mesma unidade de acesso com um valor diferente de nal_unit_type, o valor de poc_reset_idc será igual a 1 ou 2 para todas as imagens da unidade de acesso.

- Quando a imagem com nuh_layer_id igual a 0 em uma unidade de acesso é uma imagem IDR e há pelo menos uma imagem não-IDR na mesma unidade de acesso, o valor de poc_reset_idc será igual a 2 para todas as imagens da unidade de acesso.

- Quando a imagem com nuh_layer_id igual a 0 em uma unidade de acesso não é uma imagem IDR, o valor de poc_reset_idc não deve ser igual a 2 para qualquer imagem na unidade de acesso.

[0097] O valor de poc_reset_idc de uma unidade de acesso refere-se ao valor de poc_reset_idc das imagens na unidade de acesso.

[0098] O elemento de sintaxe 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 tem o mesmo valor do elemento de sintaxe poc_reset_period_id e do elemento de sintaxe poc_reset_idc igual a 1 ou 2.

OBSERVAÇÃO - Não é proibido para várias imagens em uma camada ter o mesmo valor de poc_reset_pic_id e ter poc_reset_idc igual a 1 ou 2 a menos que tais imagens ocorram em duas unidades de acesso consecutivas na ordem de decodificação. Para minimizar a probabilidade de tais duas imagens aparecerem no fluxo de bits devido a perdas de imagem, extração de fluxo de bits, busca, ou operações de junção, codificadores devem definir o valor de poc_reset_pic_id para ser um valor aleatório para cada período de redefinição de POC (sujeito às limitações acima especificadas).

[0099] 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 de 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 será a primeira unidade de acesso em um período de redefinição de POC.

[Removidos: - Quando as imagens com poc_reset_idc igual a 1 e 3 estão presentes no mesmo período de redefinição de POC, todas as imagens com poc_reset_idc igual a 3 devem seguir a imagem com poc_reset_idc igual a 1, em ordem de decodificação]

[Removidos: - Quando as imagens com poc_reset_idc igual a 2 e 3 estão presentes no mesmo período de redefinição de POC, todas as imagens com poc_reset_idc igual a 3 devem seguir a imagem com poc_reset_idc igual a 2, em ordem de decodificação]

[0100] o elemento de sintaxe full_poc_reset_flag igual a 1 especifica que ambos os bits mais significativos e 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. O elemento de sintaxe `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.

[0101] O elemento de sintaxe `poc_lsb_val` especifica um valor que pode ser utilizado para derivar a contagem de ordem de imagem da imagem atual. O comprimento do elemento de sintaxe `poc_lsb_val` é $\log_2_{\max_pic_order_cnt_lsb_minus4} + 4$ bits.

[0102] É uma exigência da conformidade de fluxo de bits que, quando `poc_reset_idc` é igual a 3, e a imagem anterior `picA` em ordem de decodificação que está na mesma camada que a imagem atual, tem `poc_reset_idc` igual a 1 ou 2, que pertence ao mesmo período de redefinição de POC está presente no fluxo de bits, `picA` será a mesma imagem da imagem anterior, em ordem de decodificação que está na mesma camada que a imagem atual, que não é uma imagem RASL, uma imagem RADL ou uma imagem de não-referência de subcamada, e que tem `TemporalId` igual a 0, e o valor de `poc_lsb_val` da imagem atual é igual ao valor de `slice_pic_order_cnt_lsb` de `picA`.

[0103] As variáveis `numRsvBits` e `BytesInSliceSegmtHdrExt` são derivadas da seguinte forma:

```

    se(!poc_reset_info_present_flag) {
        numRsvBits = 0
        BytesInSliceSegmtHdrExt = 0
    } mesmo se(poc_reset_idc == 0) {
        numRsvBits = 6
        BytesInSliceSegmtHdrExt = 1
    }
    mesmo se(poc_reset_idc == 2 && nuh_layer_id > 0) || poc_reset_idc == 1) {
        numRsvBits = 0
        BytesInSliceSegmtHdrExt = 1
    }
    mesmo se(poc_reset_idc == 2 && nuh_layer_id == 0) {
        numRsvBits = 8 - (log2_max_pic_order_cnt_lsb_minus4 + 4) % 8
        bitsInSliceSegmtHdrExt = 1 + Ceil((log2_max_pic_order_cnt_lsb_minus4 + 4) / 8)
    }
    mesmo { // poc_reset_idc == 3
        numRsvBits = 8 - (log2_max_pic_order_cnt_lsb_minus4 + 5) % 8
        bitsInSliceSegmtHdrExt = 1 + Ceil((log2_max_pic_order_cnt_lsb_minus4 + 5) / 8)
    }

```

[0104] O elemento de sintaxe `slice_segment_header_extension_reserved_bits` pode ter qualquer valor. O comprimento do elemento de sintaxe `slice_segment_header_extension_reserved_bits` é igual a `numRsvBits` bits. Decodificadores, tal como decodificador de vídeo 30, deve ignorar o valor de `slice_segment_header_extension_reserved_bits`. Seu valor não afeta o decodificador em conformidade com perfis especificados nesta versão desta especificação.

[0105] O decodificador de vídeo 30 pode executar o seguinte processo de decodificação para iniciar a decodificação de uma imagem codificada com `nuh_layer_id`

maior que 0. Cada "imagem" referida no processo a seguir é uma imagem codificada completa. O processo de decodificação funciona da seguinte maneira para a imagem atual CurrPic:

- Decodificar unidades NAL, tal como aqui descrito.

- Os processos na subcláusula F.8.3 especificam os seguintes processos de decodificação utilizando elementos de sintaxe na camada de segmento de fatia e acima:

- Variáveis e funções relacionadas à contagem de ordem de imagem são derivadas na subcláusula F.8.3.1. Isso precisa ser invocado apenas para o primeiro segmento de fatia de uma imagem. É uma exigência de conformidade de fluxo de bits que PicOrderCntVal deve permanecer inalterada dentro de uma unidade de acesso. É também uma exigência de conformidade de fluxo de bits que DeltaPocVal deve permanecer inalterada dentro de uma unidade de acesso.

- O processo de decodificação para RPS na subcláusula F.8.3.2 é invocado, em que apenas imagens de referência com uma nuh_layer_id igual a de CurrPic podem ser marcadas como "não utilizadas para referência" ou "usadas para referência a longo prazo" e qualquer imagem com um valor diferente de nuh_layer_id não está marcada. Isso precisa ser invocado apenas para o primeiro segmento de fatia de uma imagem.

- Quando FirstPicInLayerDecodedFlag [nuh_layer_id] é igual a 0, o processo de decodificação para gerar imagens de referência indisponíveis especificadas na subcláusula F.8.1.3 é invocado, que precisa ser invocado apenas para o primeiro segmento de fatia de uma imagem.

- Quando FirstPicInLayerDecodedFlag [nuh_layer_id] não é igual a 0 e a imagem atual é um imagem

IRAP com NoRaslOutputFlag igual a 1, o processo de decodificação para gerar imagens de referência indisponíveis especificadas na subcláusula F.8.1.3 é invocado, que só precisa ser invocado para o primeiro segmento de fatia de uma imagem.

[0106] Em um primeiro exemplo, de acordo com aspectos da presente divulgação, decodificador de vídeo 30 pode executar o seguinte processo de decodificação para valores de POC. O primeiro exemplo pode corresponder à primeira técnica exemplar para diminuir os valores de POC descritos acima. Isto é, o exemplo que se segue pode ser utilizado para diminuir os valores de todas as imagens POC (de todas as camadas) armazenadas em um DPB antes da decodificação de uma imagem inicial de uma unidade de acesso em que redefinição de POC é realizada. O processo de diminuição de POC pode ser invocado antes de decodificar a imagem inicial para a unidade de acesso, mas depois de codificação de informações de cabeçalho de fatia da primeira fatia da imagem inicial. O primeiro exemplo de processo pode ser realizado como se segue:

Saída desse processo é PicOrderCntVal, a contagem de ordem de imagem da imagem atual.

Contagens de ordem de imagem são usadas para identificar imagens, para derivar parâmetros de movimento no modo de mesclagem e vetor de movimento de predição, e para decodificador verificação de conformidade (ver subcláusula C.5).

Cada imagem codificada está associada com uma variável de contagem de ordem de imagem, denotado como PicOrderCntVal.

Se FirstPicInLayerDecodedFlag [nuh_layer_id] é igual a 1, poc_reset_idc é maior que 0, e a unidade de acesso contendo a imagem atual é a primeira unidade de

acesso, em ordem de decodificação em um período de redefinição de POC, o seguinte é aplicável:

- As variáveis pocMsbDelta, pocLsbDelta e DeltaPocVal são derivadas da seguinte forma:
prevPicOrderCntLsb =

```

prevPicOrderCntLsb =
PrevPicOrderCnt[ nuh_layer_id ] & ( MaxPicOrderCntLsb - 1 )
prevPicOrderCntMsb = PrevPicOrderCnt[ nuh_layer_id ] - prevPicOrderCntLsb
se( poc_reset_idc == 3 || ( poc_reset_idc == 2 && nuh_layer_id == 0 ) )
pocLsbVal = poc_lsb_val

```

mesmo

```

pocLsbVal = slice_pic_order_cnt_lsb
pocMsbDelta = getCurrMsb( pocLsbVal, prevPicOrderCntLsb,
prevPicOrderCntMsb, MaxPicOrderCntLsb )
se( poc_reset_idc == 2 || ( poc_reset_idc == 3 && full_poc_reset_flag ) )
pocLsbDelta = pocLsbVal

```

mesmo

```

pocLsbDelta = 0
DeltaPocVal = pocMsbDelta + pocLsbDelta

```

- O PicOrderCntVal da imagem atual é derivado como segue

```

se( poc_reset_idc == 1 )
PicOrderCntVal = slice_pic_order_cnt_lsb
mesmo se( poc_reset_idc == 2 )
PicOrderCntVal = 0
mesmo { // poc_reset_idc == 3
PicOrderCntMsb = getCurrMsb( slice_pic_order_cnt_lsb, poc_lsb_val, 0,
MaxPicOrderCntLsb )
PicOrderCntVal = PicOrderCntMsb + slice_pic_order_cnt_lsb
}

```

- O valor de `PrevPicOrderCnt[nuh_layer_id]` é derivado como segue:

- Se a imagem atual não é uma imagem RASL, uma imagem RADL, ou uma imagem de não-referência de subcamada, e a imagem atual tem `TemporalId` igual a 0, `PrevPicOrderCnt[nuh_layer_id]` é definida igual a `PicOrderCntVal`.

- Caso contrário `poc_reset_idc` é igual a 3. `PrevPicOrderCnt[nuh_layer_id]` é igual a `full_poc_reset_flag` setembro? 0: `poc_lsb_val`.

Caso contrário, o seguinte é aplicável:

- O valor da `DeltaPocVal` é definido igual a 0.

- O `PicOrderCntVal` da imagem atual é derivado da seguinte forma:

```

se( !FirstPicInLayerDecodedFlag[ nuh_layer_id ] ) {
    se( poc_reset_idc == 1 )
        PicOrderCntVal = slice_pic_order_cnt_lsb
mesmo se( poc_reset_idc == 2 )
        PicOrderCntVal = 0
mesmo se( poc_reset_idc == 3 ) {
        PicOrderCntMsb = getCurrMsb(slice_pic_order_cnt_lsb, poc_lsb_val, 0,
MaxPicOrderCntLsb )
        PicOrderCntVal = PicOrderCntMsb + slice_pic_order_cnt_lsb
} mesmo // a imagem atual é uma imagem IRAP com NoRaslOutputFlag igual a
1 // a imagem atual é uma imagem IRAP com
        PicOrderCntVal = slice_pic_order_cnt_lsb
} mesmo { //a derivação de POC como em HEVC versão 1
    se(a imagem atual é um imagem IRAP com NoRaslOutputFlag igual a 1)
        PicOrderCntMsb = 0
    mesmo {
        prevPicOrderCntLsb =
PrevPicOrderCnt[ nuh_layer_id ] & ( MaxPicOrderCntLsb - 1 )
        prevPicOrderCntMsb =
PrevPicOrderCnt[ nuh_layer_id ] - prevPicOrderCntLsb
        PicOrderCntMsb = getCurrMsb( slice_pic_order_cnt_lsb,
prevPicOrderCntLsb,
prevPicOrderCntMsb, MaxPicOrderCntLsb )
    }
    PicOrderCntVal = PicOrderCntMsb + slice_pic_order_cnt_lsb
}

```

- O valor de PrevPicOrderCnt [nuh_layer_id] é derivado como segue:

- Se a imagem atual não é uma imagem RASL, uma imagem RADL, ou uma imagem de não-referência de subcamada, ea imagem atual tem TemporalId igual a 0, PrevPicOrderCnt [nuh_layer_id] está definida igual a PicOrderCntVal.

- Caso contrário, quando FirstPicInLayerDecodedFlag[nuh_layer_id] é igual a 0 e

`poc_reset_idc` é igual a 3, `PrevPicOrderCnt [nuh_layer_id]` está definida igual a `full_poc_reset_flag? 0: poc_lsb_val`.

Se a imagem atual é a primeira imagem em uma unidade de acesso (ou seja, a imagem com o valor mais baixo de `nuh_layer_id` entre todas as imagens da unidade de acesso) e `DeltaPocVal` é maior do que 0, os valores `PicOrderCntVal` de todas as imagens em todos os sub-DPBs são diminuídos pelo valor do `DeltaPocVal`.

O valor de `PicOrderCntVal` deve estar na faixa de -2^{31} a $2^{31} - 1$, inclusive. Em um CVS, os valores `PicOrderCntVal` para quaisquer duas imagens codificadas na mesma camada não devem ser os mesmos.

A função `PicOrderCnt (picX)` é especificada como segue:

`PicOrderCnt(picX) = PicOrderCntVal da imagem picX`
(M-25)

A função `DiffPicOrderCnt(picA, picB)` é especificada como segue:

`DiffPicOrderCnt(picA, picB) = PicOrderCnt(picA) - PicOrderCnt(picB)` (F-26)

O fluxo de bits não deve conter dados que resultam em valores de `DiffPicOrderCnt(picA, picB)` utilizados no processo de decodificação que não estão na faixa de -2^{15} a $2^{15} - 1$, inclusive.

OBSERVAÇÃO - Seja X a imagem atual e Y e Z as duas outras imagens na mesma sequência, Y e Z são considerados na mesma direção em ordem de emissão de X, quando ambos `DiffPicOrderCnt (X, Y)` e `DiffPicOrderCnt (X, Z)` são positivos ou ambos são negativos.

[0107] Em um segundo exemplo, de acordo com aspectos da presente divulgação, decodificador de vídeo 30 pode executar o seguinte processo de decodificação para valores de POC. O segundo exemplo pode corresponder à

segunda técnica exemplar para diminuir os valores de POC descritos acima. Ou seja, o exemplo a seguir pode ser usado para chamar o processo de diminuição de POC de imagens anteriores em cada sub-DPB antes de decodificar cada imagem na unidade de acesso, mas após a análise e decodificação da informação do cabeçalho de fatia da primeira fatia da imagem. O decodificador de vídeo 30 também pode executar um processo de diminuição de POC adicional de imagens anteriores em um sub-DPB para uma camada que tem uma imagem que não está presente ou é não recebida pela unidade de acesso depois de decodificar as imagens que estão presentes na unidade de acesso. O segundo exemplo de processo pode ser realizado como se segue:

Saída desse processo é `PicOrderCntVal`, a contagem de ordem de imagem da imagem atual.

Contagens de ordem de imagem são usadas para identificar imagens, para derivar parâmetros de movimento no modo de mesclagem e vetor de movimento de predição, e para decodificador verificação de conformidade (ver subcláusula C.5).

Cada imagem codificada está associada com uma variável de contagem em ordem de imagem, denotado como `PicOrderCntVal`.

Se a imagem atual é a primeira imagem na unidade de acesso, o seguinte é aplicável:

- O valor da `PocResetFlag` é definido como 0.
- O `DeltaPocVal` variável é definido como 0.
- A variável `UpdateSubDpbFlag [i]` é definida igual a 0 para todos os valores de `i` de 0 a 63, inclusive.

Se `FirstPicInLayerDecodedFlag [nuh_layer_id]` é igual a 1, `poc_reset_idc` é maior que 0, e a unidade de acesso contendo a imagem atual é a primeira unidade de

acesso, em ordem de decodificação em um período de redefinição de POC, o seguinte é aplicável:

Se `FirstPicInLayerDecodedFlag [nuh_layer_id]` é igual a 1, `poc_reset_idc` é maior que 0, e a unidade de acesso contendo a imagem atual é a primeira unidade de acesso, em ordem de decodificação em um período de redefinição de POC, o seguinte é aplicável:

- As variáveis `pocMsbDelta`, `pocLsbDelta` e `DeltaPocVal` são derivadas da seguinte forma:

```

prevPicOrderCntLsb =
PrevPicOrderCnt[ nuh_layer_id ] & ( MaxPicOrderCntLsb - 1 )
prevPicOrderCntMsb = PrevPicOrderCnt[ nuh_layer_id ] -- prevPicOrderCntLsb
if( poc_reset_idc == 3 || ( poc_reset_idc == 2 && nuh_layer_id == 0 ) )
    pocLsbVal = poc_lsb_val
else
    pocLsbVal = slice_pic_order_cnt_lsb
pocMsbDelta = getCurrMsb( pocLsbVal, prevPicOrderCntLsb,
prevPicOrderCntMsb, MaxPicOrderCntLsb )
if( poc_reset_idc == 2 || ( poc_reset_idc == 3 && full_poc_reset_flag ) )
    pocLsbDelta = pocLsbVal
else
    pocLsbDelta = 0
DeltaPocVal = pocMsbDelta + pocLsbDelta

```

- A `PicOrderCntVal` de cada imagem que está no DPB e pertence à mesma camada que a imagem atual é diminuída em `DeltaPocVal`.

- `PocResetFlag` = 1
- `UpdateSubDpbFlag [nuh_layer_id]` = 1
- O `PicOrderCntVal` da imagem atual é derivada da seguinte forma:

```

Se ( poc_reset_idc == 1 )
PicOrderCntVal = slice_pic_order_cnt_lsb

```

```

Mesmo se (poc_reset_idc == 2)
PicOrderCntVal = 0
Mesmo { // poc_reset_idc == 3 =
PicOrderCntMsb = getCurrMsb
(slice_pic_order_cnt_lsb, poc_lsb_val, 0,
MaxPicOrderCntLsb)
PicOrderCntVal = PicOrderCntMsb +
slice_pic_order_cnt_lsb
}

```

- O valor de PrevPicOrderCnt [nuh_layer_id] é derivado como segue:

- Se a imagem atual não é uma imagem RASL, uma imagem RADL, ou uma imagem de não-referência de subcamada, e a imagem atual tem TemporalId igual a 0, PrevPicOrderCnt [nuh_layer_id] está definida igual a PicOrderCntVal.

- Caso contrário poc_reset_idc Quando é igual a 3 PrevPicOrderCnt [nuh_layer_id] é igual a full_poc_reset_flag setembro? 0: poc_lsb_val.

Caso contrário, o seguinte é aplicável:

- O valor da DeltaPocVal está definido igual a 0

- O PicOrderCntVal da imagem atual é derivado da seguinte forma:

```

se (! FirstPicInLayerDecodedFlag [nuh_layer_id])
{

```

```

Se (poc_reset_idc == 1)

```

```

PicOrderCntVal = slice_pic_order_cnt_lsb

```

```

Mesmo se (poc_reset_idc == 2)

```

```

PicOrderCntVal = 0

```

```

Mesmo se (poc_reset_idc == 3) {

```

```

PicOrderCntMsb = getCurrMsb

```

```

(slice_pic_order_cnt_lsb, poc_lsb_val, 0,
MaxPicOrderCntLsb)

```

```

        PicOrderCntVal          =          PicOrderCntMsb          +
slice_pic_order_cnt_lsb
    } mesmo // a imagem atual é uma imagem IRAP com
NoRaslOutputFlag igual a 1
        PicOrderCntVal = slice_pic_order_cnt_lsb
    } mesmo { // a derivação de POC como em HEVC
versão 1
        se (a imagem atual é uma imagem IRAP com
NoRaslOutputFlag igual a 1) PicOrderCntMsb = 0
        mesmo {
            prevPicOrderCntLsb =
            PrevPicOrderCnt          [nuh_layer_id]          &
(MaxPicOrderCntLsb 1)
            prevPicOrderCntMsb =
            PrevPicOrderCnt          [nuh_layer_id]          -
prevPicOrderCntLsb
            PicOrderCntMsb          =          getCurrMsb
(slice_pic_order_cnt_lsb,
            prevPicOrderCntLsb,
            prevPicOrderCntMsb, MaxPicOrderCntLsb)
        }
        PicOrderCntVal          =          PicOrderCntMsb          +
slice_pic_order_cnt_lsb
    }

```

- O valor de PrevPicOrderCnt [nuh_layer_id] é derivado como segue:

- Se a imagem atual não é uma imagem RASL, uma imagem RADL, ou uma imagem de não-referência de subcamada, e a imagem atual tem TemporalId igual a 0, PrevPicOrderCnt [nuh_layer_id] está definida igual a PicOrderCntVal.

- Caso contrário, quando FirstPicInLayerDecodedFlag [nuh_layer_id] é igual a 0 e

`poc_reset_idc` é igual a 3, `PrevPicOrderCnt [nuh_layer_id]` está definida igual a `full_poc_reset_flag? 0: poc_lsb_val`.

Se a imagem atual é a última imagem em uma unidade de acesso e o valor da variável `PocResetFlag` é igual a 1, o seguinte é aplicável:

- Para todos os valores de `i` de 0 a 63, inclusive, se `UpdateSubDpbFlag [i]` é igual a 0, o `PicOrderCntVal` de todas as imagens no DPB que `nuh_layer_id` igual a `i` é diminuído pelo valor da variável `DeltaPocVal`

O valor de `PicOrderCntVal` deve estar na faixa de -2^{31} a $2^{31} - 1$, inclusive. Em um CVS, os valores `PicOrderCntVal` para quaisquer duas imagens codificadas na mesma camada não devem ser os mesmos.

A função `PicOrderCnt (picX)` é especificada como segue:

`PicOrderCnt (picX) = PicOrderCntVal` da imagem `picX` (M-25)

A função `DiffPicOrderCnt(picA, picB)` é especificada como segue:

`DiffPicOrderCnt(picA, picB) = PicOrderCnt(picA) - PicOrderCnt(picB)` (F- 26)

O fluxo de bits não deve conter dados que resultam em valores de `DiffPicOrderCnt (PICA, PICB)` utilizados no processo de decodificação que não estão na faixa de -2^{15} a $2^{15} - 1$, inclusive.

OBSERVAÇÃO - Seja X a imagem atual e Y e Z as duas outras imagens na mesma sequência, Y e Z são considerados na mesma direção em ordem de emissão de X, quando ambos `DiffPicOrderCnt (X, Y)` e `DiffPicOrderCnt (X, Z)` são positivos ou ambos são negativos.

[0108] Em um terceiro exemplo, de acordo com aspectos da presente divulgação, decodificador de vídeo 30 pode executar o seguinte processo de decodificação para

valores de POC. O terceiro exemplo pode corresponder à terceira técnica exemplar para diminuir os valores de POC descritos acima. Ou seja, o exemplo a seguir pode ser usado para chamar o processo de diminuição de POC de imagens anteriores em cada sub-DPB antes da decodificação de cada imagem na unidade de acesso, mas após a análise e decodificação da informação do cabeçalho da fatia da primeira fatia da respectiva imagem. Antes do processo de diminuição de POC de imagens anteriores em um sub-DPB ser executado, se o mesmo processo que ainda não foi realizado para um ou mais DPBs de subcamada inferior, o processo é realizado para os sub-DPBs. O terceiro exemplo de processo pode ser realizado como se segue:

Saída desse processo é `PicOrderCntVal`, a contagem de ordem de imagem da imagem atual.

Contagens de ordem de imagem são usadas para identificar imagens, para derivar parâmetros de movimento no modo de mesclagem e vetor de movimento de predição, e para decodificador verificação de conformidade (ver subcláusula C.5).

Cada imagem codificada está associada com uma variável de contagem de ordem de imagem, denotada como `PicOrderCntVal`.

Se a imagem atual é a primeira imagem na unidade de acesso, o seguinte é aplicável:

- O valor da `PocResetFlag` é definido como 0.
- O `DeltaPocVal` variável é definido como 0.
- A variável `UpdateSubDpbFlag [i]` está definida igual a 0 para todos os valores de `i` de 0 a 63, inclusive.

Se `FirstPicInLayerDecodedFlag [nuh_layer_id]` é igual a 1, `poc_reset_idc` é maior que 0, e a unidade de acesso contendo a imagem atual é a primeira unidade de

acesso, em ordem de decodificação em um período de redefinição de POC, o seguinte é aplicável:

- As variáveis pocMsbDelta, pocLsbDelta e DeltaPocVal são derivadas da seguinte forma:
prevPicOrderCntLsb =

```

PrevPicOrderCnt [nuh_layer_id] &
(MaxPicOrderCntLsb 1)
prevPicOrderCntMsb PrevPicOrderCnt =
[nuh_layer_id] t prevPicOrderCntLsb if (poc_reset_idc == 3
| | (2 && poc_reset_idc == 0 == nuh_layer_id)) pocLsbVal =
poc_lsb_val

```

outro

```
pocLsbVal = slice_pic_order_cnt_lsb
```

```
pocMsbDelta = getCurrMsb (pocLsbVal,
prevPicOrderCntLsb,
```

```
prevPicOrderCntMsb, MaxPicOrderCntLsb)
```

```
Se (poc_reset_idc == 2 || (poc_reset_idc == 3 &&
full_poc_reset_flag)) pocLsbDelta = pocLsbVal
```

outro

```
pocLsbDelta = 0
```

```
DeltaPocVal = pocMsbDelta + pocLsbDelta
```

```
- PocResetFlag = 1
```

```
- UpdateSubDpbFlag [nuh_layer_id] = 1
```

- Para todos os valores de i de 0 a nuh_layer_id-1, inclusive, se UpdateSubDpbFlag [i] é igual a 0, então o PicOrderCntVal de cada imagem que está no DPB e pertencente à mesma camada que a imagem atual é decrementada por DeltaPocVal.

- O PicOrderCntVal de cada imagem que está no DPB e pertence à mesma camada que a imagem atual é diminuída em DeltaPocVal.

- O PicOrderCntVal da imagem atual é derivada da seguinte forma:

```

Se (poc_reset_idc == 1)
PicOrderCntVal = slice_pic_order_cnt_lsb
mesmo se (poc_reset_idc == 2)
PicOrderCntVal = 0
Mesmo { // poc_reset_idc == 3
PicOrderCntMsb = getCurrMsb
(slice_pic_order_cnt_lsb, poc_lsb_val, 0,
MaxPicOrderCntLsb)
PicOrderCntVal = PicOrderCntMsb +
slice_pic_order_cnt_lsb
}

```

- O valor de PrevPicOrderCnt [nuh_layer_id] é derivado como segue:

- Se a imagem atual não é uma imagem RASL, uma imagem RADL, ou uma imagem de não-referência de subcamada, e a imagem atual tem TemporalId igual a 0, PrevPicOrderCnt [nuh_layer_id] está definida igual a PicOrderCntVal.

- Caso contrário poc_reset_idc Quando é igual a 3 PrevPicOrderCnt [nuh_layer_id] é igual a full_poc_reset_flag? 0: poc_lsb_val.

Caso contrário, o seguinte é aplicável:

- O valor da DeltaPocVal está definido igual a 0

- O PicOrderCntVal da imagem atual é derivada da seguinte forma:

```

se (!FirstPicInLayerDecodedFlag [nuh_layer_id]) {
Se (poc_reset_idc == 1)
PicOrderCntVal = slice_pic_order_cnt_lsb
else if (poc_reset_idc == 2)
PicOrderCntVal = 0
else if (poc_reset_idc == 3) {
PicOrderCntMsb = getCurrMsb
(slice_pic_order_cnt_lsb, poc_lsb_val, 0,
MaxPicOrderCntLsb)
}
}

```

```

        PicOrderCntVal          =          PicOrderCntMsb          +
slice_pic_order_cnt_lsb
    } mesmo // a imagem atual é uma imagem IRAP com
NoRaslOutputFlag igual a 1
        PicOrderCntVal = slice_pic_order_cnt_lsb
    } mesmo { // a derivação de POC como em HEVC
versão 1
        se (a imagem atual é uma imagem IRAP com
NoRaslOutputFlag igual a 1) PicOrderCntMsb = 0
        mesmo {
            prevPicOrderCntLsb =
            PrevPicOrderCnt          [nuh_layer_id]          &
(MaxPicOrderCntLsb 1)
            prevPicOrderCntMsb =
            PrevPicOrderCnt          [nuh_layer_id]          em
prevPicOrderCntLsb
            PicOrderCntMsb          =          getCurrMsb
(slice_pic_order_cnt_lsb,
            prevPicOrderCntLsb,
            prevPicOrderCntMsb, MaxPicOrderCntLsb)
        }
        PicOrderCntVal          =          PicOrderCntMsb          +
slice_pic_order_cnt_lsb
    }

```

- O valor de PrevPicOrderCnt [nuh_layer_id] é derivado como segue:

- Se a imagem atual não é uma imagem RASL, uma imagem RADL, ou uma imagem de não-referência de subcamada, e a imagem atual tem TemporalId igual a 0, PrevPicOrderCnt [nuh_layer_id] está definida igual a PicOrderCntVal.

- Caso contrário, quando FirstPicInLayerDecodedFlag [nuh_layer_id] é igual a 0 e

`poc_reset_idc` é igual a 3, `PrevPicOrderCnt [nuh_layer_id]` está definida igual a `full_poc_reset_flag? 0: poc_lsb_val`.

O valor de `PicOrderCntVal` deve estar na faixa de -2^{31} a $2^{31} - 1$, inclusive. Em um CVS, os valores `PicOrderCntVal` para quaisquer duas imagens codificadas na mesma camada não devem ser os mesmos.

A função `PicOrderCnt (picX)` é especificada como segue:

`PicOrderCnt (picX) = PicOrderCntVal da imagem picX (M-25)`

A função `DiffPicOrderCnt(picA, picB)` é especificada como segue:

`DiffPicOrderCnt(picA, picB) = PicOrderCnt(picA) - PicOrderCnt(picB) (F-26)`

O fluxo de bits não deve conter dados que resultam em valores de `DiffPicOrderCnt (picA, picB)` utilizados no processo de decodificação que não estão na faixa de -2^{15} a $2^{15} - 1$, inclusive.

OBSERVAÇÃO - Seja X a imagem atual e Y e Z as duas outras imagens na mesma sequência, Y e Z são considerados na mesma direção em ordem de emissão de X, quando ambos `DiffPicOrderCnt (X, Y)` e `DiffPicOrderCnt (X, Z)` são positivos ou ambos são negativos.

[0109] Em um quarto exemplo, de acordo com aspectos da presente divulgação, decodificador de vídeo 30 pode executar o seguinte processo de decodificação para valores de POC. O quarto exemplo pode corresponder à quarta técnica exemplar para diminuir os valores de POC descritos acima. Isto é, o exemplo que se segue pode ser usado para realizar o processo de diminuição de POC de imagens anteriores no DPB (isto é, para todos os sub-DPBs) é aplicado uma vez, no final de decodificação de cada unidade de acesso que requer redefinição de POC. Ou seja,

decodificador de vídeo 30 pode decodificar todas as imagens de uma unidade de acesso com redefinição de POC sendo realizada, seguido por diminuir os valores de POC de todas as imagens do DPB. O quarto exemplo de processo pode ser realizado como se segue:

Saída desse processo é PicOrderCntVal, a contagem de ordem de imagem da imagem atual.

Contagens de ordem de imagem são usadas para identificar imagens, para derivar parâmetros de movimento no modo de mesclagem e vetor de movimento de predição, e para decodificador verificação de conformidade (ver subcláusula C.5).

Cada imagem codificada está associada com uma variável de contagem em ordem de imagem, denotado como PicOrderCntVal.

Se FirstPicInLayerDecodedFlag [nuh_layer_id] é igual a 1, poc_reset_idc é maior que 0, e a unidade de acesso contendo a imagem atual é a primeira unidade de acesso, em ordem de decodificação em um período de redefinição de POC, o seguinte é aplicável:

- As variáveis pocMsbDelta, pocLsbDelta e DeltaPocVal são derivadas da seguinte forma:

prevPicOrderCntLsb =

PrevPicOrderCnt [nuh_layer_id] &
(MaxPicOrderCntLsb 1)

prevPicOrderCntMsb PrevPicOrderCnt =
[nuh_layer_id] t prevPicOrderCntLsb

se (poc_reset_idc == 3 || (poc_reset_idc == 2
&& nuh_layer_id == 0)) pocLsbVal poc_lsb_val

mesmo

pocLsbVal = slice_pic_order_cnt_lsb

pocMsbDelta = getCurrMsb (pocLsbVal,
prevPicOrderCntLsb,

```

prevPicOrderCntMsb, MaxPicOrderCntLsb)
se (poc_reset_idc == 2 || (poc_reset_idc == 3 &&
full_poc_reset_flag)) = pocLsbDelta pocLsbVal
mesmo
pocLsbDelta = 0
DeltaPocVal = pocMsbDelta + pocLsbDelta
- O PicOrderCntVal da imagem atual é derivada da

```

seguinte forma:

```

Se (poc_reset_idc == 1)
PicOrderCntVal = slice_pic_order_cnt_lsb
Mesmo se (poc_reset_idc == 2)
PicOrderCntVal = 0
Mesmo { // poc_reset_idc == 3
PicOrderCntMsb = getCurrMsb
(slice_pic_order_cnt_lsb, poc_lsb_val, 0,
MaxPicOrderCntLsb)
PicOrderCntVal = PicOrderCntMsb +
slice_pic_order_cnt_lsb
}

```

- O valor de PrevPicOrderCnt[nuh_layer_id] é derivado como segue:

- Se a imagem atual não é uma imagem RASL, uma imagem RADL, ou uma imagem de não-referência de subcamada, e a imagem atual tem TemporalId igual a 0, PrevPicOrderCnt[nuh_layer_id] está definida igual a PicOrderCntVal.

- Caso contrário quando poc_reset_idc é igual a 3 PrevPicOrderCnt[nuh_layer_id] é igual a full_poc_reset_flag setembro? 0: poc_lsb_val.

Caso contrário, o seguinte é aplicável:

- O valor da DeltaPocVal está definido igual a 0
- O PicOrderCntVal da imagem atual é derivado da seguinte forma:

```

se (!FirstPicInLayerDecodedFlag [nuh_layer_id]) {

```

```

Se (poc_reset_idc == 1)
PicOrderCntVal = slice_pic_order_cnt_lsb
else if (poc_reset_idc == 2)
PicOrderCntVal = 0
Mesmo se (poc_reset_idc == 3) {
PicOrderCntMsb = getCurrMsb
(slice_pic_order_cnt_lsb, poc_lsb_val, 0,
MaxPicOrderCntLsb)
PicOrderCntVal = PicOrderCntMsb +
slice_pic_order_cnt_lsb
} mesmo // a imagem atual é uma imagem IRAP com
NoRaslOutputFlag igual a 1
PicOrderCntVal = slice_pic_order_cnt_lsb
} mesmo { // a derivação de POC como em HEVC
versão 1
se (a imagem atual é uma imagem IRAP com
NoRaslOutputFlag igual a 1) PicOrderCntMsb = 0
mesmo {
prevPicOrderCntLsb =
PrevPicOrderCnt [nuh_layer_id] &
(MaxPicOrderCntLsb 1)
prevPicOrderCntMsb =
PrevPicOrderCnt[nuh_layer_id]- prevPicOrderCntLsb
PicOrderCntMsb = getCurrMsb
(slice_pic_order_cnt_lsb,
prevPicOrderCntLsb,
prevPicOrderCntMsb, MaxPicOrderCntLsb)
}
PicOrderCntVal = PicOrderCntMsb +
slice_pic_order_cnt_lsb
}
- O valor de PrevPicOrderCnt [nuh_layer_id] é
derivado como segue:

```

- Se a imagem atual não é uma imagem RASL, uma imagem RADL, ou uma imagem de não-referência de subcamada, e a imagem atual tem TemporalId igual a 0, PrevPicOrderCnt [nuh_layer_id] está definida igual a PicOrderCntVal.

- Caso contrário, quando FirstPicInLayerDecodedFlag [nuh_layer_id] é igual a 0 e poc_reset_idc é igual a 3, PrevPicOrderCnt [nuh_layer_id] está definida igual a full_poc_reset_flag? 0: poc_lsb_val.

Se a imagem atual é a última imagem em uma unidade de acesso e o valor da variável DeltaPocVal é maior que 0, PicOrderCntVal de todas as imagens em todos os sub-DPBs é diminuído pelo valor da variável DeltaPocVal

O valor de PicOrderCntVal deve estar na faixa de -2^{31} a $2^{31} - 1$, inclusive. Em um CVS, os valores PicOrderCntVal para quaisquer duas imagens codificadas na mesma camada não devem ser os mesmos.

A função PicOrderCnt (picX) é especificada como segue:

PicOrderCnt (picX) = PicOrderCntVal da imagem picX (M-25)

A função DiffPicOrderCnt(picA, picB) é especificada como segue:

DiffPicOrderCnt (Pica, PICB) = PicOrderCnt (picA) - PicOrderCnt (picB) F-26

O fluxo de bits não deve conter dados que resultam em valores de DiffPicOrderCnt (picA, picB) utilizado no processo de decodificação que não estão na faixa de -2^{15} a $2^{15} - 1$, inclusive.

OBSERVAÇÃO - Seja X a imagem atual e Y e Z as duas outras imagens na mesma sequência, Y e Z são considerados na mesma direção em ordem de emissão de X, quando ambos DiffPicOrderCnt (X, Y) e DiffPicOrderCnt (X, Z) são positivos ou ambos são negativos.

[0110] Deste modo, o codificador de vídeo 20 e o decodificador de vídeo 30 representam exemplos de codificadores de vídeo configurados para codificar dados que indicam uma redefinição de contagem de ordem de imagem (POC) para um valor de POC de uma primeira imagem de uma primeira camada dos dados de vídeo de multicamada, em que a primeira imagem está incluída em uma unidade de acesso, e, com base nos dados que indicam a redefinição de POC para o valor de POC da primeira imagem e antes de decodificar a primeira imagem, diminuir valores de POC de todas as imagens armazenadas no DPB que precedem a primeira imagem, em ordem de codificação, incluindo pelo menos uma imagem de uma segunda camada de dados de vídeo de multicamada.

[0111] A figura 2 é um diagrama de blocos que ilustra um exemplo de um codificador de vídeo 20, que pode aplicar técnicas de gerenciamento de valores de POC de acordo com as técnicas desta divulgação. Em particular, o codificador de vídeo 20 pode ser configurado para gerenciar valores de POC de imagens de um fluxo de bits de multicamada de acordo com as técnicas da presente divulgação. Como observado acima, o codificador de vídeo 20 pode ser adaptado para executar codificação de vídeo multivista e/ou escalonável. Por exemplo, o codificador de vídeo 20 pode ser configurado para codificar um fluxo de bits que está de acordo com uma ou mais extensões de codificação de vídeo padrão, tal como SHVC, MV-HEVC, ou 3D-HEVC. No entanto, embora seja feita referência a padrões de codificação específicos, deve entender-se que as técnicas não são específicas para qualquer um de codificação padrão, e podem ser implementadas com padrões futuro e/ou ainda não desenvolvidos.

[0112] Codificador de vídeo 20 pode realizar intra e intercodificação de blocos de vídeo em fatias de

vídeo. Intracodificação se baseia na predição espacial para reduzir ou eliminar a redundância espacial em vídeo dentro de um determinado intervalo de vídeo ou imagem. Intercodificação se baseia em predições temporais para reduzir ou eliminar a redundância temporal de vídeo dentro de quadros ou imagens de uma sequência de vídeo adjacente. Modo Intra (modo I) pode se referir a qualquer um dos vários modos de codificação espacial. Modo Inter, tais como predição unidirecional (modo P) ou bi-predição (modo B), pode se referir a qualquer um dos vários modos de codificação temporais.

[0113] Como mostrado na figura 2, codificador de vídeo 20 recebe um bloco de vídeo atual dentro de um quadro de vídeo a ser codificado. No exemplo da figura 2, codificador de vídeo 20 inclui memória de dados de vídeo 38, unidade de seleção de modo 40, memória de imagem de referência 64, somador 50, unidade de processamento de transformada 52, unidade de quantização 54 e unidade de codificação de entropia 56. Unidade de seleção de modo 40, por sua vez, inclui unidade de compensação de movimento 44, unidade de estimação de movimento 42, unidade de intrapredição 46, e a unidade de partição 48. Para reconstrução do bloco de vídeo, um codificador de vídeo 20 também inclui a unidade de quantização inversa 58, unidade de transformada inversa 60, e somador 62. Um filtro de desbloqueio (não mostrado na figura 2) também pode ser incluído para filtrar limites de bloco para remover artefatos de bloqueio do vídeo reconstruído. Se desejado, o filtro de desbloqueio tipicamente filtra a saída do somador 62. Os filtros adicionais (em circuito ou pós circular) podem também ser usados em adição ao filtro de desbloqueio. Tais filtros não são mostrados por questões de brevidade,

mas, se desejado, podem filtrar a saída do somador 50 (como um filtro em circuito).

[0114] Durante o processo de codificação, codificador de vídeo 20 recebe um quadro de vídeo ou uma fatia de ser codificada. O quadro ou a fatia pode ser dividido em vários blocos de vídeo. A memória de dados de vídeo 38 pode armazenar os dados de vídeo a serem codificados pelos componentes do codificador de vídeo 20. Os dados de vídeo armazenados na memória de dados de vídeo 38 podem ser obtidos, por exemplo, a partir de fonte de vídeo 18. Memória de imagem de referência 64 pode ser referida como um DPB que armazena dados de referência de vídeo para utilização na codificação de dados de vídeo por um codificador de vídeo 20, por exemplo, em modos de codificação intra ou inter. Memória de dados de vídeo 38 e memória de imagem de referência 64 podem ser formadas por qualquer um de uma variedade de dispositivos de memória, tais como a memória dinâmica de acesso aleatório (DRAM), incluindo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM), ou outros tipos de dispositivos de memória. Memória de dados de vídeo 38 e memória de imagem de referência 64 podem ser providas pelo mesmo dispositivo de memória ou dispositivos de memória separados. Em vários exemplos, memória de dados de vídeo 38 podem ser em chip com outros componentes de um codificador de vídeo 20, ou fora do chip em relação a esses componentes.

[0115] A unidade de estimação de movimento 42 e a unidade de compensação de movimento 44 executam codificação interpreditiva do bloco de vídeo recebido relativo a um ou mais blocos em um ou mais quadros de referência para prover predição temporal. A unidade de intrapredição 46 pode, alternativamente, executar codificação intrapreditiva do

bloco de vídeo recebido em relação a um ou mais blocos vizinhos no mesmo quadro ou fatia que o bloco a ser codificado para prover uma predição espacial. O codificador de vídeo 20 pode executar várias passagens de codificação, por exemplo, para selecionar um modo de codificação apropriada para cada bloco de dados de vídeo.

[0116] Além disso, a unidade de partição 48 pode particionar blocos de dados de vídeo em sub-blocos, com base na avaliação dos esquemas de particionamento anteriores em passagens de codificação anteriores. Por exemplo, a unidade de partição 48 pode inicialmente particionar um quadro ou fatia em LCUs, e particionar cada uma das LCUs em sub-CUs com base na análise da taxa de distorção (por exemplo, otimização de taxa de distorção). A unidade de seleção de modo 40 pode produzir ainda uma estrutura de dados quadtree indicativa de particionamento de uma LCU em sub-CUs. CUs de nó folha da quadtree pode incluir uma ou mais PUs e uma ou mais TUs.

[0117] A unidade de seleção de modo 40 pode selecionar um dos modos de codificação, intra ou inter, por exemplo, com base em resultados de erro, e provê o bloco intra ou intercodificado resultante para o somador 50 para gerar dados de bloco residuais e para o somador 62 para reconstruir o bloco codificado para o uso como um quadro de referência. Unidade de seleção de modo 40 também provê elementos de sintaxe, como vetores de movimento, indicadores de intramodo, informações de partição, e outras informações de sintaxe, para unidade de codificação de entropia 56.

[0118] A unidade de estimação de movimento 42 e unidade de compensação de movimento 44 podem ser altamente integradas, mas são ilustradas separadamente para fins conceituais. Estimação de movimento, realizada pela unidade

de estimação de movimento 42, é o processo de geração de vetores de movimento, que estimam o movimento para blocos de vídeo. Um vetor de movimento, por exemplo, pode indicar o deslocamento de uma PU de um bloco de vídeo dentro de um quadro de vídeo atual ou da imagem em relação a um bloco preditivo dentro de uma imagem de referência (ou outra unidade codificada) em relação ao bloco atual a ser codificado no âmbito da fatia atual (ou outra unidade de codificação). Um bloco de predição é um bloco que se encontra em proximidade ao bloco a ser codificado, em termos de diferença de pixels, o que pode ser determinado pela soma da diferença absoluta (SAD), soma de diferença quadrada (SSD), ou outras métricas de diferença.

[0119] A unidade de estimação de movimento 42 calcula um vetor de movimento para uma PU de um bloco de vídeo em uma fatia intercodificada pela comparação da posição da PU com a posição de um bloco preditivo de uma imagem de referência. A imagem de referência pode ser selecionada a partir de uma primeira lista de imagem de referência (Lista 0) ou uma segunda lista de imagem de referência (Lista 1), cada uma das quais identifica uma ou mais imagens de referência armazenadas na memória de imagem de referência 64. Unidade de estimação de movimento 42 envia o vetor de movimento calculado à unidade de codificação de entropia 56 e unidade de compensação de movimento 44.

[0120] A compensação de movimento, realizada pela unidade de compensação de movimento 44, pode envolver buscar ou gerar o bloco preditivo com base no vetor de movimento determinado pela unidade de estimação de movimento 42. Mais uma vez, a unidade de estimação de movimento 42 e a unidade de compensação de movimento 44 podem ser funcionalmente integradas, em alguns exemplos. Ao

receber o vetor de movimento para a PU do bloco de vídeo atual, a unidade de compensação de movimento 44 pode localizar o bloco de predição para o qual o vetor de movimento aponta em uma das listas de imagem de referência. Somador 50 forma um bloco de vídeo residual, subtraindo os valores de pixel do bloco de predição dos valores de pixel do bloco de vídeo atual a ser codificado, formando os valores de diferença de pixel, como discutido abaixo. Em geral, a unidade de estimação de movimento 42 executa estimação de movimento em relação aos componentes de luminância, e a unidade de compensação de movimento 44 utiliza os vetores de movimento calculados com base nos componentes de luminância para ambos os componentes de crominância e de componentes de luminância. A unidade de seleção de modo 40 pode também gerar elementos de sintaxe associados com os blocos de vídeo e a fatia de vídeo para utilização pelo decodificador de vídeo 30 na decodificação dos blocos de vídeo da fatia de vídeo.

[0121] Unidade de estimação de movimento 42 pesquisa as imagens de referência armazenadas em um DPB da memória de imagem de referência 64. Unidade de seleção de modo 40 pode determinar quais imagens de referência devem ser utilizadas para codificar uma imagem, fatia de imagem, e/ou uma sequência de imagens. O codificador de vídeo 20 pode codificar valores de contagem de ordem de imagem (POC) para imagens de referência utilizadas para codificar a imagem, fatia, ou sequência. O codificador de vídeo 20 pode codificar os valores de POC indicativos das imagens de referência para serem incluídos na lista de imagem de referência em cabeçalhos de fatias e/ou conjuntos de parâmetros, tais como PPSS e/ou SPSS.

[0122] Deste modo, um decodificador de vídeo, como o decodificador de vídeo 30, pode reproduzir a lista

de imagem de referência, incluindo as imagens de referência indicadas no cabeçalho da fatia e/ou conjunto (s) de parâmetro. Além disso, após a codificação de um bloco usando um vetor de movimento produzido pela unidade de estimação de movimento 42, codificador de vídeo 20 pode codificar a informação de movimento para o bloco, onde a informação de movimento pode incluir dados representantes do vetor de movimento, um identificador para a lista de imagem de referência, e um índice de referência que identifica uma imagem de referência na lista de imagem de referência.

[0123] Em alguns exemplos, codificador de vídeo 20 pode calcular os valores para as posições de pixel subinteiros de imagens de referência armazenadas na memória de imagem de referência 64. Por exemplo, codificador de vídeo 20 pode interpolar valores das posições de um quarto de pixel, posições de um oitavo de pixel, ou outras posições de pixel fracionado da imagem de referência. Por conseguinte, a unidade de estimação de movimento 42 pode executar uma busca de movimento relativo para as posições de pixel completo e posições de pixel fracionado e emitir um vetor de movimento com uma precisão de pixel fracionado.

[0124] Codificador de vídeo 20 também pode formar períodos de redefinição de POC e valores de codificação para identificadores de período de redefinição de POC para fatias de imagens para atribuir as fatias aos respectivos períodos de redefinição de POC. O codificador de vídeo 20 pode atribuir fatias para períodos de redefinição de POC individualmente dentro de cada camada de dados de vídeo. Além disso, codificador de vídeo 20 pode determinar tipos de redefinição de POC para as fatias e valores de codificação para as indicações (por exemplo, valores de dois bits) dos tipos de redefinição de POC para as fatias.

[0125] Tal como discutido em relação à figura 1, codificador de vídeo 20 pode ser configurado de acordo com as restrições estabelecidas por um padrão de codificação de vídeo aplicável. Por exemplo, codificador de vídeo 20 pode garantir que a primeira imagem de cada período de redefinição de POC tenha um tipo de redefinição de POC indicando que os valores de POC devem ser total ou parcialmente redefinidos. Além disso, o codificador de vídeo 20 pode codificar informação adicional indicativa de um valor de POC LSB em alguns exemplos, o que pode indicar um valor de POC LSB para uma imagem de POC âncora do período de redefinição de POC.

[0126] Codificador de vídeo 20 pode também diminuir valores de POC de imagens de referência no DPB de memória de imagem de referência 64. De acordo com aspectos desta divulgação, codificador de vídeo 20 pode ser configurado para diminuir valores de POC de todas as imagens (de todas as camadas) armazenadas na memória de imagem de referência 64 antes da codificação de uma imagem inicial de uma unidade de acesso em que redefinição de POC é executada. Em alguns exemplos, o codificador de vídeo 20 pode aplicar um processo de diminuição de POC nas imagens anteriores em todos os sub-DPBs (onde cada camada de dados de vídeo está associada com um respectivo sub-DPB) da memória de imagem de referência 64 apenas uma vez ao codificar a unidade de acesso para a qual redefinição de POC está sendo realizada.

[0127] Ao diminuir as imagens de referência armazenadas na memória de imagem de referência 64, codificador de vídeo 20 pode usar os valores de POC decrementados ao codificar os valores de POC nos cabeçalhos de fatia e/ou conjuntos de parâmetro para indicar imagens de referência a serem incluídas em uma lista de imagens de

referência. O valor pelo qual codificador de vídeo 20 diminui os valores de POC de imagens no DPB pode ser o mesmo valor pelo qual a imagem POC âncora é diminuída. A redefinição de POC da imagem âncora POC pode ser limitada apenas a redefinição de POC MSB ou redefinição do valor de POC completo da imagem POC âncora.

[0128] Unidade de intrapredição 46 pode intrapredizer um bloco atual, como uma alternativa para a interpredição realizada pela unidade de estimação de movimento 42 e a unidade de compensação de movimento 44, conforme descrito acima. Em particular, a unidade de intrapredição 46 pode determinar um modo de intrapredição a ser usado para codificar um bloco atual. Em alguns exemplos, a unidade de intrapredição 46 pode codificar um bloco atual utilizando diversos modos de intrapredição, por exemplo, durante a passagem de codificação separada, e unidade de intrapredição 46 (ou unidade de seleção de modo 40, em alguns exemplos) pode selecionar um modo de intrapredição adequado para ser usado a partir dos meios testados.

[0129] Por exemplo, a unidade de intrapredição 46 pode calcular os valores de taxa de distorção usando uma análise da taxa de distorção para os vários modos de intrapredição testados, e selecionar o modo de intrapredição tendo as melhores características de taxa de distorção entre os modos testados. Análise da taxa de distorção determina geralmente uma quantidade de distorção (ou de erro) entre um bloco codificado e um bloco original, não codificado que tenha sido codificado para produzir o bloco codificado, bem como uma taxa de bits (isto é, um número de bits) usada para produzir o bloco codificado. A unidade de intrapredição 46 pode calcular as razões das distorções e taxas para os vários blocos codificados para

determinar qual o modo de intrapredição apresenta o melhor valor da taxa de distorção para o bloco.

[0130] Depois de selecionar um modo de intrapredição para um bloco, unidade de intrapredição 46 pode prover informação indicativa do modo de intrapredição selecionado para o bloco para a unidade de codificação de entropia 56. A unidade de codificação de entropia 56 pode codificar a informação indicando o modo de intrapredição selecionado. O codificador de vídeo 20 pode incluir nos dados de configuração de fluxo de bits transmitidos, que pode incluir uma pluralidade de tabelas de índice de modo intrapredição e uma pluralidade de tabelas de índice de modo de intrapredição modificadas (também referidas tabelas de mapeamento de palavras de código), definições de contextos de codificação para vários blocos, e indicações de um modo de intrapredição mais provável, uma tabela de índice de modo de intrapredição, e uma tabela de índice de modo de intrapredição modificada para usar para cada um dos contextos.

[0131] O codificador de vídeo 20 forma um bloco de vídeo residual, subtraindo os dados de predição da unidade de seleção de modo 40 do bloco de vídeo original que está sendo codificado. Somador 50 representa o componente ou componentes que executam esta operação de subtração. A unidade de processamento de transformada 52 aplica uma transformada, tal como uma transformada de cosseno discreta (DCT) ou uma transformada conceitualmente semelhante, para o bloqueio residual, produzindo um bloco de vídeo compreendendo valores de coeficientes de transformada residuais. Unidade de processamento de transformada 52 pode realizar outras transformadas que são conceitualmente semelhantes a DCT. Transformada de Wavelet, transformada de inteiro, transformadas de sub-bandas ou

outros tipos de transformadas também poderiam ser utilizados. Em qualquer caso, unidade de processamento de transformada 52 aplica transformada ao bloco residual, produzindo um bloco de coeficientes de transformada residuais. A transformada pode converter a informação residual de um domínio de valores de pixel para um domínio de transformada, tal como um domínio de frequência.

[0132] Unidade de processamento de transformada 52 pode enviar os coeficientes de transformada resultantes para a unidade de quantização 54. A unidade de quantização 54 quantifica os coeficientes de transformada para reduzir ainda mais a taxa de bits. O processo de quantização pode reduzir a profundidade do bit associado com alguns ou todos os coeficientes. O grau de quantização pode ser modificado por ajuste de um parâmetro de quantização. Em alguns exemplos, a unidade de quantização 54 pode, em seguida, executar um varrimento da matriz incluindo os coeficientes de transformada quantificados. Alternativamente, unidade de codificação de entropia 56 pode realizar a varredura.

[0133] Após quantização, unidade de codificação de entropia 56 codifica por entropia coeficientes de transformada quantificados. Por exemplo, unidade de codificação de entropia 56 pode realizar codificação de comprimento de variável adaptativa de contexto (CAVLC), codificação aritmética binária adaptativa de contexto (CABAC), codificação aritmética binária adaptativa de contexto baseada em sintaxe (SBAC), codificação de entropia de particionamento de intervalo de probabilidade (PIPE) ou outra técnica de codificação por entropia. No caso de codificação de entropia baseada em contexto, o contexto pode ser baseado em blocos vizinhos. Após a codificação de entropia pela unidade de codificação de entropia 56, o fluxo de bits codificado pode ser transmitido para outro

dispositivo (por exemplo, decodificador de vídeo 30) ou arquivado para posterior transmissão ou recuperação.

[0134] A unidade de quantização inversa 58 e unidade de transformada inversa 60 aplicam quantização inversa e a transformada inversa, respectivamente, para reconstruir o bloco residual no domínio do pixel, por exemplo, para utilização posterior como um bloco de referência. A unidade de compensação de movimento 44 pode calcular um bloco de referência, adicionando o bloco residual a um bloco preditivo de um dos quadros de memória de imagem de referência 64. Unidade de compensação de movimento 44 pode também aplicar um ou mais filtros de interpolação ao bloco residual reconstruído para calcular valores de pixel subinteiros para uso na estimação de movimento. Somador 62 acrescenta o bloco residual reconstruído ao bloco de predição compensado por movimento produzido pela unidade de compensação de movimento 44 para produzir um bloco de vídeo reconstruído para armazenamento na memória de imagem de referência 64. O bloco de vídeo reconstruído pode ser utilizado pela unidade de estimação de movimento 42 e unidade de compensação de movimento 44 como um bloco de referência para intercodificar um bloco em um quadro de vídeo subsequente.

[0135] Deste modo, o codificador de vídeo 20 da figura 2 representa um exemplo de um codificador de vídeo configurado para codificar os dados que indicam uma redefinição de contagem de ordem de imagem (POC) para um valor de POC de uma primeira imagem de uma primeira camada de dados de vídeo de multicamada, em que a primeira imagem é incluída em uma unidade de acesso. O codificador de vídeo 20 é também configurado para, com base nos dados que indicam a redefinição de POC para o valor de POC da primeira imagem e antes da decodificação da primeira

imagem, diminuir valores de POC de todas as imagens armazenadas em um armazenador de imagem decodificada (DPB) que precedem a primeira imagem, em ordem de codificação, incluindo pelo menos uma imagem de uma segunda camada de dados de vídeo de multicamada.

[0136] A figura 3 é um diagrama de blocos que ilustra um exemplo do decodificador de vídeo 30, que pode aplicar técnicas de gerenciamento de valores de POC de acordo com as técnicas desta divulgação. Como notado acima, um decodificador de vídeo 30 pode ser adaptado para executar codificação de vídeo multivista e/ou escalonável. Por exemplo, decodificador de vídeo 30 pode ser configurado para decodificar um fluxo de bits que está de acordo com uma ou mais extensões de codificação de vídeo padrão, tal como SHVC, MV-HEVC, ou 3D-HEVC. No entanto, embora seja feita referência a padrões de codificação específicos, deve entender-se que as técnicas não são específicas para qualquer um de codificação padrão, e podem ser implementadas com padrões futuros e/ou ainda não desenvolvidos.

[0137] No exemplo da figura 3, decodificador de vídeo 30 inclui memória de dados de vídeo 68, uma unidade de decodificação de entropia 70, unidade de compensação de movimento 72, unidade de intrapredição 74, unidade de quantização inversa 76, unidade de transformada inversa 78, memória de imagem de referência 82 e somador 80. O decodificador de vídeo 30 pode, em alguns exemplos, executar uma passagem de decodificação geralmente recíproca à passagem de codificação descrita em relação ao codificador de vídeo 20 (figura 2). Unidade de compensação de movimento 72 pode gerar dados de predição baseados em vetores de movimento recebidos da unidade de decodificação de entropia 70, enquanto a unidade de intrapredição 74 pode

gerar dados de predição com base em indicadores de modo de intrapredição recebidos da unidade de decodificação de entropia 70.

[0138] A memória de dados de vídeo 68 pode armazenar os dados de vídeo, tal como um fluxo de bits de vídeo codificado, para ser decodificados pelos componentes do decodificador de vídeo 30. Os dados de vídeo armazenados na memória de dados de vídeo 68 podem ser obtidos, por exemplo, a partir do meio legível por computador, por exemplo, a partir de uma fonte de vídeo local, como uma câmera, através de comunicação em rede com ou sem fio de dados de vídeo, ou acessando meios de armazenamento de dados físico. A memória de dados de vídeo 68 pode formar um armazenador de imagem codificada (CPB) que armazena os dados codificados de vídeo a partir de um fluxo de bits de vídeo codificado. Memória de imagem de referência 82 pode ser referida como um DPB que armazena dados de referência de vídeo para utilização na decodificação de dados de vídeo pelo decodificador de vídeo 30, por exemplo, nos modos de intra ou intercodificação. A memória de dados de vídeo 68 e a memória de imagem de referência 82 podem ser formadas por qualquer um de uma variedade de dispositivos de memória, tais como a memória dinâmica de acesso aleatório (DRAM), incluindo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM), ou outros tipos de dispositivos de memória. A memória de dados de vídeo 68 e memória de imagem de referência 82 podem ser providas pelo mesmo dispositivo de memória ou dispositivos de memória separados. Em vários exemplos, memória de dados de vídeo 68 pode ser em chip com outros componentes do decodificador de vídeo 30, ou fora do chip em relação a esses componentes.

[0139] Durante o processo de decodificação, decodificador de vídeo 30 recebe um fluxo de bits de vídeo

codificado que representa blocos de vídeo de uma fatia de vídeo codificado e elementos de sintaxe associados provenientes de codificador de vídeo 20. A unidade de decodificação de entropia 70 do decodificador de vídeo 30 decodifica por entropia o fluxo de bits para gerar coeficientes quantificados, vetores de movimento ou indicadores de modo de intrapredição e outros elementos de sintaxe. Unidade de decodificação de entropia 70 encaminha os vetores de movimento e outros elementos de sintaxe para a unidade de compensação de movimento 72. O decodificador de vídeo 30 pode receber os elementos de sintaxe no nível da fatia de vídeo e/ou no nível de bloco de vídeo.

[0140] Quando a fatia de vídeo é codificada como uma fatia intracodificada (I), a unidade de intrapredição 74 pode gerar dados de predição para um bloco de vídeo da fatia de vídeo atual com base em um modo de intrapredição sinalizado e dados de blocos anteriormente decodificados do quadro ou imagem atual. Quando o quadro de vídeo é codificado como uma fatia intercodificada (isto é, B, P ou GPB), a unidade de compensação de movimento 72 produz blocos preditivos para um bloco de vídeo da fatia de vídeo atual com base nos vetores de movimento e outros elementos de sintaxe recebidos da unidade de decodificação de entropia 70. Os blocos preditivos podem ser produzidos a partir de uma das imagens de referência dentro de uma das listas de imagem de referência. O decodificador de vídeo 30 pode construir as listas de quadro de referência, lista 0 e lista 1, utilizando técnicas de construção padrão com base em imagens de referência armazenadas em um DPB de memória de imagem de referência 82.

[0141] Mais particularmente, decodificador de vídeo 30 pode decodificar um PPS indicando se fatias de uma imagem correspondente ao PPS incluem informações de

redefinição de contagem de ordem de imagem (POC). Assumindo que as fatias incluem informações de redefinição de POC, decodificador de vídeo 30 pode decodificar um cabeçalho do segmento de fatia de uma fatia da imagem incluindo as informações de redefinição de POC. As informações de redefinição de POC podem incluir um identificador de período de redefinição de POC e um tipo de redefinição de POC.

[0142] O identificador de período de redefinição de POC pode indicar um período de redefinição de POC ao qual a fatia corresponde. O tipo de redefinição de POC pode indicar se o valor de POC de uma imagem incluindo a fatia está completamente redefinido ou se apenas o valor MSB da POC é redefinido. Além disso, a o tipo de redefinição de POC indica que pode nenhuma redefinição de POC deve ser realizada ou que informações adicionais são sinalizadas. Se o tipo de redefinição de POC indica que informações adicionais são sinalizadas, decodificador de vídeo 30 pode decodificar as informações adicionais, o que pode indicar um valor de POC LSB e se o valor de POC é totalmente redefinido ou somente MSB é redefinido.

[0143] De acordo com aspectos da presente divulgação, depois de decodificar a informação que indica uma redefinição de POC para uma fatia de dados de vídeo de uma imagem inicial de uma unidade de acesso, mas antes de decodificar os dados de vídeo da unidade de acesso, um decodificador de vídeo 30 pode diminuir valores de POC de imagens armazenadas na memória de imagem de referência 82. De acordo com aspectos desta divulgação, em alguns exemplos, decodificador de vídeo 30 pode ser configurado para diminuir valores de POC de todas as imagens (de todas as camadas) armazenadas na memória de imagem de referência 82 antes da decodificação de dados de vídeo da imagem

inicial da unidade de acesso para a qual redefinição de POC está sendo executada. O decodificador de vídeo 30 pode aplicar um processo de diminuição de POC em imagens anteriores em todos os sub-DPBs (onde cada camada de dados de vídeo está associada com um respectivo sub-DPB) da memória de imagem de referência 82 apenas uma vez ao decodificar a unidade de acesso para a qual redefinição de POC está sendo realizada.

[0144] Depois de realizar a redefinição de POC, decodificador de vídeo 30 pode decodificar valores de POC de imagens de referência para ser incluídas em uma lista de imagens de referência. O decodificador de vídeo 30 pode decodificar estes valores de POC nos cabeçalhos de segmento de fatia e/ou a partir de conjuntos de parâmetros, tais como um PPS ou um SPS. Decodificador de vídeo 30 pode então construir uma lista de imagens de referência, incluindo as imagens de referência identificadas pelos valores de POC decodificados.

[0145] Depois de construir a lista de imagem de referência para uma fatia, decodificador de vídeo 30 pode decodificar blocos da fatia. Unidade de compensação de movimento 72 determina informações de predição para um bloco de vídeo da fatia de vídeo atual ao analisar os vetores de movimento e outros elementos de sintaxe, e usa as informações de predição para produzir os blocos preditivos para o bloco de vídeo atual sendo decodificado. Por exemplo, a unidade de compensação de movimento 72 utiliza algum dos elementos de sintaxe recebidos para determinar um modo de predição (por exemplo, intra ou interpredição) utilizado para codificar os blocos de vídeo da fatia de vídeo, um tipo de fatia de interpredição (por exemplo, fatia B, fatia P, ou fatia GPB), vetores de movimento para cada bloco de vídeo intercodificado da

fatia, status de interpredição para cada bloco de vídeo intercodificado da fatia, e outra informação para decodificar os blocos de vídeo na fatia de vídeo atual. A informação de movimento de um bloco interpredito pode incluir uma lista de identificador de imagem de referência e um índice de referência para identificar uma imagem de referência na lista de imagem de referência para usar para prever o bloco.

[0146] A Unidade de compensação de movimento 72 pode também executar a interpolação com base em filtros de interpolação. A unidade de compensação de movimento 72 pode usar filtros de interpolação como usados pelo codificador de vídeo 20 durante a codificação dos blocos de vídeo para calcular os valores interpolados para pixels subinteiros de blocos de referência. Neste caso, a unidade de compensação de movimento 72 pode determinar os filtros de interpolação utilizados por um codificador de vídeo 20, a partir dos elementos de sintaxe recebidos e utilizar os filtros de interpolação para produzir blocos preditivos.

[0147] Unidade de quantização inversa 76 quantifica inversamente, ou seja, desquantifica coeficientes de transformada quantificados providos no fluxo de bits e decodificados pela unidade de decodificação de entropia 70. O processo de quantização inversa pode incluir a utilização de um parâmetro de quantização QPy calculado pelo decodificador de vídeo 30 para cada um dos blocos de vídeo na fatia de vídeo para determinar um grau de quantização e, do mesmo modo, um grau de quantização inversa, que deve ser aplicado.

[0148] A unidade de transformada inversa 78 aplica uma transformada inversa, por exemplo, uma transformada discreta de cosseno inversa, uma transformada de inteiro inversa, ou um processo de transformada inverso

conceptualmente semelhante, aos coeficientes de transformada, a fim de produzir blocos residuais no domínio do pixel.

[0149] Após a unidade de compensação de movimento 72 gerar o bloco preditivo para o bloco de vídeo atual com base nos vetores de movimento e outros elementos de sintaxe, decodificador de vídeo 30 forma um bloco de vídeo decodificado pela soma dos blocos residuais da unidade de transformada inversa 78 com os blocos preditivos correspondentes gerados pela unidade de compensação de movimento 72. Somador 80 representa o componente ou componentes que executam esta operação somatória. Se desejado, um filtro de desbloqueio pode também ser aplicado para filtrar os blocos decodificados, a fim de remover artefatos de bloqueio. Outros filtros de loop (quer no loop de codificação ou após o ciclo de codificação) também podem ser usados para suavizar as transições de pixel, ou de outra forma melhorar a qualidade do vídeo. Os blocos decodificados de vídeo de um dado quadro ou imagem são então armazenados na memória de imagem de referência 82, que armazena imagens de referência utilizadas para a compensação de movimento subsequente. Memória de imagem de referência 82 também armazena vídeo decodificado para a apresentação posterior em um dispositivo de exibição, como o dispositivo de exibição 32 da figura 1.

[0150] Deste modo, decodificador de vídeo 30 da figura 3 representa um exemplo de um decodificador de vídeo configurado para decodificar os dados que indicam uma redefinição de contagem de ordem de imagem (POC) para um valor de POC de uma primeira imagem de uma primeira camada de dados de vídeo de multicamada, em que a primeira imagem é incluída em uma unidade de acesso. O decodificador de vídeo 30 é também configurado para, com base nos dados que

indicam a redefinição de POC para o valor de POC da primeira imagem e antes da decodificação da primeira imagem, diminuir os valores de POC de todas as imagens armazenadas em um armazenador de imagem decodificada (DPB) que precede a primeira imagem, em ordem de codificação, incluindo pelo menos uma imagem de uma segunda camada de dados de vídeo de multicamada.

[0151] A figura 4 é um fluxograma que ilustra um exemplo de método para codificar dados que indicam um período de redefinição do POC para uma fatia de acordo com as técnicas desta divulgação. Além disso, o método da figura 4 inclui codificação de um bloco atual. O bloco atual pode compreender uma CU atual ou uma porção da CU atual. Embora descrito em relação ao codificador de vídeo 20 (figuras 1 e 2), deve ser compreendido que outros dispositivos podem ser configurados para realizar um método semelhante ao da figura 4.

[0152] Neste exemplo, codificador de vídeo 20 pode atribuir a fatia atual a um período de redefinição de POC (148). Por exemplo, se a fatia atual faz parte de uma imagem âncora, codificador de vídeo 20 pode sinalizar que a fatia atual constitui o início de um novo período de redefinição de POC com um identificador de período de redefinição de POC que é diferente dos identificadores de período de redefinição de POC anteriores. Alternativamente, se a fatia atual não forma parte de um quadro âncora, codificador de vídeo 20 pode sinalizar que a fatia atual faz parte de um período de redefinição de POC existente.

[0153] O codificador de vídeo 20, em seguida, diminui os valores de POC de imagens de referência de todas as imagens que estão atualmente em um DPB (150). Isto é, de acordo com aspectos desta divulgação, codificador de vídeo 20 pode diminuir os valores POC das imagens em todos os

sub-DPBs do DPB. O codificador de vídeo 20 pode diminuir os valores de POC depois de atribuir a fatia atual para um período de redefinição de POC (por exemplo, que codifica o cabeçalho da fatia), mas antes da codificação de dados de vídeo de primeira fatia. O codificador de vídeo 20 pode diminuir os valores POC das imagens no DPB pelo mesmo valor diminuído.

[0154] O codificador de vídeo 20, em seguida, forma uma lista de imagens de referência, incluindo pelo menos algumas das imagens de referência (152). O codificador de vídeo 20 adicionalmente sinaliza valores de POC das imagens de referência incluídas na lista de imagem de referência (154). Por exemplo, o codificador de vídeo 20 pode codificar os valores de POC (ou porções dos valores POC, por exemplo, valores de POC LSB) para as imagens de referência em um conjunto de parâmetros, tais como um SPS ou PPS, e/ou em um cabeçalho de fatia da fatia. Algumas imagens de referência (por exemplo, imagens de referência de longo prazo) podem ser sinalizadas no conjunto de parâmetros, enquanto outras imagens de referência (por exemplo, imagens de referência de curto prazo) podem ser sinalizadas no cabeçalho de fatia.

[0155] Deve ser entendido que as etapas de formar uma lista de imagem de referência e sinalizar quais imagens são incluídas na lista de imagem de referência podem ser realizadas várias vezes por várias passagens de codificação diferentes, a fim de determinar o conjunto de imagens de referência que proporciona as melhores características de taxa de distorção para, por exemplo, blocos da fatia atual. Ou seja, codificador de vídeo 20 pode selecionar o conjunto de imagens de referência incluído na lista de imagem de referência com base em características de todos os blocos

na fatia atual, e não apenas com base nas características individuais de um único bloco.

[0156] Codificador de vídeo 20 pode, em seguida, prever o bloco atual da fatia atual (156). Por exemplo, codificador de vídeo 20 pode calcular uma ou mais unidades de predição (PUs) para o bloco atual. Mais particularmente, a unidade de estimação de movimento 42 pode realizar uma busca de movimento para o bloco atual entre as imagens de referência da lista de imagem de referência para identificar um bloco de adaptação usado como um bloco de referência, por exemplo, com base no SAD, SSD, MAD, MSD, ou outras métricas de cálculo de erro. A unidade de estimativa de movimento 42 pode produzir um vetor de movimento para o bloco atual com base na busca de movimento do codificador de vídeo 20 pode, em seguida, codificar a informação de movimento para o bloco (158), que pode incluir valores de diferença de vetores de movimento para o vetor de movimento, um identificador de lista de imagens de referência, e um índice de referência, que em conjunto identificam o bloco de referência.

[0157] O codificador de vídeo 20 pode então calcular um bloco residual para o bloco atual, por exemplo, para produzir uma unidade de transformada (TU) (160). Para calcular o bloco residual, codificador de vídeo 20 pode calcular a diferença entre o bloco original, não codificado e o bloco predito para o bloco atual. O codificador de vídeo 20 pode, em seguida, transformar e quantizar coeficientes do bloco residual (162). Em seguida, o codificador de vídeo 20 pode varrer coeficientes de transformada quantificados do bloco residual (164). Durante a varredura, ou após a digitalização, o codificador de vídeo 20 pode codificar por entropia os coeficientes (166). Por exemplo, o codificador de vídeo 20 pode codificar os

coeficientes usando CAVLC ou CABAC. O codificador de vídeo 20 pode, em seguida, emitir dados codificados por entropia do bloco (168).

[0158] Deste modo, o método da figura 4 representa um exemplo de um método que inclui a codificação, por um codificador de vídeo, de dados que indicam uma redefinição de contagem de ordem de imagem (POC) para um valor de POC de uma primeira imagem de uma primeira camada de dados de vídeo de multicamada, em que a primeira imagem é incluída em uma unidade de acesso. O método de exemplo da figura 4 também inclui, com base nos dados que indicam a redefinição de POC para o valor de POC da primeira imagem e antes de decodificar a primeira imagem, diminuir os valores de POC de todas as imagens armazenadas em um armazenador de imagem decodificada (DPB) que precedem a primeira imagem em ordem de codificação, incluindo pelo menos uma imagem de uma segunda camada de dados de vídeo de multicamada.

[0159] A figura 5 é um fluxograma que ilustra um exemplo de método para a decodificação de dados que indicam um período de redefinição do POC para uma fatia de acordo com as técnicas desta divulgação. Além disso, o método da figura 4 inclui codificação de um bloco atual. O bloco atual pode compreender uma CU atual ou uma porção da CU atual. Embora descrito em relação ao decodificador de vídeo 30 (figuras 1 e 3), deve ser compreendido que outros dispositivos podem ser configurados para realizar um método semelhante ao da figura 5.

[0160] Inicialmente, decodificador de vídeo 30 decodifica os dados que indicam uma redefinição de POC (200). Em alguns exemplos, tal como aqui descrito, os dados que indicam a redefinição de POC podem ser um identificador de redefinição de POC que está incluído em um segmento de

cabeçalho de fatia da fatia atual. Nesses exemplos, o decodificador de vídeo 30 pode decodificar ainda um indicador de tipo de redefinição de POC. O método da figura 5 baseia-se na suposição de que o período de identificador de redefinição de POC indica que a fatia atual faz parte de um novo período de redefinição de POC. O decodificador de vídeo 30 pode ainda determinar que a imagem atual é a primeira imagem em uma unidade especial de acesso (por exemplo, a imagem com o menor identificador de camada entre todas as imagens de uma unidade de acesso particular).

[0161] O decodificador de vídeo 30 pode também diminuir valores de POC de todas as imagens armazenadas em um DPB (202). Por exemplo, de acordo com aspectos desta divulgação, decodificador de vídeo 30 pode diminuir os valores de POC das imagens em todos os sub-DPBs do DPB. Em alguns exemplos, decodificador de vídeo 30 pode diminuir os valores de POC depois da decodificação do cabeçalho da fatia para a fatia atual (por exemplo, como realizado na etapa 200), mas antes da decodificação de dados de vídeo da primeira fatia tal como descrito abaixo. O decodificador de vídeo 30 pode diminuir os valores de POC das imagens no DPB pelo mesmo valor diminuído.

[0162] O decodificador de vídeo 30 decodifica então valores de POC de imagens de referência a serem incluídas em uma lista de imagem de referência (204), por exemplo, a partir do segmento de cabeçalho da fatia e/ou um conjunto de parâmetros, tais como PPS ou SPS. Decodificador de vídeo 30, em seguida, forma a lista de imagem de referência (206).

[0163] Em seguida, decodificador de vídeo 30 decodifica informações de movimento de um bloco atual (208). A informação de movimento pode incluir, por exemplo, uma lista de identificador de imagem de referência e um

índice de referência para a lista de imagem de referência. O decodificador de vídeo 30, em seguida, prediz o bloco atual (200), por exemplo, usando o modo de interpredição para calcular um bloco previsto para o bloco atual. Mais particularmente, decodificador de vídeo 30 usa o identificador de lista de imagem de referência para identificar qual lista de imagem de referência usar, e o índice de referência para identificar um índice de referência na lista de imagem de referência. O decodificador de vídeo 30 decodifica então um vetor de movimento do bloco atual e identifica um bloco de referência na imagem de referência identificada.

[0164] O decodificador de vídeo 30 também pode receber dados codificados por entropia para o bloco atual, tal como os dados codificados por entropia para coeficientes de um bloco residual correspondente ao bloco atual (212). O decodificador de vídeo 30 pode decodificar por entropia os dados codificados por entropia para reproduzir coeficientes do bloco residual (214). O decodificador de vídeo 30 pode então varrer inversamente os coeficientes reproduzidos (216), para criar um bloco de coeficientes de transformada quantificados. Isto é, utilizando a varredura inversa, decodificador de vídeo 30 converte um vetor unidimensional em uma matriz bidimensional. O decodificador de vídeo 30 pode então quantificar inversamente e transformar inversamente os coeficientes para produzir um bloco residual (218). O decodificador de vídeo 30 pode, em última análise decodificar o bloco atual, combinando o bloco previsto e o bloco residual (220).

[0165] Deste modo, o método da figura 5 representa um exemplo de um método incluindo dados de decodificação que indicam uma redefinição de contagem de

ordem de imagem (POC) para um valor de POC de uma primeira imagem de uma primeira camada de dados de vídeo de multicamada, em que a primeira imagem é incluída em uma unidade de acesso. O método de exemplo da figura 5 também inclui, com base nos dados que indicam a redefinição de POC para o valor de POC da primeira imagem e antes de decodificar a primeira imagem, diminuir os valores de POC de todas as imagens armazenadas em um armazenador de imagem decodificada (DPB) que precedem a primeira imagem em ordem de codificação, incluindo pelo menos uma imagem de uma segunda camada de dados de vídeo de multicamada.

[0166] Deve ser reconhecido que, dependendo do exemplo, certos atos ou eventos de qualquer uma das técnicas aqui descritas podem ser realizados em uma sequência diferente, podem ser adicionados, fundidos, ou deixados de fora (por exemplo, nem todos atos ou eventos descritos são necessários para a prática das técnicas). Além disso, em certos exemplos, atos ou eventos podem ser realizados simultaneamente, por exemplo, através do processamento multisequenciado, processamento de interrupção, ou múltiplos processadores, em vez de sequencialmente.

[0167] Em um ou mais exemplos, as funções descritas podem ser implementadas em hardware, software, firmware, ou qualquer combinação dos mesmos. Se implementadas em software, as funções podem ser armazenadas ou transmitidas como uma ou mais instruções de código ou um meio legível por computador e executadas por uma unidade de processamento com base em hardware. Meios legíveis por computador podem incluir meios de armazenamento legíveis por computador, que correspondem a um meio tangível, tal como mídia de armazenamento de dados ou mídia de comunicação, incluindo qualquer meio que facilite a

transferência de um programa de computador de um lugar para outro, por exemplo, de acordo com um protocolo de comunicação. Deste modo, meios legíveis por computador podem geralmente corresponder a (1) um meio de armazenamento legível por computador tangível que é não transitório ou (2) um meio de comunicação, tal como um sinal ou onda de portadora. Meios de armazenamento de dados podem ser qualquer material disponível que pode ser acessado por um ou mais computadores ou um ou mais processadores para recuperar instruções de código, e/ou estruturas de dados para a implementação das técnicas descritas nesta divulgação. Um produto de programa de computador pode incluir um meio legível por computador.

[0168] A título de exemplo, e não como limitação, tais meios de armazenamento legíveis por computador podem compreender RAM, ROM, EEPROM, CD-ROM ou outro armazenamento em disco óptico, armazenamento em disco magnético, ou outros dispositivos de armazenamento magnéticos, memória flash, ou qualquer outro meio que pode ser usado para armazenar o código de programa desejado sob a forma de instruções ou estruturas de dados, e que pode ser acessado por um computador. Além disso, qualquer ligação é denominada adequadamente um meio legível por computador. Por exemplo, se as instruções são transmitidas de um site, servidor ou outra fonte remota utilizando um cabo coaxial, cabo de fibra óptica, par trançado, linha de assinante digital (DSL) ou tecnologias sem fio, tais como infravermelhos, rádio e micro-ondas, então o cabo coaxial, cabo de fibra óptica, par trançado, DSL, ou tecnologias sem fio, tais como infravermelho, rádio e micro-ondas estão incluídos na definição de meio. Deve ser entendido, no entanto, que os meios de armazenamento legíveis por computador e meios de armazenamento de dados não incluem

conexões, ondas de portadoras, sinais ou outros meios de comunicação transitória, mas são direcionados para meios de armazenamento tangível, não transitórios. Disco e disquete, como aqui utilizado, inclui disco compacto (CD), disco laser, disco óptico, disco versátil digital (DVD), disquete e disco Blu-ray, onde disquetes geralmente reproduzem dados magneticamente, enquanto que discos reproduzem dados opticamente com lasers. Combinações dos anteriores também devem ser incluídas dentro do âmbito dos meios legível por computador.

[0169] As instruções podem ser executadas por um ou mais processadores, tais como um ou mais processadores de sinal digital (DSPs), microprocessadores de uso geral, circuitos integrados de aplicação específica (ASIC), matrizes de portas de programáveis em campo (FPGA), ou outro conjunto de circuitos integrados equivalentes ou lógicos discretos. Por conseguinte, o termo "processador" tal como aqui utilizado pode referir-se a qualquer uma da estrutura precedente ou qualquer outra estrutura adequada para aplicação das técnicas aqui descritas. Além disso, em alguns aspectos, a funcionalidade aqui descrita pode ser provida dentro de módulos de hardware e/ou software dedicado configurados para a codificação e decodificação, ou incorporados em um codec combinado. Além disso, as técnicas podem ser totalmente implementadas em um ou mais circuitos ou elementos lógicos.

[0170] As técnicas da presente divulgação podem ser implementadas em uma vasta variedade de dispositivos ou aparelhos, incluindo um aparelho telefônico sem fio, um circuito integrado (IC) ou um conjunto de ICs (por exemplo, um conjunto de chips). Vários componentes, módulos ou unidades são descritos nesta divulgação para enfatizar os aspectos funcionais de dispositivos configurados para

executar as técnicas divulgadas, mas não precisam necessariamente de realização por diferentes unidades de hardware. Em vez disso, tal como descrito acima, várias unidades podem ser combinadas em uma unidade de hardware codec ou providas por um conjunto de unidades de hardware interoperativo, incluindo um ou mais processadores, conforme descrito acima, em conjunto com o software e/ou firmware adequado.

[0171] Vários exemplos foram descritos. Estes e outros exemplos estão dentro do âmbito das reivindicações seguintes.

REIVINDICAÇÕES

1. Método para decodificar dados de vídeo, compreendendo:

decodificar (200) dados para uma primeira imagem de uma primeira camada de dados de vídeo de multicamada, em que os dados decodificados indicam uma redefinição de contagem de ordem de imagem, POC, para um valor de POC da primeira imagem, e em que a primeira imagem está incluída em uma unidade de acesso;

o método **caracterizado** pelo fato de que compreende adicionalmente, com base nos dados decodificados para a primeira imagem, e antes de decodificar a primeira imagem, diminuir (202) valores de POC de todas as imagens armazenadas em um *buffer* de imagem decodificada, DPB, que precedem a primeira imagem na ordem de decodificação incluindo pelo menos uma imagem de uma segunda camada dos dados de vídeo de multicamada.

2. Método para codificar dados de vídeo, compreendendo:

codificar dados para uma primeira imagem de uma primeira camada de dados de vídeo de multicamada, em que os dados codificados indicam uma redefinição de contagem de ordem de imagem, POC, para um valor de POC da primeira imagem, e em que a primeira imagem está incluída em uma unidade de acesso;

o método **caracterizado** pelo fato de que compreende adicionalmente, com base nos dados codificados para a primeira imagem, e antes de codificar a primeira imagem, diminuir (105) valores de POC de todas as imagens armazenadas em um *buffer* de imagem decodificada, DPB, que

precedem a primeira imagem na ordem de decodificação incluindo pelo menos uma imagem de uma segunda camada dos dados de vídeo de multicamada.

3. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que diminuir os valores de POC de todas as imagens armazenadas no DPB que precedem a primeira imagem na ordem de decodificação compreende diminuir os valores de POC apenas uma vez durante a codificação da unidade de acesso, ou alternativamente, em que diminuir os valores de POC compreende diminuir os valores de POC depois de decodificar um cabeçalho de fatia para uma fatia inicial de dados de vídeo da primeira imagem.

4. Método, de acordo com a reivindicação 2, **caracterizado** pelo fato de que diminuir os valores de POC de todas as imagens armazenadas no DPB que precedem a primeira imagem na ordem de decodificação compreende diminuir os valores de POC apenas uma vez durante a codificação da unidade de acesso, ou alternativamente, em que diminuir os valores de POC compreende diminuir os valores de POC depois de codificar um cabeçalho de fatia para uma fatia inicial dos dados de vídeo da primeira imagem.

5. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que decodificar os dados que indicam a redefinição de POC compreende decodificar dados que representam um índice de redefinição de POC para uma fatia da primeira imagem de dados de vídeo.

6. Método, de acordo com a reivindicação 2, **caracterizado** pelo fato de que codificar os dados que

indicam a redefinição de POC compreende codificar dados que representam um índice de redefinição de POC para uma fatia da primeira imagem de dados de vídeo.

7. Método, de acordo com a reivindicação 1 ou reivindicação 2, **caracterizado** pelo fato de que os dados que indicam a redefinição de POC compreendem dados que indicam um período de redefinição de POC, e em que a unidade de acesso é a primeira unidade de acesso em ordem de decodificação do período de redefinição de POC; ou alternativamente, em que a primeira imagem é uma imagem associada com um identificador de camada mais baixa que está presente na unidade de acesso, e em que diminuir os valores de POC compreende diminuir os valores de POC usando um valor de diminuição de POC comum.

8. Método, de acordo com a reivindicação 7, **caracterizado** pelo fato de que compreende adicionalmente decodificar um elemento de sintaxe DeltaPocVal que indica um valor de diminuição de POC para diminuir os valores de POC.

9. Dispositivo para decodificar dados de vídeo (30), compreendendo:

uma memória compreendendo um *buffer* de imagem decodificada, DPB, configurado para armazenar dados de vídeo de multicamada; e

um codificador de vídeo configurado para:

decodificar dados para uma primeira imagem de uma primeira camada de dados de vídeo de multicamada, em que os dados decodificados indicam uma redefinição de contagem de ordem de imagem, POC, para um valor de POC da primeira

imagem, e em que a primeira imagem está incluída em uma unidade de acesso;

o dispositivo para decodificar dados de vídeo **caracterizado** pelo fato de que é configurado adicionalmente para:

com base nos dados decodificados para a primeira imagem, e antes de decodificar a primeira imagem, diminuir valores de POC de todas as imagens armazenadas no DPB que precedem a primeira imagem na ordem de decodificação incluindo pelo menos uma imagem de uma segunda camada dos dados de vídeo de multicamada.

10. Dispositivo para codificar dados de vídeo (20), compreendendo:

uma memória compreendendo um *buffer* de imagem decodificada, DPB, configurado para armazenar dados de vídeo de multicamada; e

um codificador de vídeo configurado para:

codificar dados para uma primeira imagem de uma primeira camada de dados de vídeo de multicamada,

em que os dados codificados indicam uma redefinição de contagem de ordem de imagem, POC, para um valor de POC da primeira imagem, e em que a primeira imagem está incluída em uma unidade de acesso;

o dispositivo para codificar dados de vídeo **caracterizado** pelo fato de que é configurado adicionalmente para:

com base nos dados codificados para a primeira imagem, e antes de codificar a primeira imagem, diminuir valores de POC de todas as imagens armazenadas no DPB que precedem a primeira imagem na ordem de decodificação

incluindo pelo menos uma imagem de uma segunda camada dos dados de vídeo de multicamada.

11. Dispositivo de acordo com a reivindicação 9, **caracterizado** pelo fato de que a unidade de acesso inclui a primeira imagem e pelo menos uma segunda imagem, em que os dados indicando a redefinição de POC compreendem um índice de redefinição de POC, e em que o decodificador de vídeo é configurado adicionalmente para decodificar o mesmo índice de redefinição de POC para todas as outras imagens da unidade de acesso incluindo a pelo menos uma segunda imagem da unidade de acesso.

12. Dispositivo, de acordo com a reivindicação 10, **caracterizado** pelo fato de que a unidade de acesso inclui a primeira imagem e pelo menos uma segunda imagem, em que os dados indicando a redefinição de POC compreendem um índice de redefinição de POC, e em que o decodificador de vídeo é configurado adicionalmente para codificar o mesmo índice de redefinição de POC para todas as outras imagens da unidade de acesso incluindo a pelo menos uma segunda imagem da unidade de acesso.

13. Dispositivo, de acordo com a reivindicação 9 ou 10, **caracterizado** pelo fato de que diminuir os valores de POC compreende diminuir um dentre diminuir todos os bits dos valores de POC, diminuir os bits mais significativos dos valores de POC, ou diminuir os bits menos significativos dos valores de POC, ou

em que os dados que indicam o valor de redefinição de POC estão associados com um período de redefinição de POC que é aplicável a todas as camadas dos dados de vídeo de multicamada.

14. Dispositivo, de acordo com a reivindicação 9 ou 10, **caracterizado** pelo fato de que o dispositivo compreende pelo menos um dentre:

um circuito integrado;

um microprocessador; ou

um dispositivo de comunicação sem fio.

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

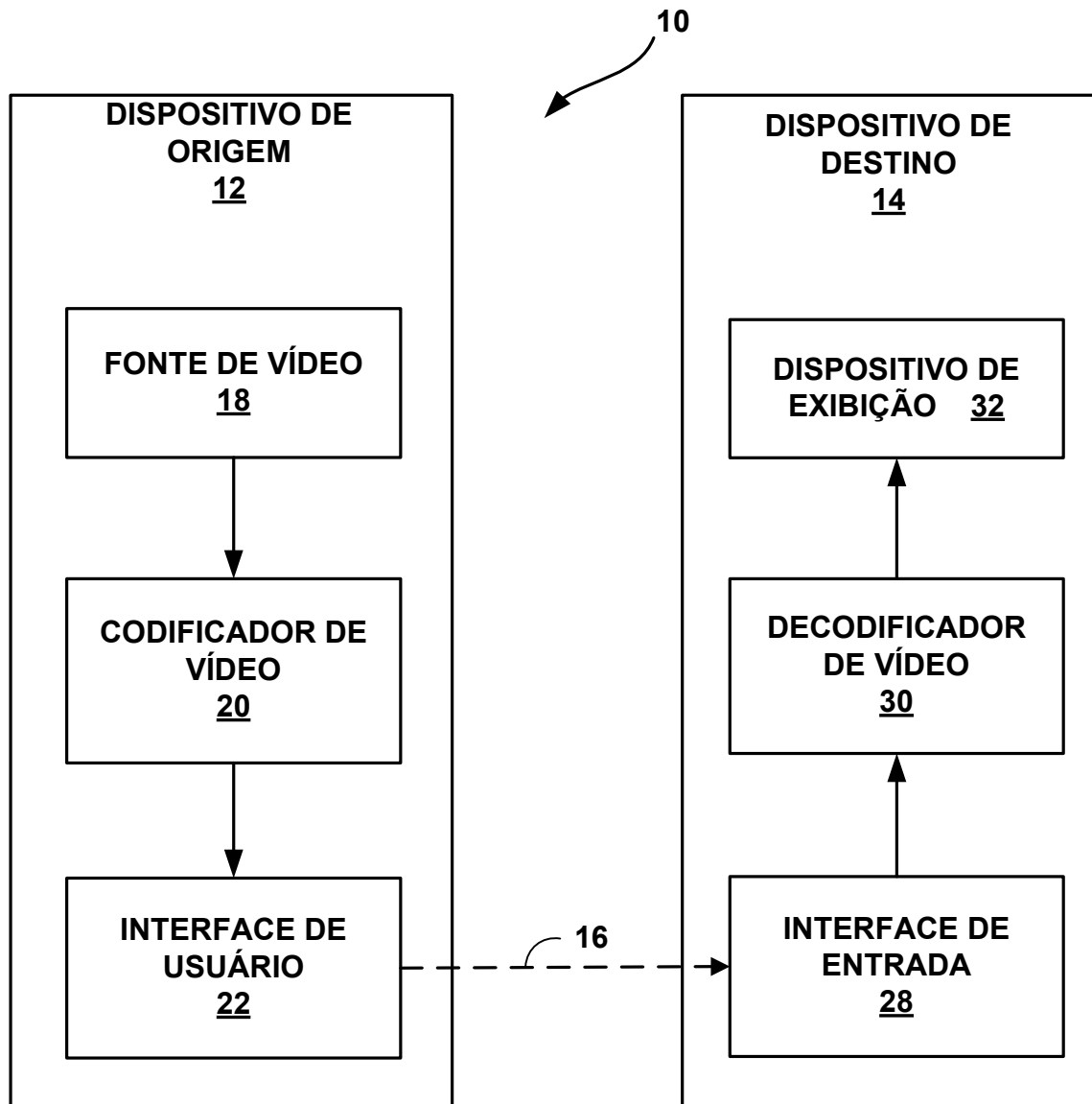


FIG. 1

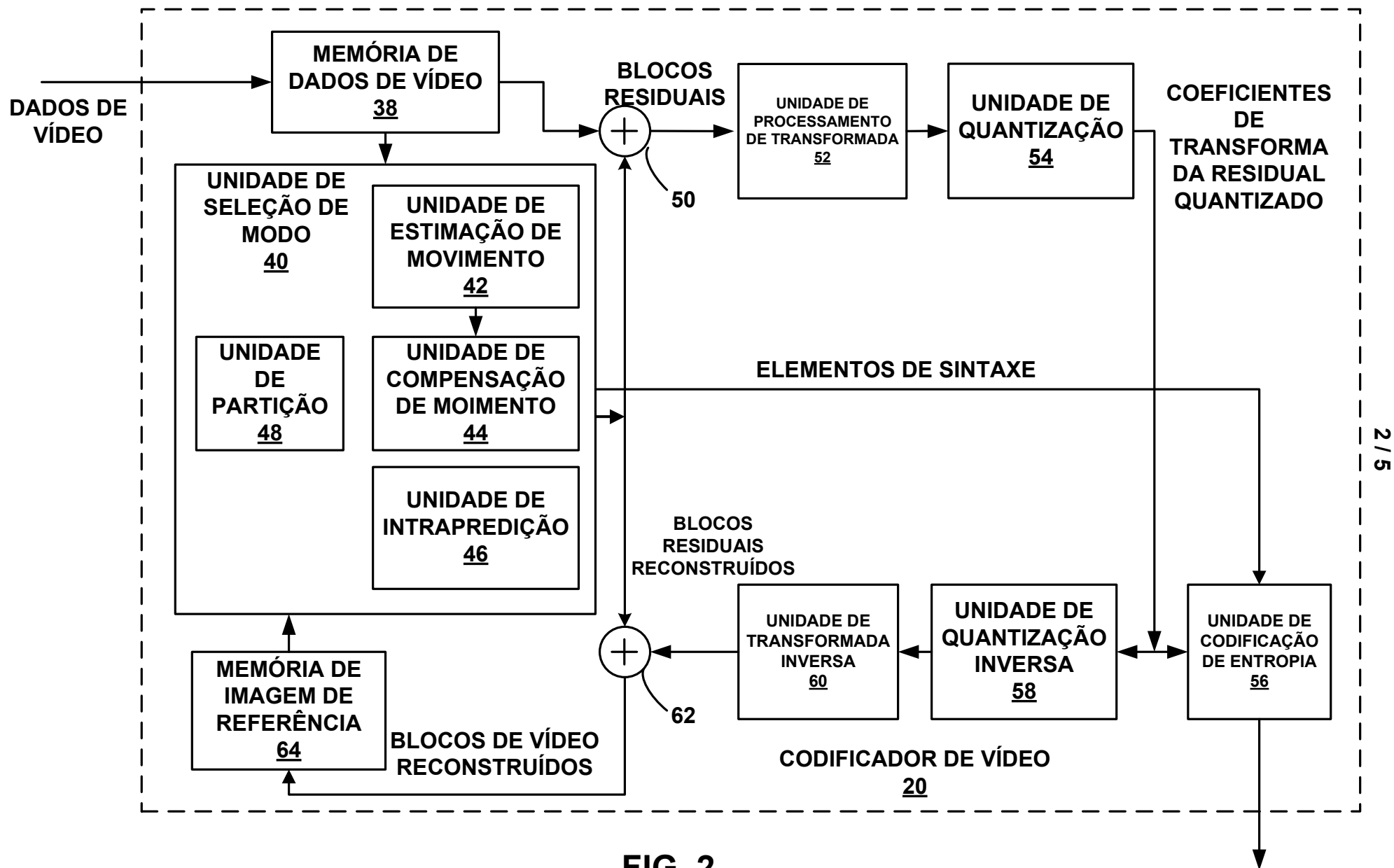


FIG. 2

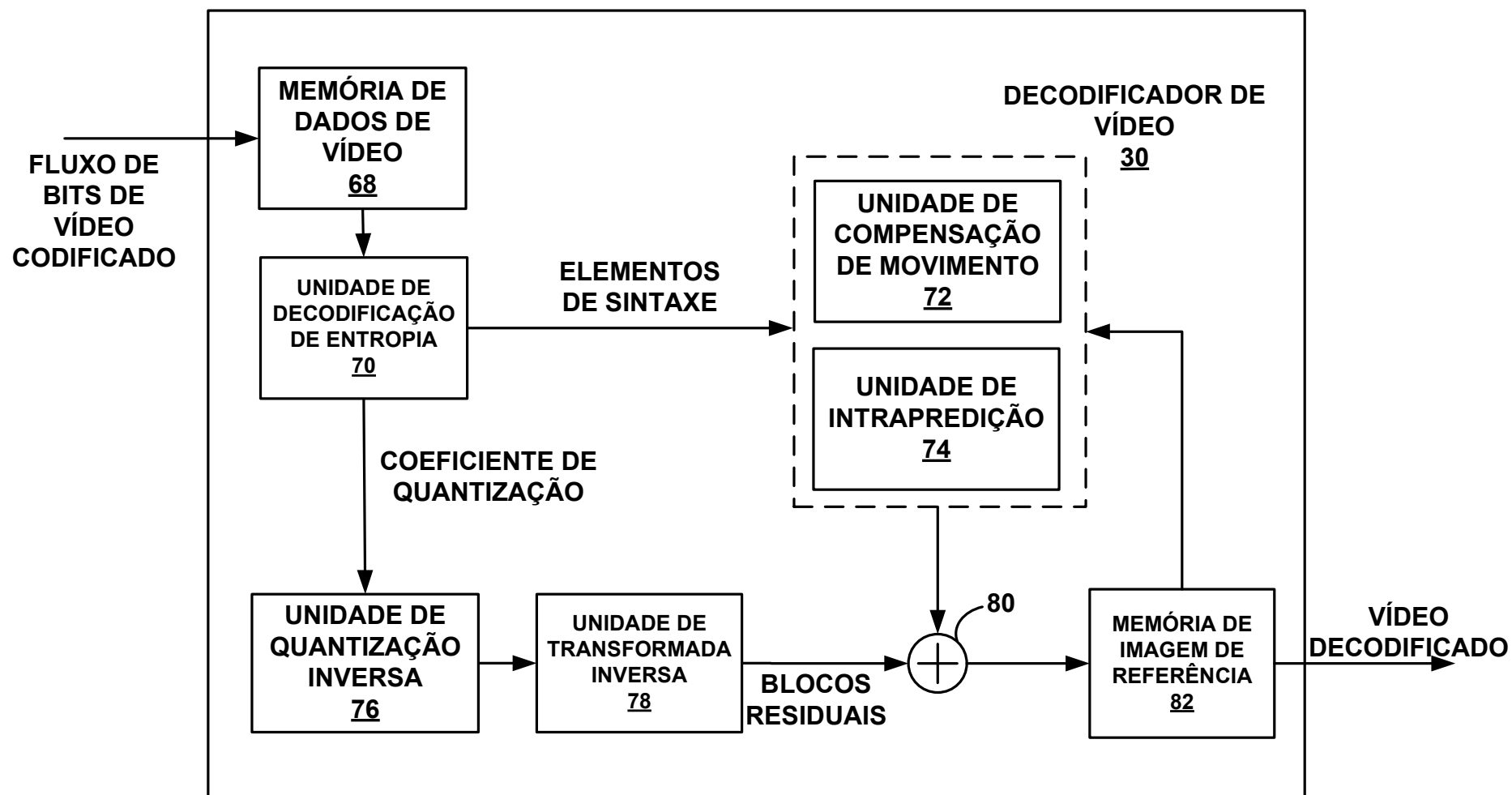


FIG. 3

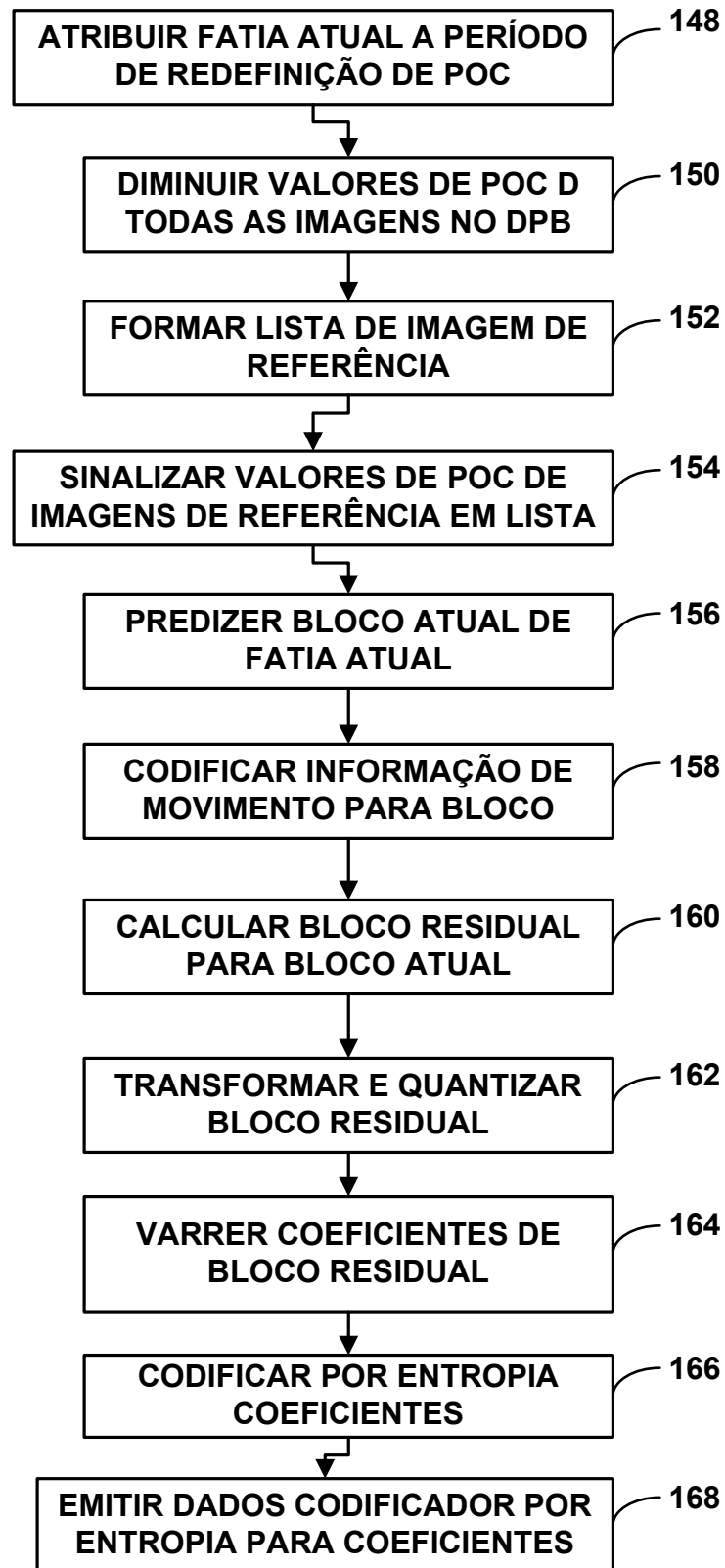


FIG. 4

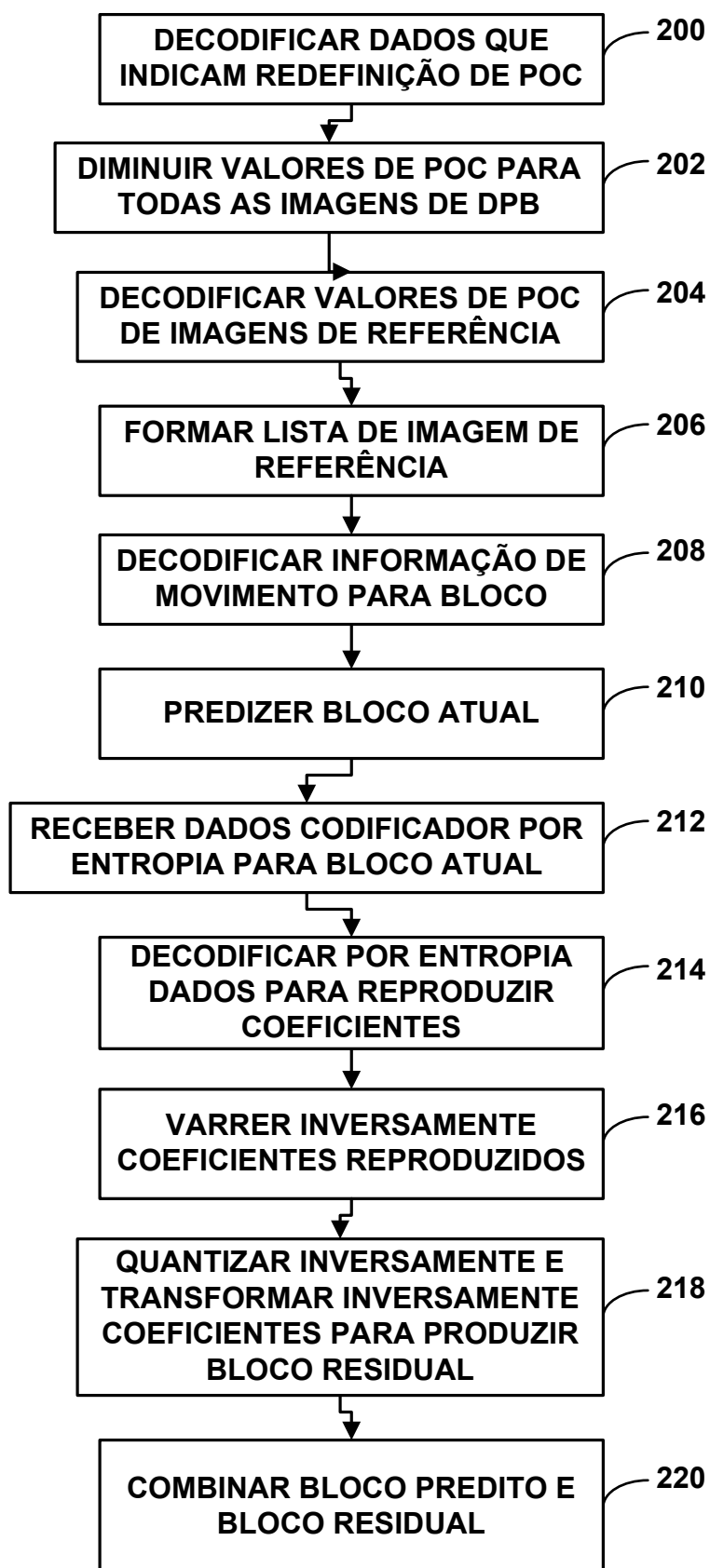


FIG. 5