



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

(21) BR 112021009732-5 A2



(22) Data do Depósito: 06/12/2019

(43) Data da Publicação Nacional: 17/08/2021

(54) Título: PADRÕES DE PREDIÇÃO DE VETOR DE MOVIMENTO ESPAÇO-TEMPORAL PARA CODIFICAÇÃO DE VÍDEO

(51) Int. Cl.: H04N 19/52.

(30) Prioridade Unionista: 05/12/2019 US 16/704,827; 06/12/2018 US 62/776,373.

(71) Depositante(es): QUALCOMM INCORPORATED.

(72) Inventor(es): DMYTRO RUSANOVSKYY; NIKOLAY MIKHAYLOVICH SHLYAKHOV; WEI-JUNG CHIEN; MARTA KARCZEWICZ.

(86) Pedido PCT: PCT US2019064953 de 06/12/2019

(87) Publicação PCT: WO 2020/118191 de 11/06/2020

(85) Data da Fase Nacional: 19/05/2021

(57) Resumo: A presente invenção refere-se a técnicas para construir listas de preditores de vetor de movimento com base em blocos espacialmente vizinhos e blocos colocalizados. Um método de codificar dados de vídeo inclui, para um primeiro bloco, em uma primeira imagem tendo uma ordem de codificação da esquerda para a direita, construir uma primeira lista de preditores de vetor de movimento, em que uma primeira entrada na primeira lista de preditores de vetor de movimento baseia-se em informações de vetor de movimento de um bloco vizinho à esquerda do primeiro bloco, codificar o primeiro bloco na primeira imagem com base na primeira lista de preditores de vetor de movimento, para um segundo bloco, em uma segunda imagem tendo uma ordem de codificação da direita para a esquerda, construir uma segunda lista de preditores de vetor de movimento, em que uma primeira entrada na segunda lista de preditores de vetor de movimento baseia-se em informações de vetor de movimento de um bloco vizinho à direita do segundo bloco, e codificar o segundo bloco na segunda imagem com base na segunda lista de preditores de vetor de movimento.

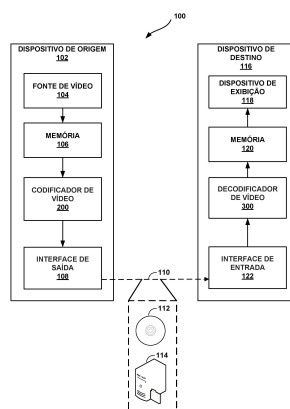


FIG. 1

**“PADRÕES DE PREDIÇÃO DE VETOR DE MOVIMENTO ESPAÇO-TEMPORAL
PARA CODIFICAÇÃO DE VÍDEO”**

[0001] Este pedido reivindica prioridade para Pedido dos EUA N°. 16/704,827, depositado em 05 de dezembro de 2019, que reivindica o benefício do Pedido Provisório dos EUA N°. 62/776,373, depositado em 6 de dezembro de 2018, todo o conteúdo dos quais é incorporado por referência neste documento.

CAMPO TÉCNICO

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

FUNDAMENTO

[0003] Os recursos de vídeo digital podem ser incorporados a uma ampla gama de dispositivos, incluindo televisores digitais, sistemas de radiodifusão digital direta, sistemas de radiodifusão sem fio, assistentes digitais pessoais (PDAs), laptops ou desktops, tablets, leitores de livro digital, câmeras digitais, dispositivos de gravação digital, reprodutores de mídia digital, dispositivos de videogame, consoles de videogame, telefones celulares ou de rádio via satélite, os chamados “smartphones”, dispositivos de teleconferência de vídeo, dispositivos de fluxo contínuo de vídeo e semelhantes. Dispositivos de vídeo digital implementam técnicas de codificação de vídeo, tais como aquelas descritas nos padrões definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificação de Vídeo Avançada (AVC), padrão de Codificação de Vídeo de Alta Eficiência (HEVC), ITU-T H.265/Codificação de Vídeo de Alta Eficiência

(HEVC), e extensões de tais padrões. Os dispositivos de vídeo podem transmitir, receber, codificar, decodificar e/ou armazenar informações de vídeo digital de forma mais eficiente, implementando tais técnicas de codificação de vídeo.

[0004] As técnicas de codificação de vídeo incluem predição espacial (intraimagem) e/ou predição temporal (interimagem) para reduzir ou remover a redundância inerente em sequências de vídeo. Para codificação de vídeo baseada em bloco, uma fatia de vídeo (por exemplo, uma imagem de vídeo ou uma porção de uma imagem de vídeo) pode ser particionada em blocos de vídeo, que também podem ser referidos como unidades de árvore de codificação (CTUs), unidades de codificação (CUs) e/ou nós codificadores. Os blocos de vídeo em uma fatia intracodificada (I) de uma imagem são codificados usando predição espacial com respeito a amostras de referência em blocos vizinhos na mesma imagem. Os blocos de vídeo em uma fatia intercodificada (P ou B) de uma imagem podem usar predição espacial com respeito a amostras de referência em blocos vizinhos na mesma imagem ou predição temporal com respeito a amostras de referência em outras imagens de referência. As imagens podem ser referidas como quadros, e as imagens de referência podem ser referidas como quadros de referência.

SUMÁRIO

[0005] Em geral, esta divulgação descreve técnicas para interpredição e predição de vetor de movimento em codificação de vídeo. Mais especificamente,

esta divulgação descreve técnicas para predição de vetor de movimento espaço-temporal, tal como predição de vetor de movimento com base em uma lista de preditores de vetor de movimento. As técnicas desta divulgação podem ser usadas com qualquer um dos codecs de vídeo existentes, tais como H.265/HEVC (Codificação de Vídeo de Alta Eficiência), ou podem ser uma ferramenta de codificação eficiente em quaisquer padrões de codificação de vídeo futuros, tais como H.266/VVC (Codificação de Vídeo Versátil).

[0006] Como descrito em mais detalhes, a divulgação descreve exemplos técnicas de construção de uma lista de preditores de vetor de movimento que inclui informações de vetor de movimento de blocos espacialmente e temporalmente vizinhos. Em alguns casos, a ordem de codificação de uma imagem (por exemplo, esquerda para direita ou direita para esquerda) pode impactar quais blocos espacialmente vizinhos estão disponíveis para construir a lista de preditores de vetor de movimento. Por conseguinte, esta divulgação descreve exemplo técnicas pelas quais um codificador de vídeo (por exemplo, codificador de vídeo ou decodificador de vídeo) pode construir a lista de preditores de vetor de movimento com base na ordem de codificação usada para uma imagem. Além disso, para blocos temporalmente vizinhos (também chamados de blocos colocalizados), o codificador de vídeo pode ser configurado para acessar e verificar disponibilidade de informações de vetor de movimento dos blocos temporalmente vizinhos em uma ordem específica.

[0007] Em um exemplo, a divulgação descreve um

método de codificar dados de vídeo, o método compreendendo para um primeiro bloco, em uma primeira imagem tendo uma ordem de codificação da esquerda para a direita, construir uma primeira lista de preditores de vetor de movimento, em que uma primeira entrada na primeira lista de preditores de vetor de movimento baseia-se em informações de vetor de movimento de um bloco vizinho à esquerda do primeiro bloco, codificar o primeiro bloco na primeira imagem com base na primeira lista de preditores de vetor de movimento, para um segundo bloco, em uma segunda imagem tendo uma ordem de codificação da direita para a esquerda, construir uma segunda lista de preditores de vetor de movimento, em que uma primeira entrada na segunda lista de preditores de vetor de movimento baseia-se em informações de vetor de movimento de um bloco vizinho à direita do segundo bloco, e codificar o segundo bloco na segunda imagem com base na segunda lista de preditores de vetor de movimento.

[0008] Em um exemplo, a divulgação descreve um dispositivo para codificar dados de vídeo, o dispositivo compreendendo memória configurada para armazenar informações de vetor de movimento e circuitos de processamento acoplados à memória. Os circuitos de processamento são configurados para, para um primeiro bloco, em uma primeira imagem tendo uma ordem de codificação da esquerda para a direita, construir uma primeira lista de preditores de vetor de movimento, em que uma primeira entrada na primeira lista de preditores de vetor de movimento baseia-se em informações de vetor de movimento de um bloco vizinho à esquerda do primeiro bloco

armazenado na memória, codificar o primeiro bloco na primeira imagem com base na primeira lista de preditores de vetor de movimento, para um segundo bloco, em uma segunda imagem tendo uma ordem de codificação da direita para a esquerda, construir uma segunda lista de preditores de vetor de movimento, em que uma primeira entrada na segunda lista de preditores de vetor de movimento baseia-se em informações de vetor de movimento de um bloco vizinho à direita do segundo bloco armazenado na memória, e codificar o segundo bloco na segunda imagem com base na segunda lista de preditores de vetor de movimento.

[0009] Em um exemplo, a divulgação descreve um meio de armazenamento legível por computador nele armazenando instruções que, quando executadas, levam um ou mais processadores para um primeiro bloco, em uma primeira imagem tendo uma ordem de codificação da esquerda para a direita, construir uma primeira lista de preditores de vetor de movimento, em que uma primeira entrada na primeira lista de preditores de vetor de movimento baseia-se em informações de vetor de movimento de um bloco vizinho à esquerda do primeiro bloco, codificar o primeiro bloco na primeira imagem com base na primeira lista de preditores de vetor de movimento, para um segundo bloco, em uma segunda imagem tendo uma ordem de codificação da direita para a esquerda, construir uma segunda lista de preditores de vetor de movimento, em que uma primeira entrada na segunda lista de preditores de vetor de movimento baseia-se em informações de vetor de movimento de um bloco vizinho à direita do segundo bloco, e codificar o segundo bloco na

segunda imagem com base na segunda lista de preditores de vetor de movimento.

[0010] Os detalhes de um ou mais exemplos são apresentados nos desenhos anexos e na descrição abaixo. Outras características, objetos e vantagens serão evidentes a partir da descrição, desenhos e reivindicações.

BREVE DESCRIÇÃO DOS DESENHOS

[0011] A Figura 1 é um diagrama de blocos ilustrando um sistema de codificação e decodificação de vídeo exemplificativo que pode executar as técnicas desta divulgação.

[0012] A Figura 2 é um diagrama de blocos ilustrando um codificador de vídeo exemplificativo que pode executar as técnicas desta divulgação.

[0013] A Figura 3 é um diagrama de blocos ilustrando um decodificador de vídeo exemplificativo que pode executar as técnicas desta divulgação.

[0014] A Figura 4A é um diagrama conceitual mostrando candidatos vizinhos espaciais para modo de mesclagem.

[0015] A Figura 4B é um diagrama conceitual mostrando candidatos vizinhos espaciais para modo de predição de vetor de movimento avançado (AMVP).

[0016] A Figura 5A é um diagrama conceitual mostrando um candidato a preditor de vetor de movimento temporal (TMVP).

[0017] A Figura 5B é um diagrama conceitual mostrando escala de vetor de movimento para TMVP.

[0018] A Figura 6 é um diagrama conceitual

mostrando pesquisa e carregamento (*fetching*) de candidatos de mesclagem espacial não vizinho.

[0019] A Figura 7A é um diagrama conceitual ilustrando candidatos espaciais e temporais exemplificativos usados em predição de vetor de movimento.

[0020] A Figura 7B é um diagrama conceitual ilustrando uma ordem de visitação exemplificativa para candidatos espaciais.

[0021] A Figura 7C é um diagrama conceitual ilustrando outra ordem de visitação exemplificativa para candidatos espaciais.

[0022] A Figura 8A é um diagrama conceitual ilustrando um padrão de preditor de vetor de movimento espacial temporal exemplificativo de acordo com um exemplo da divulgação.

[0023] A Figura 8B é um diagrama conceitual ilustrando um padrão de preditor de vetor de movimento espacial temporal invertido exemplificativo de acordo com um exemplo da divulgação.

[0024] A Figura 8C é um diagrama conceitual ilustrando outro padrão de preditor de vetor de movimento espacial temporal invertido exemplificativo de acordo com um exemplo da divulgação.

[0025] A Figura 9 é um fluxograma ilustrando um método de codificação exemplificativo.

DESCRIÇÃO DETALHADA

[0026] Em alguns exemplos, codificação de vídeo (por exemplo, codificação de vídeo ou decodificação de vídeo) pode incluir técnicas de interpredição e/ou cópia

intrabloco (IBC). Em intrapredição ou IBC, um codificador de vídeo determina um bloco de predição com base em um vetor de movimento (para IBC, o vetor de movimento pode ser um vetor de bloco) para um bloco atual, determina informações residuais (por exemplo, diferença) entre o bloco de predição e o bloco atual, e sinaliza as informações residuais. Um decodificador de vídeo recebe as informações residuais. Além disso, o decodificador de vídeo determina um vetor de movimento para o bloco atual e determina o bloco de predição com base no vetor de movimento. O decodificador de vídeo adiciona as informações residuais ao bloco de predição para reconstruir o bloco atual.

[0027] Uma forma em que o decodificador de vídeo determina o vetor de movimento para o bloco atual baseia-se em uma lista de preditores de vetor de movimento. Ambos o codificador de vídeo e o decodificador de vídeo utilizam um processo similar para construir respectivas listas de preditores de vetor de movimento, tal que a lista de preditores de vetor de movimento construída pelo codificador de vídeo e a lista de preditores de vetor de movimento construída pelo decodificador de vídeo sejam a mesma. A lista de preditores de vetor de movimento inclui informações de vetor de movimento de blocos previamente codificados, tal como blocos espacialmente vizinhos (por exemplo, blocos que são vizinhos ao bloco atual na mesma imagem que o bloco atual) e blocos colocalizados (por exemplo, blocos que estão localizados em locais específicos em outras imagens). Em alguns exemplos, uma lista de

preditores de vetor de movimento pode incluir informações de vetor de movimento artificialmente geradas (por exemplo, informações de vetor de movimento que não são de um bloco previamente codificado).

[0028] O codificador de vídeo determina uma entrada na lista de preditores de vetor de movimento e sinaliza informações indicativas da entrada. O decodificador de vídeo determina informações de vetor de movimento da lista de preditores de vetor de movimento com base na entrada e determina o vetor de movimento para o bloco atual com base nas informações de vetor de movimento determinadas. Como um exemplo, o decodificador de vídeo pode definir o vetor de movimento para o bloco atual igual às informações de vetor de movimento determinadas (por exemplo, tal como em modo de mesclagem). Como outro exemplo, o decodificador de vídeo pode adicionar uma diferença de vetor de movimento (MVD), sinalizada pelo codificador de vídeo, às informações de vetor de movimento determinadas para determinar o vetor de movimento para o bloco atual (por exemplo, tal como em modo de predição de vetor de movimento avançado (AMVP)).

[0029] Como descrito acima, a lista de preditores de vetor de movimento pode incluir informações de vetor de movimento para blocos espacialmente vizinhos. Em alguns exemplos, para as informações de vetor de movimento de um bloco espacialmente vizinho estarem disponíveis, o bloco espacialmente vizinho foi previamente codificado (por exemplo, codificado ou decodificado). Se um bloco vizinho não tiver sido ainda codificado, então, o codificador de

vídeo e o decodificador de vídeo podem não ter sido ainda determinados e podem ainda não ter as informações necessárias para determinar as informações de vetor de movimento para esse bloco vizinho já codificado. Portanto, em alguns exemplos, os únicos blocos espacialmente vizinhos que podem ser verificados para informações de vetor de movimento são blocos previamente codificados.

[0030] No entanto, quais blocos são blocos previamente codificados podem ser com base na ordem de codificação da imagem. Por exemplo, o codificador de vídeo (por exemplo, codificador de vídeo ou decodificador de vídeo) pode codificar uma primeira imagem da esquerda para direita e de cima para baixo. Codificar a primeira imagem da esquerda para direita e de cima para baixo pode se referir ao codificador de vídeo codificar bloco por bloco dentro da primeira imagem começando do bloco superior-esquerdo na primeira imagem e codificar blocos em uma direção para a direita até o codificador de vídeo atingir o limite direito da primeira imagem. Então, o codificador de vídeo retorna ao bloco abaixo do bloco superior-esquerdo na primeira imagem e codifica bloco por bloco até o codificador de vídeo atingir o limite direito da primeira imagem. O codificador de vídeo repete essas operações até o codificador de vídeo codificar o bloco inferior-direito da primeira imagem.

[0031] A ordem de codificação não precisa ser necessariamente da esquerda para a direita e de cima para baixo. O codificador de vídeo pode codificar uma segunda imagem da direita para a esquerda e de cima para baixo.

Codificar a segunda imagem da direita para a esquerda e de cima para baixo pode se referir ao codificador de vídeo codificar bloco por bloco dentro da segunda imagem começando do bloco superior-direito na segunda imagem e codificar blocos em uma direção para a esquerda até o codificador de vídeo atingir o limite esquerdo da segunda imagem. Então, o codificador de vídeo retorna ao bloco abaixo do bloco superior-direito na segunda imagem e codifica bloco por bloco até o codificador de vídeo atingir o limite esquerdo da segunda imagem. O codificador de vídeo repete essas operações até o codificador de vídeo codificar o bloco inferior-esquerdo da segunda imagem.

[0032] Devido às diferentes ordens de codificação para a primeira e a segunda imagens, pode haver diferentes blocos espacialmente vizinhos que foram previamente codificados para um bloco na primeira imagem em comparação com um bloco na segunda imagem. Como um exemplo, para um primeiro bloco, na primeira imagem tendo uma ordem de codificação da esquerda para a direita, um bloco vizinho à esquerda do primeiro bloco pode ter já sido codificado antes de o primeiro bloco ser codificado e, portanto, as informações de vetor de movimento para o bloco vizinho à esquerda podem estar disponíveis (por exemplo, previamente determinadas). Além disso, para o primeiro bloco, um bloco vizinho à direita do primeiro bloco pode não ter sido ainda codificado e, portanto, as informações de vetor de movimento para o bloco vizinho à direita podem não estar disponíveis (por exemplo, não ainda determinadas).

[0033] Para um segundo bloco, na segunda imagem

tendo uma ordem de codificação da direita para a esquerda, o oposto do primeiro bloco na primeira imagem podem ser verdadeiro. Por exemplo, para o segundo bloco, na segunda imagem tendo uma ordem de codificação da direita para a esquerda, um bloco vizinho à esquerda do segundo bloco pode não ter sido ainda codificado e, portanto, as informações de vetor de movimento para o bloco vizinho à esquerda podem não estar disponíveis (por exemplo, não ainda determinadas). Além disso, para o segundo bloco, um bloco vizinho à direita do segundo bloco pode ter já sido codificado antes de o segundo bloco ser codificado e, portanto, as informações de vetor de movimento para o bloco vizinho à direita podem estar disponíveis (por exemplo, previamente determinadas).

[0034] Uma vez que quais blocos espacialmente vizinhos foram codificados pode se basear na ordem de codificação da imagem, as técnicas para construir a lista de preditores de vetor de movimento para o primeiro bloco na primeira imagem tendo a ordem de codificação da esquerda para a direita e as técnicas para construir a lista de preditores de vetor de movimento para o segundo bloco na segunda imagem tendo a ordem de codificação da direita para a esquerda podem ser diferentes. Em um ou mais exemplos descritos nesta divulgação, as posições do conjunto de blocos espacialmente vizinhos cujas informações de vetor de movimento formam a lista de preditores de vetor de movimento para o primeiro bloco podem ser invertidas em relação às posições do conjunto de blocos espacialmente vizinhos cujas informações de vetor de movimento formam a

lista de preditores de vetor de movimento para o segundo bloco.

[0035] Por exemplo, para um primeiro bloco, em uma primeira imagem tendo uma ordem de codificação da esquerda para a direita, o codificador de vídeo pode construir uma primeira lista de preditores de vetor de movimento, tal que uma primeira entrada na primeira lista de preditores de vetor de movimento baseia-se em informações de vetor de movimento de um bloco vizinho à esquerda do primeiro bloco. Para um segundo bloco, em uma segunda imagem tendo uma ordem de codificação da direita para a esquerda, o codificador de vídeo pode construir uma segunda lista de preditores de vetor de movimento, tal que uma primeira entrada na segunda lista de preditores de vetor de movimento baseia-se em informações de vetor de movimento de um bloco vizinho à direita do segundo bloco. O bloco vizinho à esquerda em relação ao primeiro bloco e o bloco vizinho à direita em relação ao segundo bloco podem ser considerados invertidos um em relação ao outro (por exemplo, bloco vizinho à esquerda está na esquerda e bloco vizinho à direita está na direita).

[0036] Em alguns exemplos, o codificador de vídeo pode construir a primeira lista de preditores de vetor de movimento, tal que a primeira lista de preditores de vetor de movimento tem informações de vetor de movimento na seguinte ordem: informações de vetor de movimento do bloco vizinho à esquerda, informações de vetor de movimento de um bloco vizinho acima, informações de vetor de movimento de um bloco vizinho acima à direita, informações de vetor de

movimento de um bloco vizinho abaixo à esquerda, e informações de vetor de movimento de um bloco vizinho acima à esquerda. O codificador de vídeo pode construir a segunda lista de preditores de vetor de movimento, tal que a segunda lista de preditores de vetor de movimento tem informações de vetor de movimento na seguinte ordem: informações de vetor de movimento do bloco vizinho à direita, informações de vetor de movimento de um bloco vizinho acima, informações de vetor de movimento de um bloco vizinho acima à esquerda, informações de vetor de movimento de um bloco vizinho abaixo à direita, e informações de vetor de movimento de um bloco vizinho acima à direita.

[0037] No exemplo acima, a primeira e segunda listas de preditores de vetor de movimento têm uma ordem específica em que as informações de vetor de movimento para os blocos espacialmente vizinhos são dispostas na(s) lista(s) de preditor de vetor de movimento. Embora haja uma ordem específica em que as informações de vetor de movimento são dispostas, o codificador de vídeo pode ser configurado para acessar as informações de vetor de movimento dos blocos espacialmente vizinhos em paralelo. Isso pode permitir que o codificador de vídeo acesse informações de vetor de movimento com uma chamada de memória (por exemplo, com uma requisição em lote) em vez de requerer múltiplas chamadas de memória. O codificador de vídeo pode, então, dispor as informações de vetor de movimento de acordo com uma ordem específica, tal como as ordens exemplificativas, como descrito acima.

[0038] Em algumas técnicas, se informações de vetor de movimento de um bloco espacialmente vizinho específico estão incluídas ou não na lista de preditores de vetor de movimento baseia-se na disponibilidade de informações de vetor de movimento de outro bloco. Dito de outra forma, se um bloco espacialmente vizinho específico deve ser adicionado na lista de preditores de vetor de movimento é condicional (por exemplo, condicionado com base na disponibilidade de informações de vetor de movimento de outro bloco). Devido a haver uma condição quanto a se as informações de vetor de movimento para a vizinhança espacial estão incluídas ou não, o codificador de vídeo pode precisar primeiramente acessar as informações de vetor de movimento do outro bloco para determinar se a condição é satisfeita ou não, e, em seguida, com base em se a condição é satisfeita ou não, enviar outra requisição para as informações de vetor de movimento do bloco espacialmente vizinho. Tais técnicas podem resultar em múltiplas chamadas de memória, que podem aumentar o tempo necessário para codificar um bloco.

[0039] Em um ou mais exemplos descritos nesta divulgação, se informações de vetor de movimento de um bloco espacialmente vizinho específico devem ser incluídas em uma lista de preditores de vetor de movimento não é condicional a se informações de vetor de movimento de outro bloco estão disponíveis ou não. Dessa forma, o codificador de vídeo pode ser capaz de acessar as informações de vetor de movimento dos blocos espacialmente vizinhos com uma requisição em lote à memória em vez de múltiplas

requisições, o que pode reduzir o tempo necessário para codificar um bloco e melhorar a operação do codificador de vídeo.

[0040] Como descrito acima, além das informações de vetor de movimento de blocos espacialmente vizinhos, o codificador de vídeo pode incluir informações de vetor de movimento de um ou mais blocos colocalizados (também chamados de blocos temporalmente vizinhos). Os um ou mais blocos colocalizados são blocos em outra imagem (denominada uma imagem de referência) que não a imagem que inclui o bloco sendo codificado. Um exemplo dos blocos colocalizados é um bloco localizado em uma posição na imagem de referência que sobrepõe uma posição do bloco atual sendo codificado na imagem atual. Dito de outra forma, o bloco atual pode definir uma área dentro da imagem atual, e um bloco que está dentro da mesma área, mas na imagem de referência, é um exemplo de um bloco colocalizado. Esse exemplo de um bloco colocalizado podem ser referido como um bloco colocalizado central.

[0041] Outro exemplo de um bloco colocalizado é um bloco que está localizado abaixo do bloco atual, mas na imagem de referência (por exemplo, abaixo da área definida pelo bloco atual na imagem atual, mas na imagem de referência). Esse exemplo de um bloco colocalizado pode ser referido como um bloco colocalizado inferior. Outro exemplo de um bloco colocalizado é um bloco que é localizado à direita do bloco atual, mas na imagem de referência (por exemplo, direita da área definida pelo bloco atual na imagem atual, mas na imagem de referência). Esse exemplo de

um bloco colocalizado pode ser referido como um bloco colocalizado esquerdo. Outro exemplo de um bloco colocalizado é um bloco que é localizado à esquerda do bloco atual, mas na imagem de referência (por exemplo, à esquerda da área definida pelo bloco atual na imagem atual, mas na imagem de referência). Esse exemplo de um bloco colocalizado pode ser referido como um bloco colocalizado esquerdo.

[0042] Em um ou mais exemplos, o codificador de vídeo pode determinar se informações de vetor de movimento para o bloco colocalizado central estão disponíveis. Se o vetor de movimento para o bloco colocalizado central está disponível, o codificador de vídeo pode adicionar as informações de vetor de movimento à lista de preditores de vetor de movimento, e não verificar quaisquer outros blocos colocalizados. Se o vetor de movimento para o bloco colocalizado central não está disponível, o codificador de vídeo pode determinar se informações de vetor de movimento para o bloco colocalizado abaixo estão disponíveis. Se o vetor de movimento para o bloco colocalizado abaixo está disponível, o codificador de vídeo pode adicionar as informações de vetor de movimento à lista de preditores de vetor de movimento, e não verificar quaisquer outros blocos colocalizados.

[0043] Se o vetor de movimento para o bloco colocalizado abaixo não está disponível, o codificador de vídeo pode determinar se informações de vetor de movimento para um do bloco colocalizado direito ou o bloco colocalizado esquerdo estão disponíveis com base na ordem

de codificação de imagem. Como um exemplo, se a ordem de codificação é da esquerda para a direita, então, o codificador de vídeo pode determinar se informações de vetor de movimento para o bloco colocalizado direito estão disponíveis. Se a ordem de codificação é da direita para a esquerda, então, o codificador de vídeo pode determinar se informações de vetor de movimento para o bloco colocalizado esquerdo estão disponíveis. Em outros exemplos, o codificador de vídeo pode realizar o inverso (por exemplo, informações de vetor de movimento para o bloco colocalizado esquerdo para ordem de codificação da esquerda para a direita e informações de vetor de movimento para o bloco colocalizado direito para ordem de codificação da direita para a esquerda). Se o vetor de movimento para o bloco colocalizado esquerdo ou direito, conforme aplicável, estão disponíveis, o codificador de vídeo pode adicionar as informações de vetor de movimento à lista de preditores de vetor de movimento, e não verificar quaisquer outros blocos colocalizados.

[0044] Pode haver exemplos adicionais de blocos colocalizados. Por exemplo, um bloco colocalizado pode ser bloco inferior-direito que é posicionado na parte inferior e à direita (por exemplo, diagonalmente na direção inferior-direita) do bloco atual, mas na imagem de referência (por exemplo, inferior-direita da área definida pelo bloco atual na imagem atual, mas na imagem de referência). Outro exemplo de um bloco colocalizado pode ser bloco inferior-esquerdo que é posicionado na parte inferior e à esquerda (por exemplo, diagonalmente na

direção inferior-esquerda) do bloco atual, mas na imagem de referência (por exemplo, inferior-esquerda da área definida pelo bloco atual na imagem atual, mas na imagem de referência). Se o codificador de vídeo verifica o bloco colocalizado inferior-esquerdo ou inferior-direito para informações de vetor de movimento pode ser com base na ordem de codificação de imagem (por exemplo, bloco colocalizado inferior-direito para ordem de codificação da esquerda para a direita e bloco colocalizado inferior-esquerdo para ordem de codificação da direita para a esquerda, ou vice-versa).

[0045] Além disso, o codificador de vídeo pode adicionar informações de vetor de movimento à lista de preditores de vetor de movimento até a lista de preditores de vetor de movimento estar completa (por exemplo, até o número de entradas na lista de preditores de vetor de movimento ser igual ao tamanho máximo da lista de preditores de vetor de movimento). Em alguns exemplos, o tamanho da lista de preditores de vetor de movimento pode ser com base no tamanho do bloco. Por exemplo, se um bloco tem um tamanho maior do que um tamanho limite, então, o tamanho máximo da lista de preditores de vetor de movimento pode ser M (por exemplo, pode haver um máximo de M número de entradas na lista de preditores de vetor de movimento). Se um bloco tem um tamanho inferior ao tamanho limite, então, o tamanho máximo da lista de preditores de vetor de movimento pode ser X (por exemplo, pode haver um máximo de X número de entradas na lista de preditores de vetor de movimento). X pode ser inferior a M .

[0046] Como um exemplo, se um bloco tem um tamanho maior do que $N \times 4$ ou $4 \times N$, então, o tamanho máximo da lista de preditores de vetor de movimento pode ser seis (por exemplo, $M = 6$). Se um bloco tem um tamanho inferior a $N \times 4$ ou $4 \times N$, então, o tamanho máximo da lista de preditores de vetor de movimento pode ser quatro (por exemplo, $X = 4$). Quando o tamanho limite é $N \times 4$ ou $4 \times N$, N pode ser menor do que ou igual a oito.

[0047] A Figura 1 é um diagrama de blocos ilustrando um sistema de codificação e decodificação de vídeo exemplificativo 100 que pode executar as técnicas desta divulgação. As técnicas desta divulgação são, de forma geral, dirigidas a codificar (codificação e/ou decodificação) de dados de vídeo. Em geral, dados de vídeo incluem quaisquer dados para processar um vídeo. Dessa forma, dados de vídeo podem incluir vídeo bruto, não codificado, vídeo codificado, vídeo decodificado (por exemplo, reconstruído), e metadados de vídeo, tal como dados de sinalização.

[0048] Como mostrado na Figura 1, o sistema 100 inclui um dispositivo de origem 102 que fornece dados de vídeo codificados a serem decodificados e exibidos pelo dispositivo de destino 116, neste exemplo. Em particular, o dispositivo de origem 102 fornece os dados de vídeo ao dispositivo de destino 116 através de um meio legível por computador 110. O dispositivo de origem 102 e o dispositivo de destino 116 podem compreender qualquer um de uma ampla variedade de dispositivos, incluindo computadores de mesa, notebooks (ou seja, laptops), tablets, conversores (set-top

boxes), aparelhos de telefone, tais como os chamados "smartphones", televisores, câmeras, dispositivos de exibição, reprodutores de mídia digital, consoles de videogame, dispositivo de fluxo contínuo de vídeo, ou semelhantes. Em alguns casos, o dispositivo de origem 102 e o dispositivo de destino 116 podem estar equipados para comunicação sem fio e, dessa forma, podem ser referidos como dispositivos de comunicação sem fio.

[0049] No exemplo da Figura 1, o dispositivo de origem 102 inclui fonte de vídeo 104, memória 106, codificador de vídeo 200 e interface de saída 108. O dispositivo de destino 116 inclui interface de entrada 122, decodificador de vídeo 300, memória 120 e dispositivo de exibição 118. De acordo com esta divulgação, o codificador de vídeo 200 do dispositivo de origem 102 e o decodificador de vídeo 300 do dispositivo de destino 116 podem ser configurados para aplicar as técnicas para predição de vetor de movimento espaço-temporal, tal como construindo a lista de preditores de vetor de movimento usando uma ou mais das técnicas exemplificativas descritas nesta divulgação. Dessa forma, o dispositivo de origem 102 representa um exemplo de um dispositivo de codificação de vídeo, enquanto o dispositivo de destino 116 representa um exemplo de um dispositivo de decodificação de vídeo. Em outros exemplos, um dispositivo de origem e um dispositivo de destino podem incluir outros componentes ou arranjos. Por exemplo, o dispositivo de origem 102 pode receber dados de vídeo de uma fonte de vídeo externa, tal como uma câmera externa. Da mesma forma, o dispositivo de destino 116 pode

fazer interface com um dispositivo de exibição externo, em vez de incluir um dispositivo de exibição integrado.

[0050] O sistema 100, conforme mostrado na Figura 1, é meramente um exemplo. Em geral, qualquer dispositivo de codificação e/ou decodificação de vídeo digital pode realizar técnicas para predição de vetor de movimento espaço-temporal. O dispositivo de origem 102 e o dispositivo de destino 116 são meramente exemplos de tais dispositivos de codificação, em que o dispositivo de origem 102 gera dados de vídeo codificados para transmissão ao dispositivo de destino 116. Esta divulgação refere-se a um dispositivo de "codificação" como um dispositivo que realiza codificação (codificação e/ou decodificação) de dados. Assim, o codificador de vídeo 200 e o decodificador de vídeo 300 representam exemplos de dispositivos de codificação, em particular, um codificador de vídeo e um decodificador de vídeo, respectivamente. Em alguns exemplos, os dispositivos 102, 116 podem funcionar de forma substancialmente simétrica, tal que cada um dos dispositivos 102, 116 inclui componentes de codificação e decodificação de vídeo. Assim, o sistema 100 pode ser suportar transmissão de vídeo unidirecional ou bidirecional entre dispositivos de vídeo 102, 116, por exemplo, para fluxo contínuo de vídeo, reprodução de vídeo, radiodifusão de vídeo ou videotelefonia.

[0051] Em geral, a fonte de vídeo 104 representa uma fonte de dados de vídeo (ou seja, dados de vídeo brutos, não codificados) e oferece uma série sequencial de imagens (também referida como "quadros") dos dados de vídeo

para o codificador de vídeo 200, que codifica dados para as imagens. A fonte de vídeo 104 do dispositivo de origem 102 pode incluir um dispositivo de captura de vídeo, tal como uma câmera de vídeo, um arquivo de vídeo contendo vídeo bruto previamente capturado, e/ou uma interface de alimentação de vídeo para receber vídeo de um provedor de conteúdo de vídeo. Como adicional, a fonte de vídeo 104 pode gerar dados baseados em gráficos de computador como a fonte de vídeo, ou uma combinação de vídeo ao vivo, vídeo arquivado e vídeo gerado por computador. Em cada caso, o codificador de vídeo 200 codifica os dados de vídeo capturados, pré-capturados ou gerados por computador. O codificador de vídeo 200 pode reorganizar as imagens da ordem recebida (às vezes chamada de "ordem de exibição") em uma ordem de codificação para codificar. O codificador de vídeo 200 pode gerar um fluxo de bits incluindo dados de vídeo codificados. O dispositivo de origem 102 pode, então, emitir os dados de vídeo codificados através da interface de saída 108 no meio legível por computador 110 para recepção e/ou recuperação, por exemplo, pela interface de entrada 122 do dispositivo de destino 116.

[0052] A memória 106 do dispositivo de origem 102 e a memória 120 do dispositivo de destino 116 representam memórias de uso geral. Em algum exemplo, as memórias 106, 120 podem armazenar dados de vídeo brutos, por exemplo, vídeo bruto da fonte de vídeo 104 e dados de vídeo brutos, e decodificados do decodificador de vídeo 300. Adicionalmente ou alternativamente, as memórias 106, 120 podem armazenar instruções de software executáveis, por

exemplo, pelo codificador de vídeo 200 e decodificador de vídeo 300, respectivamente. Embora mostradas separadamente do codificador de vídeo 200 e decodificador de vídeo 300 neste exemplo, deve-se entender que o codificador de vídeo 200 e o decodificador de vídeo 300 podem também incluir memórias internas para fins funcionalmente semelhantes ou equivalentes. Além disso, as memórias 106, 120 podem armazenar dados de vídeo codificados, por exemplo, saída do codificador de vídeo 200 e entrada para o decodificador de vídeo 300. Em alguns exemplos, porções de memórias 106, 120 podem ser alocadas como um ou mais buffers de vídeo, por exemplo, para armazenar dados de vídeo brutos, decodificados e/ou codificados.

[0053] O meio legível por computador 110 pode representar qualquer tipo de meio ou dispositivo capaz de transportar os dados de vídeo codificados do dispositivo de origem 102 para o dispositivo de destino 116. Em um exemplo, o meio legível por computador 110 representa um meio de comunicação para permitir que o dispositivo de origem 102 transmita dados de vídeo codificados diretamente ao dispositivo de destino 116 em tempo real, por exemplo, através de uma rede de radiofrequência ou rede baseada em computador. A interface de saída 108 pode modular um sinal de transmissão incluindo os dados de vídeo codificados, e a interface de entrada 122 pode modular o sinal de transmissão recebido, de acordo com um padrão de comunicação, tal como um protocolo de comunicação sem fio. O meio de comunicação pode compreender qualquer meio de comunicação sem fio ou a cabo, tal como espectro de

radiofrequência (RF) ou uma ou mais linhas de transmissão físicas. O meio de comunicação pode fazer parte de uma rede baseada em pacotes, tal como uma rede de área local, uma rede de área ampla ou uma rede global como a Internet. O meio de comunicação pode incluir roteadores, comutadores, estações base ou qualquer outro equipamento que possa ser útil para facilitar a comunicação do dispositivo de origem 102 com o dispositivo de destino 116.

[0054] Em alguns exemplos, o dispositivo de origem 102 pode gerar dados codificados da interface de saída 108 para o dispositivo de armazenamento 112. Da mesma forma, o dispositivo de destino 116 pode acessar dados codificados do dispositivo de armazenamento 112 por meio da interface de entrada 122. O dispositivo de armazenamento 112 pode incluir qualquer um de uma variedade de meios de armazenamento de dados distribuídos ou localmente acessados, tais como disco rígido, discos Blu-ray, DVDs, CD-ROMs, memória flash, memória volátil ou não volátil, ou quaisquer outros meios de armazenamento digital adequados para armazenar dados de vídeo codificados.

[0055] Em alguns exemplos, o dispositivo de origem 102 pode gerar dados de vídeo codificados para o servidor de arquivos 114 ou outro dispositivo de armazenamento intermediário que pode armazenar o vídeo codificado gerado pelo dispositivo de origem 102. O dispositivo de destino 116 pode acessar dados de vídeo armazenados do servidor de arquivos 114 via fluxo contínuo ou *download*. O servidor de arquivos 114 pode ser qualquer tipo de dispositivo de servidor capaz de armazenar dados de

vídeo codificados e transmitir aqueles dados de vídeo codificados ao dispositivo de destino 116. O servidor de arquivos 114 pode representar um servidor web (por exemplo, para um site), um Servidor de Protocolo de Transferência de Arquivos (FTP), um dispositivo de rede de distribuição de conteúdo ou um dispositivo de armazenamento conectado à rede (NAS). O dispositivo de destino 116 pode acessar dados de vídeo codificados do servidor de arquivos 114 através de qualquer conexão de dados padrão, incluindo uma conexão de Internet. Isso pode incluir um canal sem fio (por exemplo, uma conexão Wi-Fi), uma conexão a cabo (por exemplo, DSL, modem a cabo etc.), ou uma combinação de ambos, que é adequada para acessar dados de vídeo codificados armazenados no servidor de arquivos 114. O servidor de arquivos 114 e a interface de entrada 122 podem ser configurados para operar de acordo com um protocolo de transmissão de fluxo contínuo, um protocolo de transmissão de *download* ou uma combinação dos mesmos.

[0056] A Interface de saída 108 e a interface de entrada 122 podem representar transmissores/receptores sem fio, modems, componentes de rede a cabo (por exemplo, placas Ethernet), componentes de comunicação sem fio que operam de acordo com qualquer um de uma variedade de padrões IEEE 802.11, ou outros componentes físicos. Em exemplos em que a interface de saída 108 e a interface de entrada 122 compreendem componentes sem fio, a interface de saída 108 e a interface de entrada 122 podem ser configuradas para transferir de dados, tais como dados de vídeo codificados, de acordo com um padrão de comunicação

celular, tal como 4G, 4G-LTE (Evolução de Longo Prazo), LTE Avançado, 5G ou similares. Em alguns exemplos em que a interface de saída 108 compreende um transmissor sem fio, a interface de saída 108 e a interface de entrada 122 podem ser configuradas para transferir dados, tais como dados de vídeo codificados, de acordo com outros padrões sem fio, tais como uma especificação IEEE 802.11, uma especificação IEEE 802.15 (por exemplo, ZigBee™), um padrão Bluetooth™, ou semelhantes. Em alguns exemplos, o dispositivo de origem 102 e/ou o dispositivo destino 116 podem incluir respectivos dispositivos de sistema-em-chip (SoC). Por exemplo, o dispositivo de origem 102 pode incluir um dispositivo SoC para realizar a funcionalidade atribuída ao codificador de vídeo 200 e/ou interface de saída 108, e o dispositivo de destino 116 pode incluir um dispositivo SoC para realizar a funcionalidade atribuída ao decodificador de vídeo 300 e/ou interface de entrada 122.

[0057] As técnicas desta divulgação podem ser aplicadas à codificação de vídeo em suporte a qualquer uma de uma variedade de aplicações multimídia, tais como radiodifusões de televisão pelo ar, transmissões de televisão a cabo, transmissões de televisão por satélite, transmissões de vídeo em fluxo contínuo da Internet, tais como fluxo contínuo adaptativo dinâmico sobre HTTP (DASH), vídeo digital que é codificado em um meio de armazenamento de dados, decodificação de vídeo digital armazenado em um meio de armazenamento de dados, ou outras aplicações.

[0058] A interface de entrada 122 do dispositivo de destino 116 recebe um fluxo de bits de vídeo codificado

do meio legível por computador 110 (por exemplo, dispositivo de armazenamento 112, servidor de arquivos 114, ou semelhantes). O meio legível por computador de fluxo de bits de vídeo codificado 110 pode incluir sinalizar informações definidas pelo codificador de vídeo 200, que também é usado pelo decodificador de vídeo 300, tal como elementos de sintaxe tendo valores que descrevem características e/ou processamento de blocos de vídeo ou outras unidades codificadas (por exemplo, fatias, imagens, grupos de imagens, sequências ou semelhantes). O dispositivo de exibição 118 exibe imagens decodificadas dos dados de vídeo decodificados para um usuário. O dispositivo de exibição 118 pode representar qualquer um de uma variedade de dispositivos de exibição, tais como um tubo de raios catódicos (CRT), um visor de cristal líquido (LCD), um visor de plasma, um visor de diodo orgânico emissor de luz (OLED), ou outro tipo de dispositivo de exibição.

[0059] Embora não mostrado na Figura 1, em alguns exemplos, o codificador de vídeo 200 e o decodificador de vídeo 300 podem, cada um, ser integrados com um codificador de áudio e/ou decodificador de áudio, e podem incluir unidades MUX-DEMUX apropriadas, ou outro hardware e/ou software, para lidar com fluxos multiplexados incluindo áudio e vídeo em um fluxo de dados comum. Se aplicável, as unidades MUX-DEMUX podem estar em conformidade com o protocolo multiplexador ITU H.223, ou outros protocolos, tais como o protocolo de datagrama de usuário (UDP).

[0060] O codificador de vídeo 200 e decodificador de vídeo 300 podem, cada um, ser implementados como

qualquer um de uma variedade de circuitos codificadores e/ou decodificadores adequados, tais como um ou mais microprocessadores, processadores de sinal digital (DSPs), circuitos integrados de aplicação específica (ASICs), matrizes de portas programáveis em campo (FPGAs), lógica discreta, software, hardware, firmware ou qualquer combinação dos mesmos. Quando as técnicas são implementadas parcialmente em software, um dispositivo pode armazenar instruções para o software em um meio legível por computador adequado e não transitório, e executar as instruções em hardware usando um ou mais processadores para realizar as técnicas desta divulgação. Cada o codificador de vídeo 200 e o decodificador de vídeo 300 podem ser incluído em um ou mais codificadores ou decodificadores, podendo ser integrados como parte de um codificador/decodificador combinado (CODEC) em um respectivo dispositivo. Um dispositivo incluindo o codificador de vídeo 200 e/ou decodificador de vídeo 300 pode compreender um circuito integrado, um microprocessador e/ou um dispositivo de comunicação sem fio, tal como um telefone celular.

[0061] Padrões de codificação de vídeo incluem ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 ou ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (também conhecido como ISO/IEC MPEG-4 AVC), incluindo suas extensões de Codificação de Vídeo Escalável (SVC) e Codificação de Vídeo Multivista (MVC).

[0062] Além disso, um padrão de codificação de vídeo, denominado Codificação de Vídeo de Alta Eficiência

(HEVC) ou ITU-T H.265 (conforme descrito em G.J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand "Overview of the High Efficiency Video Coding (HEVC)", IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, n°. 12. pp. 1649-1668, dezembro de 2012), incluindo sua extensão de alcance, extensão multivista (MV-HEVC) e extensão escalável (SHVC), foi desenvolvido pela Equipe de Colaboração Conjunta em Codificação de Vídeo (JCT-VC), bem como pela Equipe de Colaboração Conjunta em Desenvolvimento de Extensão de Codificação de Vídeo 3D (JCT-3V) do Grupo de Especialistas em Codificação de Vídeo ITU-T (VCEG) e Grupo de Especialistas de Imagem em Movimento ISO/IEC (MPEG). Um projeto de especificação HEVC, referido como HEVC WD doravante, está disponível em http://phenix.int-evry.fr/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1003-v1.zip. A última versão do Projeto Final de Padrão Internacional (FDIS) de HEVC pode ser encontrada em http://phenix.it-sudparis.eu/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip.

[0063] ITU-T VCEG (Q6/16) e ISO/IEC MPEG (JTC 1/SC 29/WG11) estão agora estudando a necessidade potencial de padronização da tecnologia futura de codificação de vídeo com uma capacidade de compressão que excede significativamente o padrão HEVC completo (incluindo suas extensões atuais e extensões de curto prazo para codificação de conteúdo de tela e codificação de alto alcance dinâmico). Os grupos estão trabalhando juntos nessa atividade de exploração em um esforço conjunto de

colaboração conhecido como Equipe de Exploração de Vídeo Conjunta (JVET) para avaliar projetos de tecnologia de compressão propostos por seus especialistas nesta área. O JVET se reuniu pela primeira vez entre 19 e 21 de outubro de 2015. Uma versão do software de referência, ou seja, Modelo de Teste de Exploração Conjunta 7 (JEM-7), pode ser baixada em: https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/HM-16.6-JEM-7.2/. Uma descrição do algoritmo do Modelo de Teste de Exploração Conjunta 7 (JEM-7) é descrito em J. Chen, E. Alshina, G. J. Sullivan, J.-R. Ohm, J. Boyce, "Algorithm Description of Joint Exploration Test Model 7", JVET-G1001, julho de 2017. Um rascunho recente do padrão VVC é descrito em Bross, et al. "Versatile Video Coding (Draft 6)", Equipe de Especialistas em Vídeo Conjunta (JVET) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 15º Encontro: Gotemburgo, SE, 3-12 de julho de 2019, JVET-O2001-vE (doravante "Rascunho VVC 6"). As técnicas desta divulgação, no entanto, não são limitadas a nenhum padrão de codificação específico.

[0064] O codificador de vídeo 200 e o decodificador de vídeo 300 podem operar de acordo com um padrão de codificação de vídeo, tal como ITU-T H.265, também referido como Codificação de Vídeo de Alta Eficiência (HEVC) ou extensões relacionadas, tais como as extensões de codificação de vídeo escalável e/ou multivista. Alternativamente, o codificador de vídeo 200 e o decodificador de vídeo 300 podem operar de acordo com outros padrões proprietários ou industriais, tais como o

Modelo de Teste de Exploração Conjunta (JEM) e/ou VVC. Outro exemplo de um padrão de codificação de vídeo é o padrão de codificação de vídeo essencial (EVC). As técnicas desta divulgação, no entanto, não são limitadas a nenhum padrão de codificação específico.

[0065] Em geral, o codificador de vídeo 200 e o decodificador de vídeo 300 podem realizar codificação de imagens baseada em bloco. O termo "bloco" refere-se, de forma geral, a uma estrutura incluindo dados a serem processados (por exemplo, codificados, decodificados, ou de outra forma usados no processo de codificação e/ou decodificação). Por exemplo, um bloco pode incluir uma matriz bidimensional de amostras de dados de luminância e/ou crominância. Em geral, o codificador de vídeo 200 e o decodificador de vídeo 300 podem codificar dados de vídeo representados no formato YUV (por exemplo, Y, Cb, Cr). Ou seja, em vez de codificar dados em vermelho, verde e azul (RGB) para amostras de uma imagem, o codificador de vídeo 200 e o decodificador de vídeo 300 podem codificar componentes de luminância e crominância, em que os componentes de crominância podem incluir tanto componentes de crominância matiz vermelho quanto de matiz azul. Em alguns exemplos, o codificador de vídeo 200 converte dados formatados RGB recebidos em uma representação YUV antes da codificação, e o decodificador de vídeo 300 converte a representação YUV em formato RGB. Alternativamente, unidades de pré- e pós-processamento (não mostradas) podem realizar essas conversões.

[0066] Esta divulgação pode se referir, de forma

geral, à codificação (por exemplo, codificação e decodificação) de imagens para incluir o processo de codificar ou decodificar dados da imagem. Da mesma forma, esta divulgação pode se referir à codificação de blocos de uma imagem para incluir o processo de codificação ou decodificação de dados para os blocos, por exemplo, codificação residual e/ou predição. Um fluxo de bits de vídeo codificado inclui, de forma geral, uma série de valores para elementos de sintaxe representativos de decisões de codificação (por exemplo, modos de codificação) e particionamento de imagens em blocos. Assim, as referências à codificação de uma imagem ou um bloco devem, de forma geral, ser entendidas como valores de codificação para elementos de sintaxe que formam a imagem ou bloco.

[0067] HEVC define vários blocos, incluindo unidades de codificação (CUs), unidades de predição (PUs) e unidades de transformação (TUs). De acordo com HEVC, um codificador de vídeo (tal como codificador de vídeo 200) particiona uma unidade de árvore de codificação (CTU) em CUs de acordo com uma estrutura árvore quádrupla. Ou seja, o codificador de vídeo particiona CTUs e CUs em quatro quadrados iguais, não sobrepostos, e cada nó do árvore quádrupla tem zero ou quatro nós filhos. Os nós sem nós filhos podem ser referidos como "nós folhas", e as UCs de tais nós folhas podem incluir uma ou mais PUs e/ou uma ou mais TUs. O codificador de vídeo pode ainda particionar PUs e TUs. Por exemplo, em HEVC, uma árvore quádrupla residual (RQT) representa particionamento de TUs. Em HEVC, as PUs representam dados de interpredição, enquanto as TUs

representam dados residuais. As CUs que são intrapreditas incluem informações de intrapredição, tais como uma indicação intramodo.

[0068] Como outro exemplo, o codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para operar de acordo com exemplos de JEM e/ou VVC. De acordo com exemplos de JEM/VVC, um codificador de vídeo (tal como codificador de vídeo 200) particiona uma imagem em uma pluralidade de unidades de árvore de codificação (CTUs). O codificador de vídeo 200 pode particionar uma CTU de acordo com uma estrutura de árvore, tal como uma estrutura de árvore quádrupla-binária (QTBT). A estrutura QTBT dos exemplos de JEM/VVC elimina os conceitos de vários tipos de partição, tal como a separação entre CUs, PUs e TUs de HEVC. Uma estrutura QTBT dos exemplos de JEM/VVC inclui dois níveis: um primeiro nível particionado de acordo com a partição de árvore quádrupla, e um segundo nível particionado de acordo com a partição de árvore binária. Um nó raiz da estrutura QTBT corresponde a uma CTU. Os nós folhas das árvores binárias correspondem a unidades de codificação (CUs).

[0069] Em alguns exemplos, o codificador de vídeo 200 e o decodificador de vídeo 300 podem usar uma única estrutura QTBT para representar cada um dos componentes de luminância e crominância, enquanto em outros exemplos, o codificador de vídeo 200 e o decodificador de vídeo 300 podem usar duas ou mais estruturas QTBT, tais como uma estrutura QTBT para o componente de luminância e outra estrutura QTBT para ambos os componentes de crominância (ou

duas estruturas QTBT para os respectivos componentes de crominância).

[0070] O codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para usar particionamento de árvore quádrupla por HEVC, particionamento de QTBT de acordo com exemplos de JEM/VVC, ou outras estruturas de particionamento. Para fins de explicação, a descrição das técnicas desta divulgação é apresentada com respeito ao particionamento de QTBT. No entanto, deve-se entender que as técnicas desta divulgação podem também ser aplicadas a codificadores de vídeo configurados para usar particionamento de árvore quádrupla, ou outros tipos de particionamento.

[0071] Esta divulgação pode usar "NxN" e "N por N" alternadamente para se referir às dimensões de amostra de um bloco (tal como uma CU ou outro bloco de vídeo) em termos de dimensões verticais e horizontais, por exemplo, amostras 16x16 ou amostras 16 por 16. Em geral, uma CU 16x16 terá 16 amostras em direção vertical ($y = 16$) e 16 amostras em uma direção horizontal ($x = 16$). Da mesma forma, uma CU NxN de forma geral tem N amostras em uma direção vertical e N amostras em uma direção horizontal, em que N representa um valor inteiro não negativo. As amostras em uma CU podem ser organizadas em linhas e colunas. Além disso, as UCs não precisam necessariamente ter o mesmo número de amostras na direção horizontal que na direção vertical. Por exemplo, as CUs podem compreender NxM amostras, em que M não é necessariamente igual a N.

[0072] O codificador de vídeo 200 codifica dados

de vídeo para CUs representando informações de predição e/ou residuais, e outras informações. As informações de predição indicam como a CU deve ser predita para formar um bloco de predição para a CU. As informações residuais geralmente representam diferenças de amostra por amostra entre amostras da CU antes de codificar e do bloco de predição.

[0073] Para predizer uma CU, o codificador de vídeo 200 pode de forma geral formar um bloco de predição para a CU através de interpredição ou intrapredição. A interpredição se refere de forma geral à predição da CU a partir de dados de imagem de uma imagem previamente codificada, enquanto a intrapredição se refere de forma geral à predição da CU a partir de dados previamente codificados da mesma imagem. Para realizar a interpredição, o codificador de vídeo 200 pode gerar o bloco de predição usando um ou mais vetores de movimento. O codificador de vídeo 200 pode de forma geral realizar uma busca de movimento para identificar um bloco de referência que se aproxime da CU, por exemplo, em termos de diferenças entre a CU e o bloco de referência. O codificador de vídeo 200 pode calcular uma métrica de diferença usando uma soma da diferença absoluta (SAD), soma de diferenças quadradas (SSD), diferença média absoluta (MAD), diferenças médias quadradas (MSD), ou outros cálculos de diferença para determinar se um bloco de referência se aproxima da CU atual. Em alguns exemplos, o codificador de vídeo 200 pode predizer a CU atual usando predição unidirecional ou predição bidirecional.

[0074] JEM ou VVC também fornece um modo de compensação de movimento afim, que pode ser considerado um modo de interpredição. No modo de compensação de movimento afim, o codificador de vídeo 200 pode determinar dois ou mais vetores de movimento que representam movimento não translacional, tal como aproximação ou afastamento (zoom in/out), rotação, movimento em perspectiva, ou outros tipos de movimentos irregulares.

[0075] Para realizar a intrapredição, o codificador de vídeo 200 pode selecionar um modo de intrapredição para gerar o bloco de predição. JEM oferece sessenta e sete modos de intrapredição, incluindo vários modos direcionais, bem como modo planar e modo DC. Em geral, o codificador de vídeo 200 seleciona um modo de intrapredição que descreve amostras vizinhas a um bloco atual (por exemplo, um bloco de uma CU) a partir do qual predizer amostras do bloco atual. Tais amostras podem, de forma geral, estar acima, acima e à esquerda, ou à esquerda do bloco atual na mesma imagem que o bloco atual, assumindo que o codificador de vídeo 200 codifica CTUs e CUs em ordem de varredura raster (ordem de codificação da esquerda para a direita, de cima para baixo ou ordem de codificação da direita para a esquerda, de cima para baixo).

[0076] O codificador de vídeo 200 codifica dados que representam o modo de predição para um bloco atual. Por exemplo, para modos de interpredição, o codificador de vídeo 200 pode codificar dados que representam qual dos vários modos de interpredição disponíveis é usado, bem como as informações de movimento para o modo correspondente.

Para interpredição unidirecional ou bidirecional, por exemplo, o codificador de vídeo 200 pode codificar vetores de movimento usando predição avançada de vetor de movimento (AMVP) ou modo de mesclagem. O codificador de vídeo 200 pode usar modos semelhantes para codificar vetores de movimento para o modo de compensação de movimento afim.

[0077] Após a predição, tal como intrapredição ou interpredição de um bloco, o codificador de vídeo 200 pode calcular dados residuais para o bloco. Os dados residuais, tais como um bloco residual, representa diferenças amostra por amostra entre o bloco e um bloco de predição para o bloco, formado usando o modo de predição correspondente. O codificador de vídeo 200 pode aplicar uma ou mais transformadas ao bloco residual, para produzir dados transformados em um domínio de transformada em vez do domínio de amostra. Por exemplo, o codificador de vídeo 200 pode aplicar uma transformada discreta de cosseno (DCT), uma transformada de número inteiro, uma transformada wavelet, ou uma transformada conceitualmente semelhante a dados de vídeo residuais. Além disso, o codificador de vídeo 200 pode aplicar uma transformada secundária após a primeira transformada, tal como uma transformada secundária não separável dependente de modo (MDNSST), uma transformada dependente de sinal, uma transformada Karhunen-Loeve (KLT), ou semelhantes. O codificador de vídeo 200 produz coeficientes de transformada seguindo a aplicação das uma ou mais transformadas.

[0078] Como observado acima, após quaisquer transformadas para produzir coeficientes de transformada, o

codificador de vídeo 200 pode realizar quantização dos coeficientes de transformada. A quantização se refere de forma geral a um processo em que coeficientes de transformada são quantizados para possivelmente reduzir a quantidade de dados usados para representar os coeficientes, proporcionando posterior compressão. Ao realizar o processo de quantização, o codificador de vídeo 200 pode reduzir a profundidade de bits associada com alguns ou todos os coeficientes. Por exemplo, o codificador de vídeo 200 pode arredondar um valor de n-bit para um valor de m-bit durante a quantização, em que n é maior do que m. Em alguns exemplos, para realizar a quantização, o codificador de vídeo 200 pode realizar um deslocamento à direita bit a bit do valor a ser quantizado.

[0079] Após a quantização, o codificador de vídeo 200 pode fazer a varredura dos coeficientes de transformada, produzindo um vetor unidimensional a partir da matriz bidimensional incluindo os coeficientes de transformada quantizados. A varredura pode ser projetada para colocar coeficientes de energia mais alta (e, portanto, menor frequência) na frente do vetor e colocar coeficientes de transformada de energia mais baixa (e, portanto, maior frequência) na parte de trás do vetor. Em alguns exemplos, o codificador de vídeo 200 pode utilizar uma ordem de varredura predefinida para fazer a varredura dos coeficientes de transformada quantizados para produzir um vetor serializado, e, então, codificar por entropia os coeficientes de transformada quantizados do vetor. Em outros exemplos, o codificador de vídeo 200 pode realizar

uma varredura adaptativa. Após fazer a varredura dos coeficientes de transformada quantizados para formar o vetor unidimensional, o codificador de vídeo 200 pode codificar por entropia o vetor unidimensional, por exemplo, de acordo com codificador aritmético binário adaptativo ao contexto (CABAC). O codificador de vídeo 200 pode também codificar por entropia valores para elementos de sintaxe que descrevem metadados associados com os dados de vídeo codificados para uso pelo decodificador de vídeo 300 na decodificação de dados de vídeo.

[0080] Para realizar CABAC, o codificador de vídeo 200 pode atribuir um contexto dentro de um modelo de contexto a um símbolo a ser transmitido. O contexto pode se referir a, por exemplo, se os valores vizinhos do símbolo são com valor zero ou não. A determinação de probabilidade pode ser baseada em um contexto atribuído ao símbolo.

[0081] O codificador de vídeo 200 pode gerar ainda dados de sintaxe, tais como dados de sintaxe baseada em bloco, dados de sintaxe baseada em imagem, dados de sintaxe baseada em sequência, para o decodificador de vídeo 300, por exemplo, em um cabeçalho de imagem, um cabeçalho de bloco, um cabeçalho de fatia ou outros dados de sintaxe, tais como um conjunto de parâmetros de sequência (SPS), um conjunto de parâmetros de imagem (PPS), ou conjunto de parâmetros de vídeo (VPS). O decodificador de vídeo 300 pode, da mesma forma, decodificar esses dados de sintaxe para determinar como decodificar os dados de vídeo correspondentes.

[0082] Dessa forma, o codificador de vídeo 200

pode gerar um fluxo de bits incluindo dados de vídeo codificados, por exemplo, elementos de sintaxe que descrevem o particionamento de uma imagem em blocos (por exemplo, CUs) e informações de predição e/ou residuais para os blocos. Por fim, o decodificador de vídeo 300 pode receber o fluxo de bits e decodificar os dados de vídeo codificados.

[0083] Em geral, o decodificador de vídeo 300 realiza um processo recíproco ao executado pelo codificador de vídeo 200 para decodificar os dados de vídeo codificados do fluxo de bits. Por exemplo, o decodificador de vídeo 300 pode decodificar valores para elementos de sintaxe do fluxo de bits usando CABAC de maneira substancialmente semelhante a, embora recíproco a, o processo de codificação CABAC do codificador de vídeo 200. Os elementos de sintaxe podem definir informações de particionamento de uma imagem em CTUs, e particionamento de cada CTU de acordo com uma estrutura de partição correspondente, tal como uma estrutura QTBT, para definir CUs da CTU. Os elementos de sintaxe podem ainda definir informações de predição e residuais para blocos (por exemplo, CUs) de dados de vídeo.

[0084] As informações residuais podem ser representadas, por exemplo, por coeficientes de transformada quantizados. O decodificador de vídeo 300 pode realizar quantização inversa e transformada inversa dos coeficientes de transformada quantizados de um bloco para reproduzir um bloco residual para o bloco. O decodificador de vídeo 300 usa um modo de predição sinalizada (intra- ou interpredição) e informações de predição relacionadas (por

exemplo, informações de movimento para interpredição) para formar um bloco de predição para o bloco. O decodificador de vídeo 300 pode, então, combinar o bloco de predição e o bloco residual (numa base amostra por amostra) para reproduzir o bloco original. O decodificador de vídeo 300 pode realizar processamento adicional, tal como realizar um processo de desbloqueio para reduzir artefatos visuais ao longo dos limites do bloco.

[0085] De acordo com as técnicas desta divulgação, o codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para construir uma lista de preditores de vetor de movimento. Como descrito acima, para modo de mesclagem e modo AMVP, o decodificador de vídeo 300 pode determinar um vetor de movimento para um bloco atual sendo decodificado com base em informações de vetor de movimento na lista de preditores de vetor de movimento. Por exemplo, o codificador de vídeo 200 pode sinalizar informações indicativas de um índice na lista de preditores de vetor de movimento, e o decodificador de vídeo 300 pode recuperar as informações de vetor de movimento armazenadas na lista de preditores de vetor de movimento com base no índice. Para o modo de mesclagem, o decodificador de vídeo 300 pode definir o vetor de movimento para o bloco atual igual às informações de vetor de movimento recuperadas. Para o modo AMVP, o decodificador de vídeo 300 pode ainda receber informações indicativas de uma diferença de vetor de movimento (MVD) entre o vetor de movimento para o bloco atual e as informações de vetor de movimento recuperadas. Nesse exemplo, o decodificador de

vídeo 300 pode adicionar o MVD às informações de vetor de movimento recuperadas para determinar o vetor de movimento para o bloco atual.

[0086] No modo de mesclagem e AMVP, o codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para construir a lista de preditores de vetor de movimento de maneira semelhante, tal que a lista de preditores de vetor de movimento construída pelo codificador de vídeo 200 e a lista de preditores de vetor de movimento construída pelo decodificador de vídeo 300 sejam substancialmente a mesma. Para construir a lista de preditores de vetor de movimento, o codificador de vídeo 200 e o decodificador de vídeo 300 podem recuperar informações de vetor de movimento de blocos espacialmente vizinhos e blocos temporalmente vizinhos (também chamados de blocos colocalizados). Blocos espacialmente vizinhos se referem a blocos localizados na imagem atual que inclui o bloco atual sendo codificado ou decodificado. Blocos colocalizados se referem a blocos localizados em uma imagem de referência que é diferente da imagem atual.

[0087] No entanto, quais blocos espacialmente vizinhos o codificador de vídeo 200 e o decodificador de vídeo 300 verificam para informações de vetor de movimento podem ser com base em uma ordem de codificação da imagem atual em que o bloco atual sendo codificado ou decodificado está localizado. Como descrito em mais detalhes abaixo, em exemplos em que a imagem atual tem uma ordem de codificação da esquerda para a direita, exemplos dos blocos espacialmente vizinhos que são verificados são ilustrados

na Figura 8A e identificados como bloco 0 802, bloco 1 804, bloco 2 806, bloco 3 808, e bloco 4 810. Em exemplos em que a imagem atual tem uma ordem de codificação da direita para a esquerda, exemplos dos blocos espacialmente vizinhos que são verificados são ilustrados na Figura 8B e identificados como bloco 0 818, bloco 1 820, bloco 2 822, bloco 3 824, e bloco 4 826. Dessa forma, o codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para verificar diferentes blocos espacialmente vizinhos com base na ordem de codificação da imagem.

[0088] Além de blocos espacialmente vizinhos, o codificador de vídeo 200 e o decodificador de vídeo 300 podem incluir informações de vetor de movimento de blocos colocalizados. Em alguns exemplos, para ordem de codificação da esquerda para a direita da imagem atual, o codificador de vídeo 200 e o decodificador de vídeo 300 podem determinar se informações de vetor de movimento do bloco colocalizado central 5 812 da Figura 8A estão disponíveis, e se disponíveis, incluem as informações de vetor de movimento do bloco colocalizado central 5 812 na lista de preditores de vetor de movimento e não verificam ou incluem informações de vetor de movimento de nenhum outro bloco colocalizado.

[0089] Se informações de vetor de movimento do bloco colocalizado central 5 812 não estão disponíveis, o codificador de vídeo 200 e o decodificador de vídeo 300 podem determinar se informações de vetor de movimento do bloco colocalizado inferior 6 814 da Figura 8A estão disponíveis, e, se disponíveis, incluem as informações de

vetor de movimento do bloco colocalizado inferior 6 814 na lista de preditores de vetor de movimento e não verificam ou incluem informações de vetor de movimento de nenhum outro bloco colocalizado. Como ilustrado, o bloco colocalizado inferior 6 814 pode ser um bloco que é inferior e mais próximo à direita do bloco atual 800 (por exemplo, bloco 6 814 está abaixo do bloco atual 800 da Figura 8A e o limite direito de bloco 6 814 é o mesmo que o limite direito do bloco atual 800 da Figura 8A). Ou seja, um limite direito do bloco colocalizado inferior está no mesmo local na imagem, que não a imagem que inclui bloco atual 800, que um limite direito do bloco atual 800. As coordenadas para o bloco 6 814 podem ser $(xColBot, yColBot)$, em que $xColBot = xCb + cbWidth - 1$, e $yColBot = yCb + cbHeight$, em que (xCb, yCb) são as coordenadas para o bloco atual 800 da Figura 8A, $cbWidth$ é a largura do bloco atual 800 da Figura 8A, e $cbHeight$ é a altura do bloco atual 800 da Figura 8A.

[0090] Se informações de vetor de movimento do bloco colocalizado inferior 6 814 não estão disponíveis, o codificador de vídeo 200 e o decodificador de vídeo 300 podem determinar se informações de vetor de movimento do bloco colocalizado direito 7 816 da Figura 8A estão disponíveis, e, se disponíveis, incluem as informações de vetor de movimento do bloco colocalizado direito 7 816 na lista de preditores de vetor de movimento e não verificam ou incluem informações de vetor de movimento de nenhum outro bloco colocalizado. Como ilustrado, o bloco colocalizado direito 7 816 pode ser um bloco à direita e

mais próximo à parte inferior do bloco atual 800 da Figura 8A (por exemplo, bloco 7 816 está à direita do bloco atual 800 da Figura 8A e o limite inferior do bloco 7 816 é o mesmo que o limite inferior do bloco atual 800 da Figura 8A). As coordenadas para o bloco 7 816 podem ser (xColBr, yColBr), em que $xColBr = xCb + cbWidth$, e $yColBr = yCb + cbHeight - 1$, em que (xCb, yCb) são as coordenadas para o bloco atual 800 da Figura 8 A, cbWidth é a largura do bloco atual 800 da Figura 8A, e cbHeight é a altura do bloco atual 800 da Figura 8A.

[0091] Em alguns exemplos, para a ordem de codificação da direita para a esquerda da imagem atual, o codificador de vídeo 200 e o decodificador de vídeo 300 podem determinar se informações de vetor de movimento do bloco colocalizado central 5 828 da Figura 8B estão disponíveis, e, se disponíveis, incluem as informações de vetor de movimento do bloco colocalizado central 5 828 na lista de preditores de vetor de movimento e não verificam ou incluem informações de vetor de movimento de nenhum outro bloco colocalizado.

[0092] Se informações de vetor de movimento do bloco colocalizado central 5 828 não estão disponíveis, o codificador de vídeo 200 e o decodificador de vídeo 300 podem determinar se informações de vetor de movimento do bloco colocalizado inferior 6 830 da Figura 8B estão disponíveis, e, se disponíveis, incluem as informações de vetor de movimento do bloco colocalizado inferior 6 830 na lista de preditores de vetor de movimento e não verificam ou incluem informações de vetor de movimento de nenhum

outro bloco colocalizado. Como ilustrado, o bloco colocalizado inferior 6 830 pode ser um bloco que é inferior e mais próximo à esquerda do bloco atual 800 (por exemplo, bloco 6 830 está abaixo do bloco atual 800 da Figura 8B e o limite esquerdo de bloco 6 830 é o mesmo que o limite esquerdo do bloco atual 800 da Figura 8B). Ou seja, um limite esquerdo do bloco colocalizado inferior está no mesmo local na imagem que não a imagem que inclui o bloco atual 800 como um limite esquerdo do bloco atual 800. As coordenadas para o bloco 6 830 podem ser (xColBot, yColBot), em que $xColBot = xCb$, e $yColBot = yCb + cbHeight$, em que (xCb, yCb) são as coordenadas para o bloco atual 800 da Figura 8B, cbWidth é a largura do bloco atual 800 da Figura 8B, e cbHeight é a altura do bloco atual 800 da Figura 8B.

[0093] Se informações de vetor de movimento do bloco colocalizado inferior 6 830 não estão disponíveis, o codificador de vídeo 200 e o decodificador de vídeo 300 podem determinar se informações de vetor de movimento do bloco colocalizado esquerdo 7 832 da Figura 8B estão disponíveis, e, se disponíveis, incluem as informações de vetor de movimento do bloco colocalizado esquerdo 7 832 na lista de preditores de vetor de movimento e não verificam ou incluem informações de vetor de movimento de nenhum outro bloco colocalizado. Como ilustrado, o bloco colocalizado esquerdo 7 832 pode ser um bloco à esquerda e mais próximo à parte inferior do bloco atual 800 da Figura 8B (por exemplo, bloco 7 832 está à esquerda do bloco atual 800 da Figura 8B e o limite inferior do bloco 7 832 é o

mesmo que o limite inferior do bloco atual 800 da Figura 8B). As coordenadas para o bloco 7 832 podem ser (xColBr, yColBr), em que $xColBr = xCb - 1$, e $yColBr = yCb + cbHeight - 1$, em que (xCb, yCb) são as coordenadas para o bloco atual 800 da Figura 8B, cbWidth é a largura do bloco atual 800 da Figura 8B, e cbHeight é a altura do bloco atual 800 da Figura 8B.

[0094] Em alguns exemplos, as posições colocadas dos blocos 814, 816 da Figura 8A e blocos 830, 832 da Figura 8B podem ser definidos com bloco de granularidade 8x8. Por exemplo, se a CU atual (por exemplo, bloco atual 800 da Figura 8A ou 8B) é 64x64, pode haver múltiplos blocos 8x8 disponíveis na parte inferior, à esquerda ou à direita. Em alguns exemplos, apenas blocos angulados 8x8 (por exemplo, blocos 814, 816, 830, 832) podem ser usados para fornecer um vetor de movimento.

[0095] Além disso, em alguns exemplos, o tamanho da lista de preditores de vetor de movimento pode ser com base no tamanho do bloco atual. Como um exemplo, se o tamanho do bloco atual for maior do que um limite, então, o número máximo de entradas da lista de preditores de vetor de movimento pode ser maior do que se o tamanho do bloco atual é inferior ao limite. Por exemplo, se o tamanho do bloco atual é maior do que Nx4 ou 4xN, em que N é menor do que ou igual a 8, então, o tamanho máximo da lista de preditores de vetor de movimento pode ser seis. Se o tamanho do bloco atual é inferior a Nx4 ou 4xN, em que N é menor do que ou igual a 8, então, o tamanho máximo da lista de preditores de vetor de movimento podem ser quatro. Os

valores acima são meramente um exemplo para ajudar no entendimento e não devem ser considerados limitantes.

[0096] Esta divulgação pode, de forma geral, se referir a "sinalizar" determinadas informações, tais como elementos de sintaxe. O termo "sinalizar" pode, de forma geral, se referir à comunicação de valores de elementos de sintaxe e/ou outros dados usados para decodificar dados de vídeo codificados. Ou seja, o codificador de vídeo 200 pode sinalizar valores para elementos de sintaxe no fluxo de bits. Em geral, sinalizar se refere a gerar um valor no fluxo de bits. Como observado acima, o dispositivo de origem 102 pode transportar o fluxo de bits ao dispositivo de destino 116 substancialmente em tempo real, ou não em tempo real, tal como pode ocorrer ao armazenar elementos de sintaxe no dispositivo de armazenamento 112 para recuperação posterior pelo dispositivo de destino 116.

[0097] Exemplos do codificador de vídeo 200 e decodificador de vídeo 300 são ilustrados e descritos em mais detalhes com relação às Figuras 2 e 3. Antes de descrever as Figuras 2 e 3, o que segue descreve alguns detalhes adicionais do processo de codificação de vídeo, tal como predição de vetor de movimento, com relação às Figuras 4A, 4B, 5A, 5B e 6.

[0098] Como descrito em mais detalhes, as Figuras 7A-7C ilustram alguns exemplos de blocos espacialmente e temporalmente vizinhos que o codificador de vídeo 200 e o decodificador de vídeo 300 avaliam para construir listas de preditores de vetor de movimento. Pode haver certas questões com a construção de listas de preditores de vetor

de movimento usando as técnicas ilustradas nas Figuras 7A-7C. Como descrito em mais detalhes, as Figuras 8A-8C ilustram alguns exemplos adicionais de construção de listas de preditores de vetor de movimento que podem endereçar as questões de construção de listas de preditores de vetor de movimento descritas com relação às Figuras 7A-7C.

[0099] O que segue avalia a estrutura de CU e predição de vetor de movimento em HEVC. Em HEVC, a maior unidade de codificação em uma fatia é denominada bloco de árvore de codificação (CTB) ou unidade de árvore de codificação (CTU). Um CTB contém uma árvore quádrupla, os nós da qual são unidades de codificação.

[0100] O tamanho de um CTB pode variar de 16x16 a 64x64 no perfil principal de HEVC (embora tecnicamente tamanhos de CTB8x8 possam ser suportados). Uma unidade de codificação (CU) pode ter o mesmo tamanho que um CTB de até 8x8. Cada unidade de codificação é codificada com um modo, tal como intercodificada ou intracodificada. Intercodificado também pode ser referido como codificado por interpredição ou interpredito. Intracodificado também pode ser referido como codificado por intrapredição ou intrapredito.

[0101] Quando uma CU é intercodificada, a CU pode ainda ser particionada em 2 ou 4 unidades de predição (PUs) ou pode ser uma PU quando partição adicional não é aplicada. Quando duas PUs estão presentes em uma CU, as PUs podem ser retângulos com metade do tamanho ou do tamanho de dois retângulos com $\frac{1}{4}$ ou $\frac{3}{4}$ do tamanho da CU. Quando a CU é intercodificada, cada PU tem um conjunto de informações de

movimento, que é derivado de um modo de interpredição.

[0102] A seguir, analisa-se a predição de vetor de movimento. No padrão HEVC, existem dois modos de interpredição, denominados modos de mesclagem (pulo é considerado como um caso especial de mesclagem) e de predição de vetor de movimento avançado (AMVP), respectivamente, para uma unidade de predição (PU). No modo AMVP ou de mesclagem, uma lista de preditor de vetor de movimento (MV) (também chamada de lista de candidatos MV) é mantida para múltiplos preditores de vetor de movimento. A lista de preditores de vetor de movimento pode ser referida como lista de preditores de vetor de movimento. O(s) vetor(es) de movimento, assim como os índices de referência no modo de mesclagem, da PU atual são gerados pela obtenção de um candidato a partir da lista de preditores de vetor de movimento. A lista de preditores de vetor de movimento pode conter até 5 candidatos para o modo de mesclagem e apenas dois candidatos para o modo AMVP. Um candidato de mesclagem pode conter um conjunto de informações de movimento, por exemplo, vetores de movimento correspondentes a ambas as listas de imagem de referência (lista 0 e lista 1) e os índices de referência. Se um candidato de mesclagem é identificado por um índice de mesclagem, as imagens de referência utilizadas para a predição do bloco atual, bem como os vetores de movimento associados são determinadas. Em outras palavras, o vetor de movimento e a imagem de referência identificada na lista de preditores de vetor de movimento pelo índice de mesclagem são iguais ao vetor de movimento e a imagem de referência do bloco atual.

[0103] Por outro lado, no modo AMVP, para cada direção de predição potencial de qualquer lista 0 ou lista 1, um índice de referência precisa ser explicitamente sinalizado, junto com um índice de preditor MV (MVP) à lista de preditores de vetor de movimento, uma vez que o candidato de AMVP contém apenas um vetor de movimento. No modo AMVP, os vetores de movimentos preditos podem ser ainda refinados (por exemplo, com base em uma diferença de vetor de movimento (MVD) descrita acima). Os candidatos para ambos os modos são derivados da mesma forma dos mesmos blocos vizinhos espaciais e temporais.

[0104] A seguir, analisa-se blocos vizinhos espaciais cujas informações de vetor de movimento formam preditores de vetor de movimento (também chamados de candidatos de preditor de vetor de movimento). Em alguns exemplos, candidatos de MV espaciais (por exemplo, preditores de vetor de movimento espacial) são derivados de blocos vizinhos mostrados nas Figuras 4A e 4B para uma PU (PU_0) 434 e 438 específica, respectivamente, embora os métodos para gerar os candidatos dos blocos difiram para os modos de AMVP e mesclagem. A Figura 4A é um diagrama conceitual mostrando candidatos vizinhos espaciais para modo de mesclagem. A Figura 4B é um diagrama conceitual mostrando candidatos vizinhos espaciais para modo AMVP.

[0105] No modo de mesclagem, em alguns exemplos, até cinco candidatos de MV espaciais podem ser derivados com a ordem mostrada na Figura 4A, e a ordem é a seguinte: esquerda (0, A1), acima (1, B1), acima à direita (2, B0), abaixo à esquerda (3, A0), e acima à esquerda (4, B2),

conforme mostrado na Figura 4A. Por exemplo, para PU0 434, o bloco A1 é identificado como 0 e está à esquerda de PU0 434, o bloco B1 é identificado como 1 e está acima de PU0 434, o bloco B0 é identificado como 2 e está acima à direita de PU0 434 e acima de PU1 436, o bloco A0 é identificado como 3 e está abaixo à esquerda de PU0 434, e o bloco B2 é identificado como 4 e está acima à esquerda de PU0 434.

[0106] No modo AMVP, em alguns exemplos, os blocos vizinhos são divididos em dois grupos: um grupo da esquerda que inclui o bloco 0 e 1 que estão abaixo à esquerda e à esquerda de PU0 438, respectivamente, e um grupo acima que inclui os blocos 2, 3 e 4 que estão acima à direita, acima e acima à esquerda de PU01 438 conforme mostrado na Figura 4B. O bloco 2 está acima de PU1 440. Para cada grupo, o candidato potencial em um bloco vizinho referindo-se à mesma imagem de referência que indicado pelo índice de referência sinalizado tem a maior prioridade de ser escolhido para formar um candidato final do grupo. Todos os blocos vizinhos podem não conter um vetor de movimento apontando para a mesma imagem de referência. Portanto, se tal candidato não puder ser encontrado, o primeiro candidato disponível pode ser escalado para formar o candidato final e, portanto, as diferenças de distância temporal podem ser compensadas.

[0107] A seguir, analisa-se a predição de vetor de movimento temporal com as Figuras 5A e 5B. A Figura 5A é um diagrama conceitual mostrando um candidato de preditor de vetor de movimento temporal (TMVP). A Figura 5B é um

diagrama conceitual mostrando a escala de vetor de movimento para TMVP.

[0108] Em alguns exemplos, um candidato de TMVP, se habilitado e disponível, é adicionado à lista de candidatos de MV (por exemplo, lista de preditores de vetor de movimento) após os candidatos de vetor de movimento espacial (por exemplo, informações de vetor de movimento de blocos espacialmente vizinhos). O processo de derivação de vetor de movimento para o candidato de TMVP é o mesmo para ambos os modos de mesclagem e AMVP, no entanto, o índice de referência alvo para o candidato de TMVP em modo de mesclagem é definido como 0.

[0109] Em alguns exemplos, a localização do bloco primário para derivação de candidato de TMVP é o bloco inferior direito fora da PU colocalizada, como mostrado na Figura 5A como um bloco T 540. A localização do bloco T 540 pode ser para compensar o desvio para os blocos acima e à esquerda usados para gerar candidatos vizinhos espaciais. No entanto, se o bloco T 540 está localizado fora da linha do CTB atual ou informações de movimento não estão disponíveis para o bloco T 540, o bloco T 540 é substituído por um bloco central 541 da PU.

[0110] Na Figura 5B, um vetor de movimento 548 para candidato de TMVP na imagem atual 550 é derivado da PU colocalizada da imagem colocalizada 546, indicada no nível de fatia. O vetor de movimento (MV) para a PU colocalizada é denominado MV colocalizado 542. Para derivar o vetor de movimento de candidato de TMVP, o MV colocalizado 542 pode ser escalado para compensar as diferenças de distância

temporal, como mostrado na Figura 5B. Por exemplo, a diferença temporal entre a imagem colocalizada 546 e a imagem de referência colocalizada 544 e a diferença temporal entre a imagem atual 550 e a imagem de referência atual 552 é usada para escalar o MV colocalizado 542 para gerar o vetor de movimento 548.

[0111] A seguir, analisa-se alguns outros aspectos da predição de movimento em HEVC. Vários aspectos de modos de mesclagem e AMVP são descritos abaixo.

[0112] **Escala do vetor de movimento:** o valor de vetores de movimento pode ser proporcional à distância de imagens no tempo de apresentação. Um vetor de movimento associa duas imagens, a imagem de referência e a imagem que contém o vetor de movimento (a saber, imagem contentora). Quando um vetor de movimento é utilizado para prever o outro vetor de movimento, a distância da imagem contentora e a imagem de referência é calculada com base nos valores de Contagem de Ordem de Imagem (POC).

[0113] Para que um vetor de movimento seja predito, ambas a sua imagem contentora e imagem de referência podem ser diferentes. Portanto, uma nova distância (com base em POC) é calculada, o vetor de movimento é escalado com base nessas duas distâncias POC. Para um candidato vizinho espacial, as imagens contentoras para os dois vetores de movimento são as mesmas, enquanto as imagens de referência são diferentes. Em HEVC, a escala de vetor de movimento aplica-se a ambos TMVP e AMVP para candidatos vizinhos espaciais e temporais.

[0114] **Geração de candidato de vetor de movimento**

artificial: se uma lista de preditores de vetor de movimento não é completa (por exemplo, menos candidatos do que um número predeterminado), os candidatos do vetor de movimento artificial são gerados e inseridos no final da lista de preditores de vetor de movimento até que a lista de preditores de vetor de movimento tenha todos os candidatos. Em modo de mesclagem, existem dois tipos de candidatos MV artificiais: candidato combinado derivado apenas para fatias B e candidatos zero usados apenas para AMVP se o candidato combinado derivado apenas para fatias B não fornece candidatos artificiais suficientes para preencher a lista de preditores de vetor de movimento.

[0115] Para cada par de candidatos que já estão na lista de preditores de vetor de movimento e têm as informações de movimento necessárias, os candidatos de vetor de movimento combinado bidirecional são derivados por uma combinação de vetor de movimento do primeiro candidato referente a uma imagem na lista 0 e o vetor de movimento de um segundo candidato referindo-se a uma imagem na lista 1.

[0116] **Processo de poda para inserção de candidato:** candidatos de diferentes blocos podem ser iguais, o que diminui a eficiência de uma lista de preditores de vetor de movimento de mesclagem/AMVP. Um processo de poda é aplicado para resolver este problema. O processo de poda compara um candidato contra os outros na lista de preditores de vetor de movimento atual, para evitar a inserção de candidatos idênticos. Para reduzir a complexidade, apenas um número limitado de processos de poda são aplicados para evitar a comparação de cada

candidato potencial com todos os outros candidatos existentes na lista.

[0117] O desenvolvimento de VVC inclui predição de vetor de movimento aprimorada. Por exemplo, várias ferramentas de intercodificação foram propostas, as quais derivam ou refinam a lista de preditores de vetor de movimento de predição de vetor de movimento ou predição de mesclagem para um bloco atual. Vários exemplos são descritos abaixo.

[0118] A seguir, descreve-se predição de movimento com base em histórico (HMVP), conforme descrito em L. Zhang, K. Zhang, H. Liu, Y. Wang, P. Zhao, e D. Hong, "CE4-related: History-based Motion Vector Prediction", JVET-K0104, julho de 2018. HMVP é um método baseado em histórico, em que o codificador de vídeo 200 e o decodificador de vídeo 300 determinam um preditor de MV para cada bloco a partir de uma lista de MVs decodificada do passado, além daqueles MVs em campos de movimento vizinhos causais imediatamente adjacentes (por exemplo, MVs de blocos imediatamente e espacialmente vizinhos são exemplos de MVs em campos de movimento vizinhos causais imediatamente adjacentes). HMVP inclui codificador de vídeo 200 e decodificador de vídeo 300 construindo uma tabela para vetores de movimento previamente decodificados como candidatos de HMVP.

[0119] Por exemplo, o codificador de vídeo 200 e o decodificador de vídeo 300 constroem uma tabela com múltiplos candidatos de HMVP durante o processo de codificação/decodificação. Construir a tabela pode incluir

adicionar candidatos de HMVP à tabela, bem como remover candidatos de HMVP da tabela. O codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para esvaziar a tabela (por exemplo, remover os candidatos de HMVP) quando uma nova fatia for encontrada para codificar ou decodificar. Sempre que houver um bloco intercodificado, o codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para inserir as informações de movimento associadas na tabela no modo FIFO (first-in-first-out) como um novo candidato de HMVP. Então, o codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para aplicar uma regra FIFO de restrição. Em algumas técnicas, ao inserir um candidato de HMVP à tabela, o codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para primeiramente aplicar uma verificação de redundância (por exemplo, poda) para determinar se há um candidato de HMVP idêntico na tabela. Se encontrado, o codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para remover aquele candidato de HMVP específico da tabela e todos os candidatos de HMVP após aquele candidato ser movido.

[0120] O codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para usar candidatos HMVP no processo de construção de lista de candidatos de mesclagem. Por exemplo, o codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para inserir todos os candidatos HMVP da última entrada em uma primeira entrada na tabela após o

candidato TMVP. O codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para aplicar poda nos candidatos HMVP. Uma vez que o número total de candidatos de mesclagem disponíveis atinge o número máximo sinalizado ou predeterminado de candidatos de mesclagem permitidos, o codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para encerrar o processo de construção de lista de candidatos de mesclagem. A construção de lista de candidatos de mesclagem é um exemplo de construção de uma lista de preditores de vetor de movimento.

[0121] Da mesma forma, o codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para usar candidatos HMVP no processo de construção de lista de candidatos AMVP para construir uma lista de candidatos AMVP. O codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para inserir os vetores de movimento dos últimos candidatos K HMVP na tabela após o candidato TMVP. O codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para usar apenas candidatos HMVP com a mesma imagem de referência que a imagem de referência alvo AMVP para construir a lista de candidatos AMVP. O codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para aplicar poda nos candidatos HMVP. A construção de lista de candidatos AMVP é outro exemplo de construção de uma lista de preditores de vetor de movimento.

[0122] O que segue descreve candidatos de mesclagem espacial não vizinho. A construção de candidatos

de mesclagem espacial não vizinho, como descrito em R. Yu, et. al., "CE 4-2.1: Adding non-adjacents spatial merge candidates", Equipe de Especialistas em Vídeo Conjunta (JVET) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG11, 11º Encontro: Ljubljana, SI, 10-18 de julho 2018 (doravante "JVET-K0228"), envolve derivação de novos candidatos espaciais de duas posições vizinhas não adjacentes. Por exemplo, como mostrado na Figura 6, o codificador de vídeo 200 e o decodificador de vídeo 300 podem derivar novos candidatos espaciais do bloco não adjacente mais próximo 650 acima do bloco atual 648 e/ou do bloco não adjacente mais próximo 652 à esquerda do bloco atual 648. Por exemplo, a Figura 6 ilustra o bloco atual 648 com blocos vizinhos A0, A1, B2, B1 e B0. Em alguns exemplos, os candidatos de mesclagem espacial não vizinhos incluem informações de movimento de bloco HN 652 e bloco VN 650 da Figura 6 que são não espacialmente vizinhos ao bloco atual 648. Ou seja, a Figura 6 ilustra exemplos de blocos cujas informações de vetor de movimento podem ser usadas para construir a tabela de histórico de candidatos HMVP.

[0123] Os blocos 650 e 652 são limitados dentro de uma distância máxima de 1 CTU para o bloco atual 648. O processo de pesquisa e carregamento de candidatos não adjacentes inicia com rastreamento dos blocos previamente decodificados na direção vertical. O rastreamento inverso vertical para quando um interbloco é encontrado ou a distância de volta rastreada atinge o tamanho de 1 CTU.

[0124] O processo de pesquisa e carregamento, então, rastreia os blocos previamente decodificados na

direção horizontal. O critério para interromper o processo de pesquisa e carregamento horizontal depende de se há um candidato não adjacente vertical sendo eficazmente pesquisado e carregado ou não. Se nenhum candidato não adjacente vertical é pesquisado e carregado, o processo de pesquisa e carregamento horizontal para quando um bloco intercodificado é encontrado ou a distância de volta rastreada excede o limite de tamanho de uma CTU. Se há um candidato não adjacente vertical pesquisado e carregado, então, o processo de pesquisa e carregamento horizontal para quando um bloco intercodificado que contém um MV diferente do candidato não adjacente vertical é encontrado ou a distância de volta rastreada excede o limite de tamanho de uma CTU. Os candidatos vizinhos não adjacentes pesquisados e carregados são adicionados antes do candidato TMVP na lista de candidatos de mesclagem.

[0125] Como descrito acima, para modo de mesclagem e modo AMVP, o codificador de vídeo 200 e o decodificador de vídeo 300 podem construir uma lista de candidatos de vetor de movimento (MV), também denominada lista de preditores de vetor de movimento. A lista de preditores de vetor de movimento inclui informações de vetor de movimento de blocos espacialmente vizinhos (por exemplo, blocos que são vizinhos ao bloco atual e estão na mesma imagem que o bloco atual), blocos temporalmente vizinhos (por exemplo, blocos que são vizinhos ao bloco atual, mas estão na imagem diferente do bloco atual), e, em alguns exemplos, informações de vetor de movimento de candidatos HMVP (por exemplo, entradas de uma tabela para

vetores de movimento previamente decodificados que formam os candidatos HMVP). Os blocos temporalmente vizinhos são também denominados blocos colocalizados.

[0126] Portanto, para construir a lista de preditores de vetor de movimento, o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar blocos vizinhos (por exemplo, blocos espacialmente e temporalmente vizinhos) para determinar se os blocos vizinhos têm informações de vetor de movimento. Se um bloco vizinho tem informações de vetor de movimento (por exemplo, porque o bloco vizinho é intercodificado), o codificador de vídeo 200 e o decodificador de vídeo 300 podem incluir as informações de vetor de movimento do bloco vizinho na lista de preditores de vetor de movimento. O codificador de vídeo 200 e o decodificador de vídeo 300 podem manter uma tabela separada de informações de vetor de movimento de blocos recentemente codificados ou decodificados para os candidatos HMVP.

[0127] Pode haver uma pluralidade de blocos vizinhos que o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar (por exemplo, verificar) para determinar se os blocos vizinhos têm informações de vetor de movimento. Para limitar o número de blocos vizinhos que o codificador de vídeo 200 e o decodificador de vídeo 300 podem precisar avaliar, o que segue descreve alguns exemplos de projeto para predição de vetor de movimento espacial e temporal (por exemplo, exemplos de blocos espacialmente e temporalmente vizinhos que são avaliados).

[0128] Em algumas implementações de modo de

mesclagem (em alguma terminologia, isso pode ser denominado como modo direto ou de pulo), os candidatos de preditor de vetor de movimento (MVP) temporal e espacial podem ser visitados (por exemplo, verificados) em uma determinada ordem para popular a lista de preditores de vetor de movimento. Um MVP espacial (SMVP) pode ser um vetor de movimento de um bloco espacialmente vizinho que pode ser usado como um preditor de vetor de movimento para o vetor de movimento do bloco atual. Um MVP temporal (TMVP), como descrito acima, pode ser um vetor de movimento para um bloco temporalmente vizinho (por exemplo, bloco colocalizado) que pode ser usado como um preditor de vetor de movimento para o vetor de movimento do bloco atual.

[0129] Em algumas técnicas, para determinar os SMVPs e TMVPs, o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar blocos vizinhos (por exemplo, blocos espacialmente ou temporalmente vizinhos) em uma ordem específica. A ordem em que o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar blocos vizinhos pode também ser referida como uma ordem de visitação.

[0130] A Figura 7A é um diagrama conceitual ilustrando candidatos espaciais e temporais exemplificativos usados em predição de vetor de movimento. A Figura 7B é um diagrama conceitual ilustrando uma ordem de visitação exemplificativa para candidatos espaciais. A Figura 7C é um diagrama conceitual ilustrando outra ordem de visitação exemplificativa para candidatos espaciais.

[0131] A Figura 7A mostra exemplo de vizinhos

especiais utilizados como candidatos de MVP. Na Figura 7A, os vizinhos espaciais (por exemplo, blocos espacialmente vizinhos) do bloco atual 700 incluem bloco A 702, B 704, (C 706, A1 710A|B1 710B), bloco A0 714, e bloco B2 712. O codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar os blocos vizinhos com um processo de dois estágios. A ordem em que os candidatos espaciais (por exemplo, blocos espacialmente vizinhos) são verificados (isto é, a ordem de visitação marcada com números de 0-5), é mostrada na Figura 7B.

[0132] Os candidatos do Grupo 1 da Figura 7A incluem:

a. blocos A 702, B 704, C 706 (colocalizados com bloco B0 708 em notação HEVC) e são mostrados em sombreamento hash, e

b. blocos A1 710A ou B1 710B, em que o bloco A1 710A ou bloco B1 710B é incluído dependendo de se o preditor de vetor de movimento no bloco C 706 está disponível.

[0133] Os candidatos do Grupo 2 incluem:

a. blocos A0 714 e B2 712

[0134] Na Figura 7A, o bloco C 706 é colocalizado com o bloco B0 708. Por exemplo, o bloco C 706 e o bloco B0 708 podem ser os mesmos blocos, mas dois nomes diferentes são dados na Figura 7A devido a alterações na nomenclatura entre HEVC e VVC. O padrão HEVC referiu-se ao bloco ilustrado como bloco B0 708 como bloco "B0". O padrão VVC referiu-se ao bloco ilustrado como bloco B0 708 como bloco "C". Portanto, o bloco C 706 é colocalizado com o bloco B0

708 e está na mesma imagem que o bloco atual 700.

[0135] Como um exemplo de avaliação de acordo com Grupo 1 e Grupo 2, o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar (por exemplo, verificar) blocos no Grupo 1. O codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar o bloco A 702 e determinar se há informações de vetor de movimento para o bloco A 702. Se há informações de vetor de movimento para o bloco A 702 (por exemplo, porque o bloco A 702 foi intercodificado com um vetor de movimento), o codificador de vídeo 200 e o decodificador de vídeo 300 podem adicionar informações de vetor de movimento do bloco A 702 à lista de preditores de vetor de movimento. Independentemente de se o bloco A 702 tem informações de vetor de movimento, o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar o bloco B 704 e determinar se há informações de movimento para o bloco B 704. Se há informações de vetor de movimento para o bloco B 704 (por exemplo, porque o bloco B 704 foi intercodificado com um vetor de movimento), o codificador de vídeo 200 e o decodificador de vídeo 300 podem adicionar informações de vetor de movimento do bloco B 704 à lista de preditores de vetor de movimento. Independentemente de se bloco B 704 tem informações de vetor de movimento, o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar o bloco C 706 e determinar se há informações de movimento para o bloco C 706. Se há informações de vetor de movimento para o bloco C 706 (por exemplo, porque o bloco C 706 foi intercodificado com um vetor de movimento), o codificador de vídeo 200 e o

decodificador de vídeo 300 podem adicionar informações de vetor de movimento do bloco C 706 à lista de preditores de vetor de movimento.

[0136] Em alguns exemplos, o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar um do bloco A1 710A ou bloco B1 710B com base em se havia informações de vetor de movimento para o bloco C 706. Por exemplo, se há informações de vetor de movimento para o bloco C 706, então, o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar um do bloco A1 710A ou bloco B1 710B (por exemplo, bloco A1 710A). No entanto, se não há informações de vetor de movimento para o bloco C 706 (por exemplo, porque o bloco C 706 foi intracodificado), então, o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar o outro dentro o bloco A1 710A ou bloco B1 710B (por exemplo, bloco B1 710B).

[0137] Após avaliar um dos blocos A1 710A ou B1 710B, o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar os blocos do Grupo 2 (por exemplo, blocos B2 712 e A0 714) e adicionar informações de vetor de movimento à lista de preditores de vetor de movimento, como descrito acima. Em seguida, o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar os vizinhos temporais (por exemplo, blocos temporalmente vizinhos ou blocos colocalizados).

[0138] Na Figura 7A, os vizinhos temporais (por exemplo, blocos temporalmente vizinhos) do bloco atual 700 incluem o bloco C0 716 e um ou mais dos blocos 718. Como

mostrado na Figura 7A, vizinhos temporalmente colocalizados utilizados como candidatos de MVP são de um bloco colocalizado no centro (marcado como bloco CO 716) do bloco atual e um bloco 718 A (com contorno tracejado) no local mais inferior-direito fora do bloco atual. O bloco CO 716 e blocos 718 podem ser uma imagem de referência que é diferente da imagem atual que inclui o bloco atual 700. Os blocos candidatos do Grupo 3 (mostrados com contorno tracejado) incluem:

- a. blocos CO 716, H 718 A
- b. Se a localização H (por exemplo, bloco H 718A) é encontrada fora da imagem colocalizada, uma ou mais posições H d retorno (por exemplo, blocos H 718B-718D) podem ser usadas em vez disso.

[0139] Por exemplo, o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar o bloco CO 716 para determinar se há informações de vetor de movimento para o bloco CO 716 e adicionar as informações de vetor de movimento para o bloco CO 716, se disponíveis, à lista de preditores de vetor de movimento. Da mesma forma, o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar o bloco H 718 A para determinar se há informações de vetor de movimento para o bloco H 718 A e adicionar as informações de vetor de movimento para o bloco H 718 A, se disponíveis, à lista de preditores de vetor de movimento. Se o bloco H 718A não está disponível (por exemplo, porque está fora da imagem), o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar um

dos blocos H 718B-718D.

[0140] O tamanho da lista de preditores de vetor de movimento pode ser fixo (isto é, pode haver número fixo de entradas na lista de preditores de vetor de movimento). Por conseguinte, o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar (por exemplo, verificar) blocos vizinhos e adicionar informações de vetor de movimento à lista de preditores de vetor de movimento de blocos vizinhos desde que o tamanho máximo da lista de preditores de vetor de movimento não seja atingido. Além disso, se o tamanho da lista de preditores de vetor de movimento não é atingido após avaliar blocos temporalmente vizinhos (por exemplo, colocados) CO 716 e um dos blocos H 718A-718D, o codificador de vídeo 200 e o decodificador de vídeo 300 podem adicionar candidatos de HMVP à lista de preditores de vetor de movimento ou candidatos de vetor de movimento artificiais à lista de preditores de vetor de movimento.

[0141] Como descrito acima, pode haver uma ordem em que o codificador de vídeo 200 e o decodificador de vídeo 300 avaliam os blocos vizinhos. A Figura 7B ilustra uma ordem exemplificativa. Por exemplo, na Figura 7B, "0" se refere ao bloco A 702, e é o primeiro bloco que o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar. "1" se refere ao bloco B 704, e é o segundo bloco que o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar, e assim por diante.

[0142] Na Figura 7B, "3" se refere aos blocos A1 710A e B1 710B. Isso é para indicar que o codificador de

vídeo 200 e o decodificador de vídeo 300 podem avaliar um dos blocos A1 710A ou B1 710B com base em se há informações de vetor de movimento para o bloco C 706 (identificados como "2").

[0143] Em alguma implementação exemplificativa, dependendo do particionamento de bloco usado e da ordem de codificação, uma ordem de candidatos de MVP espaciais inversos pode ser usada, como mostrado na Figura 7C, em que a ordem de verificação procede de 0-5. Por exemplo, consistente com as Figuras 7A e 7B, se o codificador de vídeo 200 e o decodificador de vídeo 300 codificam ou decodificam, respectivamente, usando uma ordem de codificação da esquerda para a direita e de cima para baixo para blocos em uma imagem, então, os blocos A 702, B 704, C 706, A1 710A, B1 701B, B2 712, e A0 714 seriam codificados ou decodificados antes de o bloco atual 700 ser codificado ou decodificado. Porque os blocos A 702, B 704, C 706, A1 710A, B1 701B, B2 712, e A0 714 seriam codificados ou decodificados antes de o bloco atual 700 ser codificado ou decodificado, para codificar ou decodificar o bloco atual 700, o codificador de vídeo 200 e o decodificador de vídeo 300 podem ser capazes de avaliar informações de vetor de movimento para os blocos A 702, B 704, C 706, A1 710A, B1 701B, B2 712, e A0 714. Por exemplo, porque o bloco A 702 foi codificado ou decodificado antes do bloco atual 700 em uma ordem de codificação da esquerda para a direita, de cima para baixo, se o bloco A 702 foi intercodificado, o codificador de vídeo 200 e o decodificador de vídeo 300 podem ser capazes de acessar as informações de vetor de

movimento para o bloco A 702.

[0144] No entanto, em alguns exemplos, a ordem de codificação pode não ser da esquerda para a direita e de cima para baixo. Em vez disso, a ordem de codificação podem ser da direita para a esquerda e de cima para baixo. Em tais exemplos, alguns dos blocos vizinhos ilustrados nas Figuras 7A e 7B podem não ter sido codificados ou decodificados antes do bloco atual 700. Portanto, mesmo se o bloco vizinho tiver que ser intercodificado, as informações de vetor de movimento para os blocos vizinhos não estão ainda disponíveis porque alguns dos blocos vizinhos nas Figuras 7A e 7B não teriam sido codificados ou decodificados na ordem de codificação da direita para a esquerda.

[0145] A Figura 7C ilustra um exemplo dos blocos espacialmente vizinhos que seriam avaliados e a ordem em que os blocos espacialmente vizinhos seria avaliada em uma ordem de codificação da direita para a esquerda e de cima para baixo de blocos na imagem. Por exemplo, na Figura 7C, o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar blocos 720, 722, 724, um dos blocos 726A ou 726B, 728, e 730. Por exemplo, o codificador de vídeo 200 e o decodificador de vídeo 300 podem iniciar a avaliação, para determinar se há informações de vetor de movimento, começando do bloco 720 (identificado como "0"), seguido pelo bloco 722 (identificado como "1"), e então o bloco 724 (identificado como "2").

[0146] Similar aos blocos A1 710A e B1 710B das Figuras 7A e 7B, na Figura 7C, o codificador de vídeo 200 e

o decodificador de vídeo 300 podem avaliar um dos blocos 726A ou 726B (ambos identificados como "3") com base em se há informações de vetor de movimento para o bloco 724 (por exemplo, com base em se o bloco 724 está intercodificado ou não). Como ilustrado, após um dos blocos 726A ou 726B, o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar o bloco 728 e, então, o bloco 730.

[0147] Acima, descreve-se uma ordem exemplificativa em que o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar blocos vizinhos para determinar se há informações de vetor de movimento a serem incluídas na lista de preditores de vetor de movimento. Em alguns exemplos, em vez de avaliação um por um, pode ser possível que o codificador de vídeo 200 e o decodificador de vídeo 300 avaliem uma pluralidade dos blocos vizinhos em paralelo. Em tais exemplos, o codificador de vídeo 200 e o decodificador de vídeo 300 podem ordenar as informações de vetor de movimento na lista de preditores de vetor de movimento na ordem ilustrada nas Figuras 7A-7C.

[0148] Como um exemplo, o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar o bloco A 702 e o bloco B 704 em paralelo, em vez de um após o outro. Neste exemplo, se ambos o bloco A 702 e o bloco B 704 têm informações de vetor de movimento, então, o codificador de vídeo 200 e o decodificador de vídeo 300 podem construir a lista de preditores de vetor de movimento, tal que as informações de vetor de movimento para o bloco A 702 estão antes das informações de vetor de movimento para o bloco B

704 porque o bloco A 702 está antes do bloco B 704 na ordem de avaliação.

[0149] No entanto, nos exemplos ilustrados nas Figuras 7A-7C, pode haver um limite para avaliar informações de vetor de movimento em paralelo, que pode impactar eficiências computacionais. Por exemplo, se o bloco A1 710A ou o bloco B1 710B deve ser avaliado baseia-se em se há informações de vetor de movimento para o bloco C 706. Em tais exemplos, o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar em série o bloco C 706 e, então, um do bloco A1 710A ou bloco B1 710B com base no resultado da avaliação do bloco C 706 (por exemplo, com base em se há informações de vetor de movimento para o bloco C 706). Na Figura 7C, em vez de avaliação em paralelo, o codificador de vídeo 200 e o decodificador de vídeo 300 podem avaliar em série o bloco 724 e, então, um do bloco 726A ou bloco 726B com base no resultado de avaliação do bloco 724 (por exemplo, com base em se há informações de vetor de movimento para o bloco 724).

[0150] Da mesma forma, se o codificador de vídeo 200 e o decodificador de vídeo 300 avaliam um dos blocos H 718B-718D pode ser com base em se o bloco 718A está disponível (por exemplo, não fora da imagem). Embora blocos temporalmente vizinhos não sejam mostrados na Figura 7C, o mesmo pode se aplicar a blocos temporalmente vizinhos como se aplicaria a blocos H 718A-718D na Figura 7A.

[0151] Dito de outra forma, os projetos de candidato de MVP, tais como o padrão descrito nas Figuras 7A-7C, usam pesquisa e carregamento de MVP condicional

(A1|B1) (por exemplo, seja o bloco A1 710A ou o bloco B1 710B), com base na disponibilidade dos determinados candidatos de MVP (por exemplo, com base na disponibilidade de informações de vetor de movimento para o bloco C 706 (também chamada de disponibilidade de candidato C)), no tipo de particionamento de bloco ou na posição espacial do candidato de MVP, tal como seleção de um grupo de candidatos H (por exemplo, blocos H 718A-718D) mostrado na Figura 7A. Essa pesquisa e carregamento condicional aumenta a complexidade computacional e pode afetar a paralelização ou restrições de memória para determinada implementação de MVP.

[0152] A construção de uma lista de preditores de vetor de movimento, que, em alguns exemplos, pode requerer um grande número de dependências e condições, pode introduzir significativa complexidade relativa (por pixel), especialmente para tamanhos de bloco pequenos, por exemplo, 4x4 ou 4xN, em que $N \leq 8$. Para blocos menores, pode não haver tanta diversidade em informações de vetor de movimento para blocos espacialmente vizinhos. Por exemplo, um bloco vizinho à esquerda e um bloco vizinho acima ao bloco atual podem estar mais próximos um do outro para um bloco menor do que para um bloco maior. Como uma ilustração, na Figura 7A, quanto maior for o bloco atual 700, mais distantes o bloco A 702 e o bloco B0 708 estão um do outro, e quanto menor o bloco atual 700, mais próximos o bloco A 702 e o bloco B0 708 estão um do outro.

[0153] Dois blocos que estão mais próximos um do outro tendem a ter informações de vetor de movimento

similares. Portanto, para blocos menores, há uma maior probabilidade de que os blocos espacialmente vizinhos tenham informações de vetor de movimento similares em comparação com blocos espacialmente vizinhos para blocos maiores. Porque há uma maior probabilidade de que os blocos espacialmente vizinhos tenham informações de vetor de movimento similares, ter uma lista de preditores de vetor de movimento relativamente grande pode não prover diversidade em informações de vetor de movimento. Em outras palavras, embora ter uma lista de preditores de vetor de movimento relativamente grande para blocos menores possa fornecer muitos candidatos de preditor de vetor de movimento, a diversidade nas informações de vetor de movimento pode não ser grande. No entanto, para blocos maiores, ter uma lista de preditores de vetor de movimento relativamente grande pode fornecer diversas informações de vetor de movimento permitindo a inclusão de candidatos de preditor de vetor de movimento tendo valores relativamente próximos ao vetor de movimento real.

[0154] Por conseguinte, ter a lista de preditores de vetor de movimento de mesmo tamanho para blocos de todos os tamanhos pode não ser desejável. Para blocos pequenos, ter uma lista de preditores de vetor de movimento que é do mesmo tamanho que um bloco grande significa que o codificador de vídeo 200 e o decodificador de vídeo 300 precisam realizar várias operações para preencher a lista de preditores de vetor de movimento sem alcançar ganhos de codificação por ter múltiplos candidatos de preditor de vetor de movimento por causa da falta de diversidade nas

informações de vetor de movimento. Portanto, para blocos menores, uma lista de preditores de vetor de movimento menor pode ser benéfica. No entanto, para um bloco maior, ter uma lista de preditores de vetor de movimento relativamente pequena pode ser desvantajoso porque diversos candidatos de preditor de vetor de movimento podem ser deixados de fora da lista de preditores de vetor de movimento devido ao pequeno tamanho da lista de preditores de vetor de movimento.

[0155] Diante dessas desvantagens, esta divulgação descreve as seguintes técnicas para realizar a construção da lista de preditores de vetor de movimento espaço-temporal. As técnicas exemplificativas podem ser aplicadas separadamente ou em combinação. Além disso, as técnicas exemplificativas podem resolver os problemas acima mencionados. No entanto, as técnicas exemplificativas não devem ser consideradas limitadas a abordar as questões descritas acima.

[0156] Em um exemplo, esta divulgação descreve técnicas para determinar o padrão em que candidatos de MVP espacial e temporal são gerados. O codificador de vídeo 200 e o decodificador de vídeo 300 podem empregar o seguinte processo de construção de lista de preditores de vetor de movimento a partir de candidatos da vizinhança espaço-temporal. A prioridade de colocação dos candidatos na lista de preditores de vetor de movimento é marcada pelo número de 0 ... 7, conforme mostrado na Figura 8A.

[0157] Primeiramente, o codificador de vídeo 200 e o decodificador de vídeo 300 visitam (isto é, verificam a

disponibilidade de informações de movimento) em candidatos de MVP espaciais na ordem 0...4 mostrada em blocos de linhas sólidas na Figura 8 A. Por exemplo, o codificador de vídeo 200 e o decodificador de vídeo 300 podem determinar se informações de vetor de movimento estão disponíveis para o bloco 0 802, bloco 1 804, bloco 2 806, bloco 3 808, e bloco 4 810.

[0158] Em seguida, o codificador de vídeo 200 e o decodificador de vídeo 300 visitam (isto é, verificam a disponibilidade de informações de movimento) os candidatos de preditor de vetor de movimento a partir de posições colocalizadas temporais (5, 6 e 7), que são mostradas como blocos em linhas pontilhadas. Por exemplo, o codificador de vídeo 200 e o decodificador de vídeo 300 podem determinar se informações de vetor de movimento estão disponíveis para o bloco 5 812, também denominado bloco colocalizado central 5 812, ou bloco 6 814, também denominado bloco colocalizado abaixo 6 814 (em que o bloco colocalizado abaixo 6 814 pode estar na parte inferior e mais próximo à direita do bloco atual 800 da Figura 8A), ou bloco 7 816, também denominado bloco colocalizado direito 7 816 (em que o bloco colocalizado direito 7 816 pode estar à direita e mais próximo à parte inferior do bloco atual 800 da Figura 8B).

[0159] A Figura 8A é um diagrama conceitual ilustrando um padrão de preditor de vetor de movimento espacial temporal exemplificativo de acordo com um exemplo da divulgação. A Figura 8B é um diagrama conceitual ilustrando um padrão de preditor de vetor de movimento espacial temporal invertido exemplificativo de acordo com

um exemplo da divulgação. A Figura 8C é um diagrama conceitual ilustrando outro padrão de preditor de vetor de movimento espacial temporal invertido exemplificativo de acordo com um exemplo da divulgação. As Figuras 8A-8C mostram diferentes ordens de verificação (por exemplo, como indicado pelos números nos blocos), bem como diferentes localizações para blocos colocados temporais (por exemplo, como mostrado pelos blocos com linha tracejada).

[0160] Dependendo do modo de codificação, tipo de particionamento e/ou direção de codificação, especialmente modos de codificação que envolvem processamento de ordem de varredura inversa, o padrão espacialmente invertido (ou espelhado) pode ser empregado, conforme mostrado na Figura 8B. Em alguns exemplos, parte dos padrões mostrados na Figura 8A podem ser invertidos, por exemplo, apenas posições espaciais são invertidas, e localizações candidatas temporais permanecem as mesmas. Ainda em outro exemplo, o padrão (por exemplo, a localização relativa) de candidatos de MVP pode permanecer igual. No entanto, uma ordem de visitação pode ser invertida espacialmente (espelhada) ao longo de determinado eixo (por exemplo, eixo diagonal) como identificável mostrado na Figura 8C.

[0161] Em um ou mais exemplos, o codificador de vídeo 200 e o decodificador de vídeo 300 podem, cada um, construir uma lista de preditores de vetor de movimento com base na ordem de codificação da imagem que inclui o bloco atual 800. Um exemplo da ordem de codificação é a ordem de codificação da esquerda para a direita, em cujo caso, o codificador de vídeo 200 e o decodificador de vídeo 300

codificam ou decodificam bloco a bloco começando do canto superior esquerdo da imagem atual até atingir a extremidade direita da imagem atual, seguido pelo bloco abaixo do canto superior esquerdo e assim sucessivamente até chegar ao bloco localizado na parte inferior direita. Outro exemplo da ordem de codificação é a ordem de codificação da direita para a esquerda, em cujo caso, o codificador de vídeo 200 e o decodificador de vídeo 300 podem codificar ou decodificar bloco por bloco começando do canto superior direito da imagem atual até atingir a extremidade esquerda da imagem atual, seguido pelo bloco abaixo do canto superior direito, e assim sucessivamente até chegar ao bloco localizado na parte inferior esquerda.

[0162] A ordem de codificação pode impactar quais blocos espacialmente vizinhos estão disponíveis para verificação. Por exemplo, as informações de vetor de movimento apenas dos blocos previamente codificados ou decodificados podem estar disponíveis. Portanto, se um bloco estiver ainda a ser codificado ou decodificado, as informações de vetor de movimento para aquele bloco podem não estar ainda disponíveis. Porque a ordem de codificação define a ordem em que blocos são codificados ou decodificados, se as informações de vetor de movimento de um bloco espacialmente vizinho podem ser determinadas baseia-se na ordem de codificação.

[0163] Como um exemplo, na Figura 8A, para a ordem de codificação da esquerda para a direita, as informações de vetor de movimento do bloco 0 802 podem estar disponíveis no momento da codificar ou decodificação

do bloco atual 800. No entanto, as informações de vetor de movimento de um bloco vizinho à direita ao bloco atual 800 podem não estar disponíveis. Na Figura 8B, para a ordem de codificação da direita para a esquerda, as informações de vetor de movimento do bloco 0 818 podem estar disponíveis no momento da codificação ou decodificação do bloco atual 800. No entanto, as informações de vetor de movimento de um bloco vizinho à esquerda ao bloco atual 800 podem não estar disponíveis.

[0164] Em um ou mais exemplos, para um primeiro bloco, em uma primeira imagem tendo uma ordem de codificação da esquerda para a direita, o codificador de vídeo 200 e o decodificador de vídeo 300 podem construir uma primeira lista de preditores de vetor de movimento. Como ilustrado na Figura 8A, uma primeira entrada na primeira lista de preditores de vetor de movimento baseia-se em informações de vetor de movimento de um bloco vizinho à esquerda do primeiro bloco (por exemplo, bloco 0 802). O bloco vizinho à esquerda do primeiro bloco pode ser um bloco à esquerda do primeiro bloco tendo um limite inferior que é o mesmo que um limite inferior do primeiro bloco. Para um segundo bloco, em uma segunda imagem tendo uma ordem de codificação da esquerda para a direita, o codificador de vídeo 200 e o decodificador de vídeo 300 podem construir uma segunda lista de preditores de vetor de movimento. Como ilustrado na Figura 8B, uma primeira entrada na segunda lista de preditores de vetor de movimento baseia-se em informações de vetor de movimento de um bloco vizinho à direita do segundo bloco (por exemplo,

bloco 0 818). O bloco vizinho à direita do segundo bloco pode ser um bloco à direita do segundo bloco tendo um limite inferior que é o mesmo que um limite inferior do segundo bloco.

[0165] Como ilustrado, o bloco 0 802 está à esquerda do bloco atual 800 da Figura 8A. Em alguns exemplos, o bloco 0 802 pode ser localizado à esquerda do bloco atual 800 da Figura 8A, tal que o limite inferior do bloco 0 802 é o mesmo que o limite inferior do bloco atual 800 da Figura 8A. Como ilustrado, o bloco 0 818 está à direita do bloco atual 800 da Figura 8B. Em alguns exemplos, o bloco 0 818 pode ser localizado à direita do bloco atual 800 da Figura 8B, tal que o limite inferior do bloco 0 818 é o mesmo que o limite inferior do bloco atual 800 da Figura 8B.

[0166] Para a primeira lista de preditores de vetor de movimento, as entradas na primeira lista de preditores de vetor de movimento podem ser como segue: primeira entrada é bloco vizinho à esquerda 0 802, segunda entrada é bloco vizinho superior 1 802, terceira entrada é bloco vizinho superior à direita 2 806, quarta entrada é bloco vizinho inferior à esquerda 3 808, e quinta entrada é bloco vizinho superior à esquerda 4 810. Em alguns exemplos, o bloco vizinho superior 1 802 pode estar acima do bloco atual 800 da Figura 8A e compartilha um limite direito com bloco atual 800 da Figura 8A.

[0167] Para a segunda lista de preditores de vetor de movimento, as entradas na segunda lista de preditores de vetor de movimento podem ser como segue:

primeira entrada é bloco vizinho à direita 0 818, segunda entrada é bloco vizinho superior 1 820, terceira entrada é bloco vizinho superior à esquerda 2 822, quarta entrada é bloco vizinho inferior à direita 3 824, e quinta entrada é bloco vizinho superior à direita 4 826. Em alguns exemplos, o bloco vizinho superior 1 820 pode estar acima do bloco atual 800 da Figura 8B e compartilhar um limite esquerdo com o bloco atual 800 da Figura 8B.

[0168] Em alguns exemplos, após verificar os blocos espacialmente vizinhos, o codificador de vídeo 200 e o decodificador de vídeo 300 podem verificar blocos colocalizados. Em um exemplo, o codificador de vídeo 200 e o decodificador de vídeo 300 podem verificar o bloco colocalizado central 5 812. Se informações de vetor de movimento para o bloco colocalizado central 5 812 estão disponíveis, o codificador de vídeo 200 e o decodificador de vídeo 300 podem adicionar as informações de vetor de movimento à lista de preditores de vetor de movimento e não adicionar informações de vetor de movimento de nenhum outro bloco colocalizado.

[0169] Se informações de vetor de movimento para o bloco colocalizado central 5 812 não estão disponíveis, o codificador de vídeo 200 e o decodificador de vídeo 300 podem verificar o bloco colocalizado inferior 6 814. Se informações de vetor de movimento para o bloco colocalizado inferior 6 814 estão disponíveis, o codificador de vídeo 200 e o decodificador de vídeo 300 podem adicionar as informações de vetor de movimento à lista de preditores de vetor de movimento e não adicionar informações de vetor de

movimento de nenhum outro bloco colocalizado. Como ilustrado, o bloco colocalizado inferior 6 814 pode ser um bloco que é inferior e mais próximo à direita do bloco atual 800 (por exemplo, bloco 6 814 está abaixo do bloco atual 800 da Figura 8A e o limite direito do bloco 6 814 é o mesmo que o limite direito do bloco atual 800 da Figura 8A). As coordenadas para o bloco 6 814 podem ser $(xColBot, yColBot)$, em que $xColBot = xCb + cbWidth - 1$, e $yColBot = yCb + cbHeight$, em que (xCb, yCb) são as coordenadas para o bloco atual 800 da Figura 8 A, $cbWidth$ é a largura do bloco atual 800 da Figura 8A, e $cbHeight$ é a altura do bloco atual 800 da Figura 8A.

[0170] Se informações de vetor de movimento para o bloco colocalizado inferior 6 814 não estão disponíveis, o codificador de vídeo 200 e o decodificador de vídeo 300 podem verificar o bloco colocalizado direito 7 816. Se informações de vetor de movimento para o bloco colocalizado direito 7 816 estão disponíveis, o codificador de vídeo 200 e o decodificador de vídeo 300 podem adicionar as informações de vetor de movimento à lista de preditores de vetor de movimento e não adicionar informações de vetor de movimento de nenhum outro bloco colocalizado. Como ilustrado, o bloco colocalizado direito 7 816 pode ser um bloco à direita e mais próximo à parte inferior do bloco atual 800 da Figura 8A (por exemplo, bloco 7 816 está à direita do bloco atual 800 da Figura 8A e o limite inferior do bloco 7 816 é o mesmo que o limite inferior do bloco atual 800 da Figura 8A). As coordenadas para o bloco 7 816 podem ser $(xColBr, yColBr)$, em que $xColBr = xCb + cbWidth$, e

$y_{ColBr} = y_{Cb} + cbHeight - 1$, em que (x_{Cb}, y_{Cb}) são as coordenadas para o bloco atual 800 da Figura 8A, $cbWidth$ é a largura do bloco atual 800 da Figura 8A, e $cbHeight$ é a altura do bloco atual 800 da Figura 8A.

[0171] Em outro exemplo, o codificador de vídeo 200 e o decodificador de vídeo 300 podem verificar o bloco colocalizado central 5 828. Se informações de vetor de movimento para o bloco colocalizado central 5 828 estão disponíveis, o codificador de vídeo 200 e o decodificador de vídeo 300 podem adicionar as informações de vetor de movimento à lista de preditores de vetor de movimento e não adicionar informações de vetor de movimento de nenhum outro bloco colocalizado.

[0172] Se informações de vetor de movimento para o bloco colocalizado central 5 828 não estão disponíveis, o codificador de vídeo 200 e o decodificador de vídeo 300 podem verificar o bloco colocalizado inferior 6 830. Se informações de vetor de movimento para o bloco colocalizado inferior 6 830 estão disponíveis, o codificador de vídeo 200 e o decodificador de vídeo 300 podem adicionar as informações de vetor de movimento à lista de preditores de vetor de movimento e não adicionar informações de vetor de movimento de nenhum outro bloco colocalizado. Como ilustrado, o bloco colocalizado inferior 6 830 pode ser um bloco que é inferior e mais próximo à esquerda do bloco atual 800 (por exemplo, o bloco 6 830 está abaixo do bloco atual 800 da Figura 8B e o limite esquerdo de bloco 6 830 é o mesmo que o limite esquerdo do bloco atual 800 da Figura 8B). As coordenadas para o bloco 6 830 podem ser $(x_{ColBot},$

y_{ColBot}), em que $x_{ColBot} = x_{Cb}$, e $y_{ColBot} = y_{Cb} + cbHeight$, em que (x_{Cb}, y_{Cb}) são as coordenadas para o bloco atual 800 da Figura 8B, $cbWidth$ é a largura do bloco atual 800 da Figura 8B, e $cbHeight$ é a altura do bloco atual 800 da Figura 8B.

[0173] Se informações de vetor de movimento para o bloco colocalizado inferior 6 830 não estão disponíveis, o codificador de vídeo 200 e o decodificador de vídeo 300 podem verificar o bloco colocalizado esquerdo 7 832. Se informações de vetor de movimento para o bloco colocalizado esquerdo 7 832 estão disponíveis, o codificador de vídeo 200 e o decodificador de vídeo 300 podem adicionar as informações de vetor de movimento à lista de preditores de vetor de movimento e não adicionar informações de vetor de movimento de nenhum outro bloco colocalizado. Como ilustrado, o bloco colocalizado esquerdo 7 832 pode ser um bloco à esquerda e mais próximo à parte inferior do bloco atual 800 da Figura 8B (por exemplo, bloco 7 832 está à esquerda do bloco atual 800 da Figura 8B e o limite inferior do bloco 7 832 é o mesmo que o limite inferior do bloco atual 800 da Figura 8B). As coordenadas para o bloco 7 832 podem ser (x_{ColBr}, y_{ColBr}) , em que $x_{ColBr} = x_{Cb} - 1$, e $y_{ColBr} = y_{Cb} + cbHeight - 1$, em que (x_{Cb}, y_{Cb}) são as coordenadas para o bloco atual 800 da Figura 8B, $cbWidth$ é a largura do bloco atual 800 da Figura 8B, e $cbHeight$ é a altura do bloco atual 800 da Figura 8B.

[0174] Deve-se compreender que as posições de blocos colocalizados espacialmente vizinhos nas Figuras 8A e 8B são um exemplo e não devem ser considerados

limitantes. Por exemplo, o bloco 802 pode estar localizado de forma geral na esquerda do bloco atual 800 da Figura 8A e o bloco 818 pode ser localizado de forma geral na direita do bloco atual 800 da Figura 8B. O mesmo se aplica a outros blocos colocalizados e espacialmente.

[0175] O exemplo acima descreve formas exemplificativas, em que o codificador de vídeo 200 e o decodificador de vídeo 300 ordenam as informações de vetor de movimento na lista de preditores de vetor de movimento. Em alguns exemplos, embora o codificador de vídeo 200 e o decodificador de vídeo 300 ordene as informações de vetor de movimento, se disponíveis, na lista de preditores de vetor de movimento em uma ordem específica, o codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para acessar as informações de vetor de movimento dos blocos vizinhos (por exemplo, blocos espacialmente vizinhos 0..4 nas Figuras 8A e 8B) em paralelo. Em outras palavras, o codificador de vídeo 200 e o decodificador de vídeo 300 podem verificar informações de vetor de movimento para uma pluralidade de (por exemplo, todos) blocos espacialmente vizinhos usados para construir a lista de preditores de vetor de movimento em paralelo.

[0176] Por exemplo, esta divulgação descreve técnicas para a inclusão de um candidato de preditor de vetor de movimento a uma lista de preditores de vetor de movimento. A seguir, o processo de construção da lista de preditores de vetor de movimento é descrito. O codificador de vídeo 200 e o decodificador de vídeo 300 podem verificar posições de candidatos de preditor de vetor de movimento

(por exemplo, todos os blocos espacialmente vizinhos) em ordem não condicional e independente da disponibilidade dos outros candidatos de preditor de vetor de movimento. Dessa forma, a verificação de candidato de preditor de vetor de movimento pode ser realizada em paralelo.

[0177] O codificador de vídeo 200 e o decodificador de vídeo 300 podem verificar um bloco vizinho se a posição do bloco vizinho estiver dentro do limite da imagem. Além disso, o codificador de vídeo 200 e o decodificador de vídeo 300 podem verificar um bloco vizinho que está em uma posição, tal que o bloco vizinho já tenha sido codificado ou decodificado. Além disso, o codificador de vídeo 200 e o decodificador de vídeo 300 podem determinar se informações de vetor de movimento estão disponíveis para o bloco vizinho. Se as condições exemplificativas acima forem satisfeitas, o codificador de vídeo 200 e o decodificador de vídeo 300 podem incluir as informações de vetor de movimento de um bloco vizinho na lista de preditores de vetor de movimento.

[0178] Como descrito acima, pode haver um tamanho máximo da lista de preditores de vetor de movimento. Por exemplo, candidatos de preditor de vetor de movimento pesquisados e carregados podem ser incluídos na lista de preditores de vetor de movimento até o tamanho efetivo da lista de preditores de vetor de movimento atingir um valor limite superior especificado, por exemplo, especificado por `MAX_MVP_LIST_SIZE` que define o tamanho máximo da lista de preditores de vetor de movimento. Em alguns exemplos, o valor de `MAX_MVP_LIST_SIZE` pode ser igual a 6.

[0179] No entanto, como descrito acima, ter o mesmo tamanho máximo de lista para todos os blocos sem consideração do tamanho do bloco pode resultar em listas de preditores de vetor de movimento que são desnecessariamente complexas para construir ou que não incluem candidatos suficientes. Por exemplo, como descrito acima, para blocos menores, para reduzir os ciclos de computação, pode ser benéfico ter listas de preditores de vetor de movimento de tamanhos menores, uma vez que candidatos adicionais podem não fornecer muitos ganhos de codificação. No entanto, para blocos maiores, para garantir uma boa diversidade nas informações de vetor de movimento, pode ser benéfico ter listas de preditores de vetor de movimento maiores.

[0180] Em um exemplo, o codificador de vídeo 200 e o decodificador de vídeo 300 podem ser configurados para determinar e/ou limitar o tamanho efetivo máximo da lista de preditores de vetor de movimento construída dependendo do tamanho do bloco codificado atual. Em alguns exemplos, para blocos que incluem $4 \times N$ ou $N \times 4$ número de pixels, em que $N \leq 8$, o tamanho efetivo de lista de MVP é igual a $\text{SMAL_BLOCK_MAX_MVP_LIST} < \text{MAX_MVP_LIST_SIZE}$. A variável $\text{SMAL_MAX_BLOCK_MAX_MVP_LIST}$ define o tamanho máximo da lista de preditores de vetor de movimento para blocos menores, em que blocos de menor tamanho podem ser blocos que possuem amostras menores do que ou iguais a $4 \times N$ ou $N \times 4$, e N é 8.

[0181] Para indicar se um bloco atual é um bloco pequeno, o codificador de vídeo 200 e/ou decodificador de vídeo 300 podem determinar um `smallBlockFlag` (por exemplo,

valor binário que indica se o bloco é pequeno ou não). A equação para determinar `smallBlockFlag` pode ser

```
smallBlockFlag = (cuw*cuh < SMALL_BLOCK_SIZE);
```

`SMALL_BLOCK_SIZE = 32`, em que `cuw` é a largura do bloco atual e `cuh` é a altura do bloco atual.

[0182] Em alguns exemplos, o critério de tamanhos de blocos pequenos `SMALL_BLOCK_SIZE` pode ser definido como sendo igual a 16 pixels. Por conseguinte, o exemplo de 32 para o tamanho de bloco pequeno é meramente um exemplo, e valores diferentes de 32 e 16 são contemplados por esta divulgação para o tamanho de bloco pequeno.

[0183] Em alguns exemplos, o valor de `MAX_MVP_LIST_SIZE` pode ser igual a 4. Ou seja, para blocos de pequeno tamanho, o tamanho máximo da lista de preditores de vetor de movimento pode ser quatro. Em alguns exemplos, o critério de tamanhos de blocos pequenos pode ser expresso através do valor de comprimentos das bordas de bloco:

```
smallBlockFlag = ((max(cuw, cuh) <= 8 && min(cuw,
cuh) < 8))
se (smallBlockFlag)
    use SMALL_BLOCK_MAX_MVP_LIST;
senão,
    use
    MAX_MVP_LIST_SIZE
```

[0184] Em outro exemplo, esta divulgação descreve técnicas para a sinalização de parâmetros da lista de candidatos de preditores de vetor de movimento. Por exemplo, o tamanho máximo possível da lista de preditores de vetor de movimento (`MAX_MVP_LIST_SIZE`) pode ser

especificado no lado do decodificador (por exemplo, decodificador de vídeo 300) como uma informação lateral (por exemplo, informação sinalizada do codificador de vídeo 200).

[0185] Em alguns exemplos, o tamanho máximo de lista de preditores de vetor de movimento efetivamente usado por um aplicativo (por exemplo, MAX_MVP_LIST_SIZE e/ou SMAL_BLOCK_MAX_MVP_LIST) para as condições de codificação atuais, por exemplo, para a sequência, imagem, fatia ou grupo de CUs atual, pode ser sinalizado por elementos de sintaxe de fluxo de bits, por exemplo, no cabeçalho de sequência/imagem/fatia ou outro nível.

[0186] Em alguns exemplos, parâmetros da dependência de tamanho de bloco, por exemplo, SMAL_BLOCK_SIZE, são especificados no lado do decodificador como informação lateral. Em outro exemplo, esse parâmetro pode ser sinalizado através do fluxo de bits.

[0187] A Figura 2 é um diagrama de blocos ilustrando um codificador de vídeo exemplificativo 200 que pode executar as técnicas desta divulgação. A Figura 2 é fornecida para fins de explicação e não deve ser considerada limitante das técnicas como amplamente exemplificado e descrito nesta divulgação. Para fins de explicação, esta divulgação descreve o codificador de vídeo 200 no contexto de padrões de codificação de vídeo, tais como o padrão de codificação de vídeo HEVC e o padrão de codificação de vídeo H.266/VVC em desenvolvimento. No entanto, as técnicas desta divulgação não são limitadas a esses padrões de codificação de vídeo, e são aplicáveis de

forma geral à codificação e decodificação de vídeo.

[0188] No exemplo da Figura 2, o codificador de vídeo 200 inclui memória de dados de vídeo 230, unidade de seleção de modo 202, unidade de geração residual 204, unidade de processamento de transformada 206, unidade de quantização 208, unidade de quantização inversa 210, unidade de processamento de transformada inversa 212, unidade de reconstrução 214, unidade de filtro 216, buffer de imagem decodificada (DPB) 218, e unidade de codificação por entropia 220.

[0189] Memória de dados de vídeo 230 pode armazenar dados de vídeo a serem codificados pelos componentes de codificador de vídeo 200. O codificador de vídeo 200 pode receber os dados de vídeo armazenados na memória de dados de vídeo 230, por exemplo, da fonte de vídeo 104 (Figura 1). DPB 218 pode atuar como uma memória de imagem de referência que armazena dados de vídeo de referência para uso na previsão de dados de vídeo subsequentes pelo codificador de vídeo 200. Memória de dados de vídeo 230 e DPB 218 podem ser formados por qualquer um de uma variedade de dispositivos de memória, tais como 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 230 e DPB 218 podem ser providos pelo mesmo dispositivo de memória ou dispositivos de memória separados. Em vários exemplos, memória de dados de vídeo 230 podem ser on-chip com outros componentes de codificador de vídeo 200, como

ilustrado, ou off-chip em relação àqueles componentes.

[0190] Nesta divulgação, referência à memória de dados de vídeo 230 não deve ser interpretada como sendo limitada à memória interna do codificador de vídeo 200, a menos que especificamente assim descrito, ou memória externa ao codificador de vídeo 200, a menos que especificamente assim descrito. Em vez disso, referência à memória de dados de vídeo 230 deve ser entendida como memória que armazena dados de vídeo que o codificador de vídeo 200 recebe para codificar (por exemplo, dados de vídeo para um bloco atual a ser codificado). A memória 106 da Figura 1 pode também fornecer armazenamento temporário de saídas das diversas unidades de codificador de vídeo 200.

[0191] As várias unidades da Figura 2 são ilustradas para ajudar no entendimento das operações realizadas pelo codificador de vídeo 200. As unidades podem ser implementadas como circuitos de função fixa, circuitos programáveis ou uma combinação dos mesmos. Os circuitos de função fixa se configuram como circuitos que proporcionam uma funcionalidade particular e são predefinidos nas operações que podem ser realizadas. Os circuitos programáveis são os circuitos que podem ser programados para realizar várias tarefas, e fornecem funcionalidade flexível nas operações que podem ser realizadas. Por exemplo, circuitos programáveis podem executar software ou firmware que fazem com que os circuitos programáveis operem da maneira definida por instruções de software ou firmware. Os circuitos de função fixa podem executar instruções de

software (por exemplo, para receber parâmetros ou gerar parâmetros), mas os tipos de operações que os circuitos de função fixa realizam são de forma geral imutáveis. Em alguns exemplos, as uma ou mais das unidades podem ser blocos de circuito distintos (função fixa ou programáveis) e, em alguns exemplos, as uma ou mais unidades podem ser circuitos integrados.

[0192] O codificador de vídeo 200 pode incluir unidades lógicas aritméticas (ALUs), unidades de função elementar (EFUs), circuitos digitais, circuitos analógicos e/ou núcleos programáveis formados a partir de circuitos programáveis. Em exemplos em que as operações do codificador de vídeo 200 são realizadas utilizando software executado pelos circuitos programáveis, a memória 106 (Figura 1) pode armazenar o código de objeto do software que o codificador de vídeo 200 recebe e executa, ou outra memória dentro do codificador de vídeo 200 (não mostrado) pode armazenar tais instruções.

[0193] A memória de dados de vídeo 230 é configurada para armazenar dados de vídeo recebidos. O codificador de vídeo 200 pode recuperar uma imagem dos dados de vídeo da memória de dados de vídeo 230 e prover os dados de vídeo à unidade de geração residual 204 e unidade de seleção de modo 202. Dados de vídeo na memória de dados de vídeo 230 podem ser dados de vídeo brutos que devem ser codificados.

[0194] A unidade de seleção de modo 202 inclui uma unidade de estimativa de movimento 222, unidade de compensação de movimento 224, e uma unidade de

intrapredição 226. A unidade de seleção de modo 202 pode incluir unidades funcionais adicionais para realizar predição de vídeo de acordo com outros modos de predição. Como exemplos, a unidade de seleção de modo 202 pode incluir uma unidade de paleta, uma unidade de cópia intrabloco (que pode ser parte da unidade de estimativa de movimento 222 e/ou unidade de compensação de movimento 224), uma unidade afim, uma unidade de modelo linear (LM), ou semelhantes.

[0195] A unidade de seleção de modo 202 de forma geral coordenadas múltiplos passos de codificação para testar combinações de parâmetros de codificação e valores de distorção de taxa resultantes para tais combinações. Os parâmetros de codificação podem incluir particionamento de CTUs em CUs, modos de predição para as CUs, tipos de transformada para dados residuais das CUs, parâmetros de quantização para dados residuais das CUs, e assim por diante. A unidade de seleção de modo 202 pode, em última análise, selecionar a combinação de parâmetros de codificação tendo valores de distorção de taxa que são melhores do que as outras combinações testadas.

[0196] O codificador de vídeo 200 pode particionar uma imagem recuperada da memória de dados de vídeo 230 em uma série de CTUs, e encapsular uma ou mais CTUs dentro de uma fatia. A unidade de seleção de modo 202 pode particionar uma CTU da imagem de acordo com uma estrutura de árvore, tal como a estrutura QTBT ou a estrutura de árvore quádrupla de HEVC descrita acima. Como descrito acima, o codificador de vídeo 200 pode formar uma

ou mais CUs a partir do particionamento de uma CTU de acordo com a estrutura de árvore. Tal CU também pode ser referida de forma geral como um "bloco de vídeo" ou "bloco."

[0197] Em geral, a unidade de seleção de modo 202 também controla seus componentes (por exemplo, unidade de estimativa de movimento 222, unidade de compensação de movimento 224 e unidade de intrapredição 226) para gerar um bloco de predição para um bloco atual (por exemplo, uma CU atual, ou em HEVC, a porção sobreposta da PU e da TU). Para interpredição de um bloco atual, a unidade de estimativa de movimento 222 pode realizar uma pesquisa de movimento para identificar um ou mais blocos de referência correspondentes em uma ou mais imagens de referência (por exemplo, uma ou imagens mais previamente codificadas armazenadas em DPB 218). In particular, a unidade de estimativa de movimento 222 pode calcular um valor representativo de quão similar um bloco de referência potencial é do bloco atual, por exemplo, de acordo com a soma da diferença absoluta (SAD), soma de diferenças quadradas (SSD), diferença média absoluta (MAD), diferenças quadradas médias (MSD), ou semelhantes. A unidade de estimativa de movimento 222 pode de forma geral realizar esses cálculos usando diferenças de amostra por amostra entre o bloco atual e o bloco de referência sendo considerado. A unidade de estimativa de movimento 222 pode identificar um bloco de referência tendo um valor mais baixo resultante desses cálculos, indicando um bloco de referência que mais corresponde ao bloco atual.

[0198] A unidade de estimativa de movimento 222

pode formar um ou mais vetores de movimento (MVs) que definem as posições dos blocos de referência nas imagens de referência em relação à posição do bloco atual em uma imagem atual. A unidade de estimativa de movimento 222 pode, então, fornecer os vetores de movimento à unidade de compensação de movimento 224. Por exemplo, para interpredição unidirecional, a unidade de estimativa de movimento 222 pode prover um único vetor de movimento, enquanto que, para interpredição bidirecional, a unidade de estimativa de movimento 222 pode prover dois vetores de movimentos. A unidade de compensação de movimento 224 pode, então, gerar um bloco de predição usando os vetores de movimento. Por exemplo, a unidade de compensação de movimento 224 pode recuperar dados do bloco de referência usando o vetor de movimento. Como outro exemplo, se o vetor de movimento tem precisão de amostra fracionária, a unidade de compensação de movimento 224 pode interpolar valores para o bloco de predição de acordo com um ou mais filtros de interpolação. Além disso, para interpredição bidirecional, a unidade de compensação de movimento 224 pode recuperar dados para dois blocos de referência identificados pelos respectivos vetores de movimento e combinar os dados recuperados, por exemplo, através de média amostra por amostra ou média ponderada.

[0199] De acordo com as técnicas desta divulgação, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem realizar técnicas de interpredição e predição de vetor de movimento descritas nesta divulgação. Por exemplo, como será

explicado em mais detalhes abaixo, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem ser configuradas para construir uma lista de preditores de vetor de movimento utilizando as técnicas exemplificativas descritas nesta divulgação.

[0200] Como descrito acima, em alguns exemplos, a lista de preditores de vetor de movimento para um bloco pode ser com base na ordem de codificação da imagem que inclui o bloco e os blocos espacialmente vizinhos e/ou blocos colocalizados. Por exemplo, em um exemplo, os cinco blocos espacialmente vizinhos podem ser aqueles ilustrados na Figura 8A para uma ordem de codificação da esquerda para a direita, e os cinco blocos espacialmente vizinhos podem ser aqueles ilustrados na Figura 8B para uma ordem de codificação da direita para a esquerda.

[0201] Por exemplo, para um primeiro bloco, em uma primeira imagem tendo uma ordem de codificação da esquerda para a direita, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem construir uma primeira lista de preditores de vetor de movimento. Uma primeira entrada na primeira lista de preditores de vetor de movimento podem ser com base em informações de vetor de movimento de um bloco vizinho à esquerda do primeiro bloco (por exemplo, bloco 0 802 da Figura 8A). Para um segundo bloco, em uma segunda imagem tendo uma ordem de codificação da direita para a esquerda, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem construir uma segunda lista de preditores de vetor de movimento. Uma primeira

entrada na segunda lista de preditores de vetor de movimento pode ser com base em informações de vetor de movimento de um bloco vizinho à direita do segundo bloco (por exemplo, bloco 0 818 da Figura 8B).

[0202] A unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem verificar informações de vetor de movimento armazenadas na memória de dados de vídeo 230 ou buffer de imagem decodificada 218 (como dois exemplos) para uma pluralidade, incluindo todos, os blocos espacialmente vizinhos em paralelo. Em outras palavras, se a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 verificarem informações de vetor de movimento para um bloco espacialmente vizinho específico pode não ser dependente de informações de vetor de movimento de qualquer outro bloco.

[0203] Com base na verificação das informações de vetor de movimento dos blocos espacialmente vizinhos, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem construir uma lista de preditores de vetor de movimento tendo informações de vetor de movimento em uma ordem com base na disponibilidade das informações de vetor de movimento. A ordem das informações de vetor de movimento na lista de preditores de vetor de movimento para uma ordem de codificação da esquerda para a direita pode ser: informações de vetor de movimento do bloco vizinho à esquerda (por exemplo, bloco 0 802 da Figura 8A), informações de vetor de movimento de um bloco vizinho acima (por exemplo, bloco 1 804 da Figura 8A), informações de vetor de movimento de um bloco vizinho acima

à direita (por exemplo, bloco 2 806 da Figura 8A), informações de vetor de movimento de um bloco vizinho abaixo à esquerda (por exemplo, bloco 3 808 da Figura 8A), e informações de vetor de movimento de um bloco vizinho acima à esquerda (por exemplo, bloco 4 810 da Figura 8A). A ordem das informações de vetor de movimento na lista de preditores de vetor de movimento para uma ordem de codificação da direita para a esquerda pode ser: informações de vetor de movimento do bloco vizinho à direita (por exemplo, bloco 0 818 da Figura 8B), informações de vetor de movimento de um bloco vizinho acima (por exemplo, bloco 1 820 da Figura 8B), informações de vetor de movimento de um bloco vizinho acima à esquerda (por exemplo, bloco 2 822 da Figura 8B), informações de vetor de movimento de um bloco vizinho abaixo à direita (por exemplo, bloco 3 824 da Figura 8B), e informações de vetor de movimento de um bloco vizinho acima à direita (por exemplo, bloco 4 826 da Figura 8B).

[0204] O ordenamento acima baseia-se na disponibilidade das informações de vetor de movimento. Se informações de vetor de movimento não estão disponíveis para um bloco particular, então, aquelas informações de vetor de movimento não são adicionadas à lista de preditores de vetor de movimento. Em alguns exemplos, as próximas informações de vetor de movimento disponíveis podem ser incluídas no espaço reservado para o bloco para o qual não havia nenhuma informação de vetor de movimento. Por exemplo, informações de vetor de movimento para o bloco 1 804 da Figura 8A podem ser a segunda entrada na lista de

preditores de vetor de movimento, e informações de vetor de movimento para o bloco 2 806 da Figura 8A podem ser a terceira entrada na lista de preditores de vetor de movimento. Se informações de vetor de movimento para o bloco 1 804 da Figura 8A não estão disponíveis e informações de vetor de movimento para o bloco 2 806 da Figura A estão disponíveis, então, as informações de vetor de movimento para o bloco 2 806 podem estar na segunda entrada da lista de preditores de vetor de movimento.

[0205] No exemplo acima, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 determinaram uma primeira lista de preditores de vetor de movimento e uma segunda lista de preditores de vetor de movimento com base em blocos espacialmente vizinhos. Em alguns exemplos, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem incluir informações de vetor de movimento, armazenadas na memória de dados de vídeo 230 ou buffer de imagem decodificada 218 (como dois exemplos), de blocos colocalizados na primeira lista de preditores de vetor de movimento e na segunda lista de preditores de vetor de movimento.

[0206] Como um exemplo, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem determinar se informações de vetor de movimento para um bloco colocalizado central localizado em uma imagem, que não a primeira imagem, (por exemplo, bloco 5 812 da Figura 8A) estão disponíveis antes de determinar se informações de vetor de movimento em qualquer outro bloco colocalizado localizado na imagem, que não a primeira

imagem, estão disponíveis, e com base nas informações de vetor de movimento para o bloco colocalizado central estando disponíveis, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem adicionar as informações de vetor de movimento para o bloco colocalizado central à primeira lista de preditores de vetor de movimento.

[0207] Em alguns exemplos, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem determinar que informações de vetor de movimento para um bloco colocalizado central localizado em uma imagem, que não a primeira imagem, não estão disponíveis. Em tais exemplos, após determinar que as informações de vetor de movimento para o bloco colocalizado central não estão disponíveis, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem determinar se informações de vetor de movimento para um bloco colocalizado inferior localizado na imagem, que não a primeira imagem, estão disponíveis (por exemplo, bloco 6 814 da Figura 8A), e com base nas informações de vetor de movimento para o bloco colocalizado inferior estando disponíveis, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem adicionar as informações de vetor de movimento para o bloco colocalizado inferior à primeira lista de preditores de vetor de movimento.

[0208] Em alguns exemplos, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem determinar que informações de vetor de

movimento para um bloco colocalizado central localizado em uma imagem, que não a primeira imagem, não estão disponíveis. Em alguns casos, após determinar que as informações de vetor de movimento para o bloco colocalizado central não estão disponíveis, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem determinar que informações de vetor de movimento para um bloco colocalizado inferior localizado na imagem, que não a primeira imagem, não estão disponíveis. Nesses casos, após determinar que as informações de vetor de movimento para o bloco colocalizado inferior não estão disponíveis, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem determinar se informações de vetor de movimento para um bloco colocalizado direito localizado na imagem, que não a primeira imagem, estão disponíveis (por exemplo, bloco 7 816 da Figura 8A), e com base nas informações de vetor de movimento para o bloco colocalizado direito estando disponíveis, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem adicionar as informações de vetor de movimento para o bloco colocalizado direito à primeira lista de preditores de vetor de movimento.

[0209] A unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem realizar operações similares para adicionar informações de vetor de movimento para blocos colocalizados para construir a segunda lista de preditores de vetor de movimento. No entanto, para a segunda lista de preditores de vetor de

movimento, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem adicionar informações de vetor de movimento com base na disponibilidade do bloco colocalizado central 5 828, bloco colocalizado inferior 6 830, e bloco colocalizado esquerdo 7 832 da Figura 8B.

[0210] Na construção de listas de preditores de vetor de movimento, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem construir as listas de preditores de vetor de movimento com base em tamanho do bloco, em alguns exemplos. Por exemplo, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem construir a primeira lista de preditores de vetor de movimento, tendo um primeiro tamanho máximo, com base no primeiro bloco tendo um tamanho maior do que um tamanho limite. A unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem construir uma terceira lista de preditores de vetor de movimento, tendo um segundo tamanho máximo que é inferior ao primeiro tamanho máximo, para um terceiro bloco na primeira imagem com base no terceiro bloco tendo um tamanho inferior ao tamanho limite. Como um exemplo, o primeiro tamanho máximo é igual a seis, o segundo tamanho máximo é igual a quatro, e o tamanho limite é igual a $N \times 4$ ou $4 \times N$, em que N é menor do que ou igual a oito.

[0211] A unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem ser configuradas para codificar o primeiro bloco e o segundo

bloco. Por exemplo, a unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem determinar um bloco de predição para o primeiro bloco e determinar um vetor de movimento para o primeiro bloco que identifica o bloco de predição. A unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem determinar uma entrada na primeira lista de preditores de vetor de movimento com base no vetor de movimento. A unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem fazer com que a unidade de codificação por entropia 220 sinalize informações indicativas da entrada na primeira lista de preditores de vetor de movimento. Além disso, a unidade de geração residual 204 pode determinar dados residuais representando uma diferença entre o primeiro bloco e o bloco de predição. A unidade de codificação por entropia 220 pode sinalizar informações indicativas dos dados residuais. A unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem realizar operações similares para codificar o segundo bloco usando a segunda lista de preditores de vetor de movimento.

[0212] Como outro exemplo, para intrapredição, ou codificação por intrapredição, a unidade de intrapredição 226 pode gerar o bloco de predição a partir de amostras vizinhos ao bloco atual. Por exemplo, para modos direcionais, a unidade de intrapredição 226 pode de forma geral matematicamente combinar valores de amostras vizinhos e popular esses valores calculados na direção definida através do bloco atual para produzir o bloco de predição.

Como outro exemplo, para o modo DC, a unidade de intrapredição 226 pode calcular uma média das amostras vizinhos ao bloco atual e gerar o bloco de predição para incluir essa média resultante para cada amostra do bloco de predição.

[0213] A unidade de seleção de modo 202 fornece o bloco de predição à unidade de geração residual 204. A unidade de geração residual 204 recebe uma versão não codificada bruta do bloco atual da memória de dados de vídeo 230 e o bloco de predição da unidade de seleção de modo 202. A unidade de geração residual 204 calcula diferenças de amostra por amostra entre o bloco atual e o bloco de predição. As diferenças de amostra por amostra resultantes definem um bloco residual para o bloco atual. Em alguns exemplos, a unidade de geração residual 204 pode também determinar diferenças entre valores de amostra no bloco residual para gerar um bloco residual usando modulação de código de pulso diferencial residual (RDPCM). Em alguns exemplos, a unidade de geração residual 204 pode ser formada usando um ou mais circuitos subtratores que realizam subtração binária.

[0214] Em exemplos em que a unidade de seleção de modo 202 particiona CUs em PUs, cada PU pode ser associada a uma unidade de predição de luma e unidades de predição de croma correspondentes. O codificador de vídeo 200 e o decodificador de vídeo 300 podem suportar PUs tendo vários tamanhos. Como indicado acima, o tamanho de uma CU pode se referir ao tamanho do bloco de codificação de luma da CU e o tamanho de uma PU pode se referir ao tamanho de uma

unidade de predição de luma da PU. Assumindo que o tamanho de uma CU particular é $2N \times 2N$, o codificador de vídeo 200 pode suportar tamanhos de PU de $2N \times 2N$ ou $N \times N$ para intrapredição, e tamanhos de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$, ou similar para interpredição. O codificador de vídeo 200 e o decodificador de vídeo 300 podem também suportar particionamento assimétrico para tamanhos de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$, e $nR \times 2N$ para interpredição.

[0215] Em exemplos em que a unidade de seleção de modo 202 não mais particiona a CU em PUs, cada CU pode ser associada a um bloco de codificação de luma e blocos de codificação de croma correspondentes. Como acima, o tamanho de uma CU pode se referir ao tamanho do bloco de codificação de luma da CU. O codificador de vídeo 200 e o decodificador de vídeo 300 podem suportar tamanhos de CU de $2N \times 2N$, $2N \times N$ ou $N \times 2N$.

[0216] Para outras técnicas de codificação de vídeo, tais como uma codificação de modo de cópia intrabloco, uma codificação de modo afim, e codificação de modo de modelo linear (LM), como alguns exemplos, unidade de seleção de modo 202, via unidades respectivas associadas com as técnicas de codificação, gera um bloco de predição para o bloco atual sendo codificado. Em alguns exemplos, tal como a codificação de modo paleta, a unidade de seleção de modo 202 pode não gerar um bloco de predição, e, em vez disso, gerar elementos de sintaxe que indicam a maneira de reconstruir o bloco com base em uma paleta selecionada. Em tais modos, a unidade de seleção de modo 202 pode prover esses elementos de sintaxe à unidade de codificação por

entropia 220 a ser codificada.

[0217] Como descrito acima, a unidade de geração residual 204 recebe os dados de vídeo para o bloco atual e o bloco de predição correspondente. A unidade de geração residual 204, então, gera um bloco residual para o bloco atual. Para gerar o bloco residual, a unidade de geração residual 204 calcula diferenças de amostra por amostra entre o bloco de predição e o bloco atual.

[0218] A unidade de processamento de transformada 206 aplica uma ou mais transformadas ao bloco residual para gerar um bloco de coeficientes de transformada (referidos aqui como um "bloco de coeficiente de transformada"). A unidade de processamento de transformada 206 pode aplicar várias transformadas a um bloco residual para formar o bloco de coeficiente de transformada. Por exemplo, a unidade de processamento de transformada 206 pode aplicar uma transformada discreta de cosseno (DCT), uma transformada direcional, uma transformada Karhunen-Loeve (KLT), ou uma transformada conceitualmente similar a um bloco residual. Em alguns exemplos, a unidade de processamento de transformada 206 pode realizar múltiplas transformadas a um bloco residual, por exemplo, uma transformada primária e a transformada secundária, tal como a transformada rotacional. Em alguns exemplos, a unidade de processamento de transformada 206 não aplica transformadas a um bloco residual.

[0219] A unidade de quantização 208 pode quantizar os coeficientes de transformada em um bloco de coeficiente de transformada, para produzir um bloco de

coeficiente de transformada quantizado. A unidade de quantização 208 pode quantizar coeficientes de transformada de um bloco de coeficiente de transformada de acordo com um valor de parâmetro de quantização (QP) associado com o bloco atual. O codificador de vídeo 200 (por exemplo, via unidade de seleção de modo 202) pode ajustar o grau de quantização aplicado aos blocos de coeficiente associados com o bloco atual ajustando o valor de QP associado com a CU. A quantização pode introduzir perda de informações e, dessa forma, coeficientes de transformada quantizados podem ter precisão menor do que os coeficientes de transformada originais produzidos pela unidade de processamento de transformada 206.

[0220] A unidade de quantização inversa 210 e a unidade de processamento de transformada inversa 212 podem aplicar quantização inversa e transformadas inversas a um bloco de coeficiente de transformada quantizado, respectivamente, para reconstruir um bloco residual do bloco de coeficiente de transformada. A unidade de reconstrução 214 pode produzir um bloco reconstruído correspondente ao bloco atual (embora potencialmente com algum grau de distorção) com base no bloco residual reconstruído e um bloco de predição gerado pela unidade de seleção de modo 202. Por exemplo, a unidade de reconstrução 214 pode adicionar amostras do bloco residual reconstruído a amostras correspondentes do bloco de predição gerado pela unidade de seleção de modo 202 para produzir o bloco reconstruído.

[0221] A unidade de filtro 216 pode realizar uma

ou mais operações de filtro em blocos reconstruídos. Por exemplo, a unidade de filtro 216 pode realizar operações de desbloqueio para reduzir artefatos de bloqueio ao longo das bordas de CUs. Operações da unidade de filtro 216 podem ser puladas, em alguns exemplos.

[0222] O codificador de vídeo 200 armazena blocos reconstruídos em DPB 218. Por exemplo, em exemplos em que operações da unidade de filtro 216 não são necessárias, a unidade de reconstrução 214 pode armazenar blocos reconstruídos em DPB 218. Em exemplos em que operações da unidade de filtro 216 são necessárias, a unidade de filtro 216 pode armazenar os blocos reconstruídos filtrados em DPB 218. A unidade de estimativa de movimento 222 e a unidade de compensação de movimento 224 podem recuperar uma imagem de referência de DPB 218, formada de blocos reconstruídos (e potencialmente filtrados), a blocos de interpredição de imagens subsequentemente codificados. Além disso, a unidade de intrapredição 226 pode usar blocos reconstruídos em DPB 218 de uma imagem atual para intrapredição de outros blocos na imagem atual.

[0223] Em geral, a unidade de codificação por entropia 220 pode codificar por entropia elementos de sintaxe recebidos de outros componentes funcionais de codificador de vídeo 200. Por exemplo, a unidade de codificação por entropia 220 pode codificar por entropia blocos de coeficiente de transformada quantizados a partir da unidade de quantização 208. Como outro exemplo, a unidade de codificação por entropia 220 pode codificar por entropia elementos de sintaxe de predição (por exemplo,

informações de movimento para interpredição ou informações de intramodo para intrapredição) da unidade de seleção de modo 202. A unidade de codificação por entropia 220 pode realizar uma ou mais operações de codificação por entropia nos elementos de sintaxe, que são outro exemplo de dados de vídeo, para gerar dados codificados por entropia. Por exemplo, a unidade de codificação por entropia 220 pode realizar uma operação de codificação de comprimento variável adaptativa ao contexto (CAVLC), uma operação de CABAC, uma operação de codificação de comprimento variável (V2V), uma operação de codificação aritmética adaptável ao contexto baseada em sintaxe (SBAC), uma operação de codificação por Entropia de Particionamento de Intervalo de Probabilidade (PIPE), uma operação de codificação Exponencial-Golomb, ou outro tipo de operação de codificação por entropia nos dados. Em alguns exemplos, a unidade de codificação por entropia 220 pode operar em modo bypass, em que elementos de sintaxe não são codificados por entropia.

[0224] O codificador de vídeo 200 pode gerar um fluxo de bits que inclui os elementos de sintaxe codificados por entropia necessários para reconstruir blocos de uma fatia ou imagem. Em particular, a unidade de codificação por entropia 220 pode gerar o fluxo de bits.

[0225] As operações descritas acima são descritas com relação a um bloco. Tal descrição deve ser compreendida como sendo operações para um bloco de codificação de luma e/ou blocos de codificação de croma. Como descrito acima, em alguns exemplos, o bloco de codificação de luma e blocos

de codificação de croma são componentes luma e croma de uma CU. Em alguns exemplos, o bloco de codificação de luma e os blocos de codificação de croma são componentes luma e croma de uma PU.

[0226] Em alguns exemplos, as operações realizadas com respeito a um bloco de codificação de luma não precisam ser repetidas para os blocos de codificação de croma. Como um exemplo, operações para identificar um vetor de movimento (MV) e imagem de referência para um bloco de codificação de luma não precisam ser repetidas para identificar um MV e imagem de referência para os blocos de croma. Em vez disso, o MV para o bloco de codificação de luma pode ser escalado para determinar o MV para os blocos de croma, e a imagem de referência pode ser a mesma. Como outro exemplo, o processo de intrapredição pode ser o mesmo para os blocos de codificação de luma e os blocos de codificação de croma.

[0227] A Figura 3 é um diagrama de blocos ilustrando um decodificador de vídeo exemplificativo 300 que pode executar as técnicas desta divulgação. A Figura 3 é fornecida para fins de explicação e não é limitante das técnicas como amplamente exemplificado e descrito nesta divulgação. Para fins de explicação, esta divulgação descreve o decodificador de vídeo 300 de acordo com as técnicas de H.266/VVC, JEM, e HEVC. No entanto, as técnicas desta divulgação podem ser realizadas por dispositivos de codificação de vídeo que são configurados para outros padrões de codificação de vídeo.

[0228] No exemplo da Figura 3, o decodificador de

vídeo 300 inclui memória buffer de imagem codificada (CPB) 320, unidade de decodificação por entropia 302, unidade de processamento de predição 304, unidade de quantização inversa 306, unidade de processamento de transformada inversa 308, unidade de reconstrução 310, unidade de filtro 312, e buffer de imagem decodificada (DPB) 314. A unidade de processamento de predição 304 inclui unidade de compensação de movimento 316 e unidade de intrapredição 318. A unidade de processamento de predição 304 pode incluir unidades de adição para realizar predição de acordo com outros modos de predição. Como exemplos, a unidade de processamento de predição 304 pode incluir uma unidade de paleta, uma unidade de cópia intrabloco (que pode fazer parte da unidade de compensação de movimento 316), uma unidade afim, uma unidade de modelo linear (LM), ou semelhantes. Em outros exemplos, o decodificador de vídeo 300 pode incluir mais, menos ou diferentes componentes funcionais.

[0229] A memória CPB 320 pode armazenar dados de vídeo, tais como um fluxo de bits de vídeo codificado, a ser decodificado pelos componentes de decodificador de vídeo 300. Os dados de vídeo armazenados na memória CPB 320 podem ser obtidos, por exemplo, do meio legível por computador 110 (Figura 1). A memória CPB 320 pode incluir um CPB que armazena dados de vídeo codificados (por exemplo, elementos de sintaxe) de um fluxo de bits de vídeo codificado. Além disso, a memória CPB 320 pode armazenar dados de vídeo que não elementos de sintaxe de uma imagem codificada, tais como dados temporários representando

saídas das várias unidades de decodificador de vídeo 300. DPB 314 de forma geral armazena imagens decodificadas, cujo decodificador de vídeo 300 pode gerar e/ou usar como dados de vídeo de referência ao decodificar dados ou imagens subsequentes do fluxo de bits de vídeo codificado. A memória CPB 320 e DPB 314 podem ser formadas por qualquer um de uma variedade de dispositivos de memória, tais como 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 CPB 320 e DPB 314 podem ser fornecidas pelo mesmo dispositivo de memória ou dispositivos de memória separados. Em vários exemplos, a memória CPB 320 pode ser on-chip com outros componentes do decodificador de vídeo 300, ou off-chip em relação àqueles componentes.

[0230] Adicionalmente ou alternativamente, em alguns exemplos, o decodificador de vídeo 300 pode recuperar dados de vídeo codificados da memória 120 (Figura 1). Ou seja, a memória 120 pode armazenar dados como discutido acima com a memória CPB 320. Da mesma forma, a memória 120 pode armazenar instruções a serem executadas pelo decodificador de vídeo 300, quando parte ou toda a funcionalidade do decodificador de vídeo 300 é implementada em software a ser executado por circuitos de processamento de decodificador de vídeo 300.

[0231] As várias unidades mostradas na Figura 3 são ilustradas para ajudar no entendimento das operações realizadas pelo decodificador de vídeo 300. As unidades podem ser implementadas como circuitos de função fixa,

circuitos programáveis ou uma combinação dos mesmos. Semelhante à Figura 2, os circuitos de função fixa se referem a circuitos que proporcionam uma funcionalidade particular, e são predefinidos nas operações que podem ser realizadas. Os circuitos programáveis são os circuitos que podem ser programados para realizar várias tarefas, e fornecem funcionalidade flexível nas operações que podem ser realizadas. Por exemplo, circuitos programáveis podem executar software ou firmware que fazem com que os circuitos programáveis operem da maneira definida por instruções de software ou firmware. Os circuitos de função fixa podem executar instruções de software (por exemplo, para receber parâmetros ou gerar parâmetros), mas os tipos de operações que os circuitos de função fixa realizam são de forma geral imutáveis. Em alguns exemplos, as uma ou mais unidades podem ser blocos de circuitos distintos (função fixa ou programáveis) e, em alguns exemplos, as uma ou mais unidades podem ser circuitos integrados.

[0232] O decodificador de vídeo 300 pode incluir ALUs, EFUs, circuitos digitais, circuitos analógicos e/ou núcleos programáveis formados a partir de circuitos programáveis. Em exemplos em que as operações de decodificador de vídeo 300 são realizadas por software executado nos circuitos programáveis, a memória on-chip ou off-chip pode armazenar instruções (por exemplo, código do objeto) do software que o decodificador de vídeo 300 recebe e executa.

[0233] A unidade de decodificação por entropia 302 pode receber dados de vídeo codificados a partir do CPB

e decodificar por entropia os dados de vídeo para reproduzir elementos de sintaxe. A unidade de processamento de predição 304, unidade de quantização inversa 306, unidade de processamento de transformada inversa 308, unidade de reconstrução 310 e unidade de filtro 312 podem gerar dados de vídeo decodificados com base nos elementos de sintaxe extraídos do fluxo de bits.

[0234] Em geral, o decodificador de vídeo 300 reconstrói uma imagem bloco por bloco. O decodificador de vídeo 300 pode realizar uma operação de reconstrução em cada bloco individualmente (em que o bloco atualmente sendo reconstruído, isto é, decodificado, pode ser referido como um "bloco atual").

[0235] A unidade de decodificação por entropia 302 pode decodificar por entropia elementos de sintaxe definindo coeficientes de transformada quantizados de um bloco de coeficiente de transformada quantizado, bem como informações de transformada, tais como indicação(ões) de parâmetro de quantização (QP) e/ou modo de transformada. A unidade de quantização inversa 306 pode usar o QP associado com o bloco de coeficiente de transformada quantizado para determinar um grau de quantização e, da mesma forma, um grau de quantização inversa para a unidade de quantização inversa 306 aplicar. A unidade de quantização inversa 306 pode, por exemplo, realizar uma operação de deslocamento à esquerda bit a bit para quantização inversa dos coeficientes de transformada quantizados. A unidade de quantização inversa 306 pode assim formar um bloco de coeficientes de transformada incluindo coeficientes de

transformada.

[0236] Após a unidade de quantização inversa 306 formar o bloco de coeficientes de transformada, a unidade de processamento de transformada inversa 308 pode aplicar uma ou mais transformadas inversas ao bloco de coeficiente de transformada para gerar um bloco residual associado com o bloco atual. Por exemplo, a unidade de processamento de transformada inversa 308 pode aplicar uma DCT inversa, uma transformada inteira inversa, uma transformada de Karhunen-Loeve inversa (KLT), uma transformada rotacional inversa, uma transformada direcional inversa ou outra transformada inversa ao bloco de coeficientes.

[0237] Além disso, a unidade de processamento de predição 304 gera um bloco de predição de acordo com elementos de sintaxe de informações de predição que foram decodificados por entropia pela unidade de decodificação por entropia 302. Por exemplo, se os elementos de sintaxe de informações de predição indicam que o bloco atual é interpreditado, a unidade de compensação de movimento 316 pode gerar o bloco de predição. Nesse caso, os elementos de sintaxe de informações de predição podem indicar uma imagem de referência em DPB 314 a partir da qual recuperar um bloco de referência, bem como um vetor de movimento que identifica uma localização do bloco de referência na imagem de referência em relação à localização do bloco atual na imagem atual. A unidade de compensação de movimento 316 pode, de forma geral, realizar o processo de interpredição de maneira que semelhante ao descrito com respeito à unidade de compensação de movimento 224 (Figura 2).

[0238] De acordo com as técnicas desta divulgação, a unidade de compensação de movimento 316 pode realizar técnicas de interpredição e predição de vetor de movimento ao codificar um bloco descrito nesta divulgação. Por exemplo, como será explicado com mais detalhes abaixo, a unidade de compensação de movimento 316 pode ser configurada para construir uma lista de preditores de vetor de movimento utilizando as técnicas exemplificativas descritas nesta divulgação.

[0239] Como descrito acima, em alguns exemplos, a lista de preditores de vetor de movimento para um bloco pode ser com base na ordem de codificação da imagem que inclui o bloco e os blocos espacialmente vizinhos e/ou blocos colocalizados. Por exemplo, em um exemplo, os cinco blocos espacialmente vizinhos podem ser os ilustrados na Figura 8A para uma ordem de codificação da esquerda para a direita, e os cinco blocos espacialmente vizinhos podem ser os ilustrados na Figura 8B para uma ordem de codificação da direita para a esquerda.

[0240] Por exemplo, para um primeiro bloco, em uma primeira imagem tendo uma ordem de codificação da esquerda para a direita, a unidade de compensação de movimento 316 pode construir uma primeira lista de preditores de vetor de movimento. Uma primeira entrada na primeira lista de preditores de vetor de movimento pode ser com base em informações de vetor de movimento de um bloco vizinho à esquerda do primeiro bloco (por exemplo, bloco 0 802 da Figura 8A). Para um segundo bloco, em uma segunda imagem tendo uma ordem de codificação da direita para a

esquerda, a unidade de compensação de movimento 316 pode construir uma segunda lista de preditores de vetor de movimento. Uma primeira entrada na segunda lista de preditores de vetor de movimento pode ser com base em informações de vetor de movimento de um bloco vizinho à direita do segundo bloco (por exemplo, bloco 0 818 da Figura 8B).

[0241] A unidade de compensação de movimento 316 pode verificar informações de vetor de movimento armazenadas na buffer de imagem decodificada 314 (como um exemplo) para uma pluralidade de, incluindo todos, os blocos espacialmente vizinhos em paralelo. Em outras palavras, se a unidade de compensação de movimento 316 verificar informações de vetor de movimento para um bloco espacialmente vizinho específico pode não ser dependente de informações de vetor de movimento de qualquer outro bloco.

[0242] Com base na verificação das informações de vetor de movimento dos blocos espacialmente vizinhos, a unidade de compensação de movimento 316 pode construir uma lista de preditores de vetor de movimento tendo informações de vetor de movimento em uma ordem com base na disponibilidade das informações de vetor de movimento. A ordem das informações de vetor de movimento na lista de preditores de vetor de movimento para uma ordem de codificação da esquerda para a direita pode ser: informações de vetor de movimento do bloco vizinho à esquerda (por exemplo, bloco 0 802 da Figura 8A), informações de vetor de movimento de um bloco vizinho acima (por exemplo, bloco 1 804 da Figura 8A), informações de

vetor de movimento de um bloco vizinho acima à direita (por exemplo, bloco 2 806 da Figura 8A), informações de vetor de movimento de um bloco vizinho abaixo à esquerda (por exemplo, bloco 3 808 da Figura 8A), e informações de vetor de movimento de um bloco vizinho acima à esquerda (por exemplo, bloco 4 810 da Figura 8A). A ordem das informações de vetor de movimento na lista de preditores de vetor de movimento para uma ordem de codificação da direita para a esquerda pode ser: informações de vetor de movimento do bloco vizinho à direita (por exemplo, bloco 0 818 da Figura 8B), informações de vetor de movimento de um bloco vizinho acima (por exemplo, bloco 1 820 da Figura 8B), informações de vetor de movimento de um bloco vizinho acima à esquerda (por exemplo, bloco 2 822 da Figura 8B), informações de vetor de movimento de um bloco vizinho abaixo à direita (por exemplo, bloco 3 824 da Figura 8B), e informações de vetor de movimento de um bloco vizinho acima à direita (por exemplo, bloco 4 826 da Figura 8B).

[0243] O ordenamento acima baseia-se na disponibilidade das informações de vetor de movimento. Se informações de vetor de movimento não estão disponíveis para um bloco particular, então, tais informações de vetor de movimento não são adicionadas à lista de preditores de vetor de movimento. Em alguns exemplos, as próximas informações de vetor de movimento disponíveis podem ser incluídas no espaço reservado para o bloco para o qual não houve informações de vetor de movimento. Por exemplo, informações de vetor de movimento para o bloco 1 804 da Figura 8A podem ser a segunda entrada na lista de

preditores de vetor de movimento, e informações de vetor de movimento para o bloco 2 806 da Figura 8A podem ser a terceira entrada na lista de preditores de vetor de movimento. Se informações de vetor de movimento para o bloco 1 804 da Figura 8A não estão disponíveis e informações de vetor de movimento para o bloco 2 806 da Figura 8A estão disponíveis, então, informações de vetor de movimento para o bloco 2 806 podem estar na segunda entrada da lista de preditores de vetor de movimento.

[0244] No exemplo acima, a unidade de compensação de movimento 316 determinou uma primeira lista de preditores de vetor de movimento e uma segunda lista de preditores de vetor de movimento com base em blocos espacialmente vizinhos. Em alguns exemplos, a unidade de compensação de movimento 316 pode incluir informações de vetor de movimento, armazenadas em buffer de imagem decodificada 314 (como um exemplo), de blocos colocalizados na primeira lista de preditores de vetor de movimento e na segunda lista de preditores de vetor de movimento.

[0245] Como um exemplo, a unidade de compensação de movimento 316 pode determinar se informações de vetor de movimento para um bloco colocalizado central localizado em uma imagem, que não a primeira imagem (por exemplo, bloco 5 812 da Figura 8A), estão disponíveis antes de determinar se informações de vetor de movimento em qualquer outro bloco colocalizado localizado na imagem, que não a primeira imagem, estão disponíveis, e com base nas informações de vetor de movimento para o bloco colocalizado central estando disponíveis, a unidade de compensação de movimento

316 pode adicionar as informações de vetor de movimento ao bloco colocalizado central à primeira lista de preditores de vetor de movimento.

[0246] Em alguns exemplos, a unidade de compensação de movimento 316 pode determinar que informações de vetor de movimento para um bloco colocalizado central localizado em uma imagem, que não a primeira imagem, não estão disponíveis. Em tais exemplos, após determinar que as informações de vetor de movimento para o bloco colocalizado central não estão disponíveis, a unidade de compensação de movimento 316 pode determinar se informações de vetor de movimento para um bloco colocalizado inferior localizado na imagem, que não a primeira imagem, estão disponíveis (por exemplo, bloco 6 814 da Figura 8A), e com base nas informações de vetor de movimento para o bloco colocalizado inferior estando disponíveis, a unidade de compensação de movimento 316 pode adicionar as informações de vetor de movimento ao bloco colocalizado inferior à primeira lista de preditores de vetor de movimento.

[0247] Em alguns exemplos, a unidade de compensação de movimento 316 pode determinar que informações de vetor de movimento para um bloco colocalizado central localizado em uma imagem, que não a primeira imagem, não estão disponíveis. Em alguns casos, após determinar que as informações de vetor de movimento para o bloco colocalizado central não estão disponíveis, a unidade de compensação de movimento 316 pode determinar que informações de vetor de movimento para um bloco

colocalizado inferior localizado na imagem, que não a primeira imagem, não estão disponíveis. Nesses casos, após determinar que as informações de vetor de movimento para o bloco colocalizado inferior não estão disponíveis, a unidade de compensação de movimento 316 pode determinar se informações de vetor de movimento para um bloco colocalizado direito localizado na imagem, que não a primeira imagem, estão disponíveis (por exemplo, bloco 7 816 da Figura 8A), e com base nas informações de vetor de movimento para o bloco colocalizado direito estando disponíveis, a unidade de compensação de movimento 316 pode adicionar as informações de vetor de movimento ao bloco colocalizado direito à primeira lista de preditores de vetor de movimento.

[0248] A unidade de compensação de movimento 316 pode realizar operações similares para adicionar informações de vetor de movimento para blocos colocalizados para construir a segunda lista de preditores de vetor de movimento. No entanto, para a segunda lista de preditores de vetor de movimento, a unidade de compensação de movimento 316 pode adicionar informações de vetor de movimento com base na disponibilidade do bloco colocalizado central 5 828, bloco colocalizado inferior 6 830, e bloco colocalizado esquerdo 7 832 da Figura 8B.

[0249] Na construção de listas de preditores de vetor de movimento, a unidade de compensação de movimento 316 pode construir as listas de preditores de vetor de movimento com base em tamanho do bloco, em alguns exemplos. Por exemplo, a unidade de compensação de movimento 316 pode

construir a primeira lista de preditores de vetor de movimento, tendo um primeiro tamanho máximo, com base no primeiro bloco tendo um tamanho maior do que um tamanho limite. A unidade de compensação de movimento 316 pode construir uma terceira lista de preditores de vetor de movimento, tendo um segundo tamanho máximo que é inferior ao primeiro tamanho máximo, para um terceiro bloco na primeira imagem com base no terceiro bloco tendo um tamanho inferior ao tamanho limite. Como um exemplo, o primeiro tamanho máximo é igual a seis, o segundo tamanho máximo é igual a quatro, e o tamanho limite é igual a $N \times 4$ ou $4 \times N$, em que N é menor do que ou igual a oito.

[0250] A unidade de compensação de movimento 316, com a unidade de reconstrução 310, pode ser configurada para decodificar o primeiro bloco e o segundo bloco. Por exemplo, a unidade de compensação de movimento 316 recebe informações indicativas de uma entrada na primeira lista de preditores de vetor de movimento e determinam um vetor de movimento para o primeiro bloco com base na entrada. A unidade de compensação de movimento 316 pode determinar um bloco de predição, armazenado em buffer de imagem decodificada 314, para o primeiro bloco com base no vetor de movimento. A unidade de reconstrução 310 pode ser configurada para reconstruir o primeiro bloco com base no bloco de predição e informações recebidas indicativas de dados residuais representando uma diferença entre o bloco de predição e o primeiro bloco. A unidade de compensação de movimento 316 e a unidade de reconstrução 310 podem ser configuradas para similarmente decodificar o segundo bloco.

[0251] Como outro exemplo, se os elementos de sintaxe de informações de predição indicam que o bloco atual é intrapredito, a unidade de intrapredição 318 pode gerar o bloco de predição de acordo com um modo de intrapredição indicado pelos elementos de sintaxe de informações de predição. Novamente, a unidade de intrapredição 318 pode de forma geral realizar o processo de intrapredição de maneira que é substancialmente similar ao descrito com relação À unidade de intrapredição 226 (Figura 2). A unidade de intrapredição 318 pode recuperar dados de amostras vizinhos ao bloco atual de DPB 314.

[0252] A unidade de reconstrução 310 pode reconstruir o bloco atual usando o bloco de predição e o bloco residual. Por exemplo, a unidade de reconstrução 310 pode adicionar amostras do bloco residual a amostras correspondentes do bloco de predição para reconstruir o bloco atual.

[0253] A unidade de filtro 312 pode realizar uma ou mais operações de filtro em blocos reconstruídos. Por exemplo, a unidade de filtro 312 pode realizar operações de desbloqueio para reduzir artefatos de bloqueio ao longo das bordas dos bloco reconstruídos. As operações da unidade de filtro 312 não são necessariamente realizadas em todos os exemplos.

[0254] O decodificador de vídeo 300 pode armazenar os blocos reconstruídos em DPB 314. Como discutido acima, DPB 314 pode prover informações de referência, tais como amostras de uma imagem atual para intrapredição e imagens previamente decodificadas para

subsequente compensação de movimento, à unidade de processamento de predição 304. Além disso, o decodificador de vídeo 300 pode gerar imagens decodificadas de DPB para subsequente apresentação em um dispositivo de exibição, tal como o dispositivo de exibição 118 da Figura 1.

[0255] A Figura 9 é um fluxograma ilustrando um método exemplificativo de codificar data. Para facilidade de descrição, o exemplo da Figura 9 é descrito com referência aos circuitos de processamento. Um exemplo dos circuitos de processamento é codificador de vídeo 200. Outro exemplo dos circuitos de processamento é decodificador de vídeo 300. Os circuitos de processamento podem ser circuitos de função fixa, circuitos programáveis, ou uma combinação de circuitos de função fixa e programáveis.

[0256] Em um ou mais exemplos, a memória pode ser configurada para armazenar informações de vetor de movimento, tais como informações de vetor de movimento de blocos espacialmente vizinhos e blocos colocalizados. Como um exemplo, a memória pode ser memória de dados de vídeo 230, buffer de imagem decodificada 218, ou memória 106. Como outro exemplo, a memória pode ser buffer de imagem decodificada 314 ou memória 120.

[0257] Para um primeiro bloco, em uma primeira imagem tendo uma ordem de codificação da esquerda para a direita, os circuitos de processamento podem ser configurados para construir uma primeira lista de preditores de vetor de movimento (400). Uma primeira entrada na primeira lista de preditores de vetor de

movimento pode ser com base em informações de vetor de movimento de um bloco vizinho à esquerda do primeiro bloco armazenado na memória. Um exemplo do bloco vizinho à esquerda é bloco 0 802 da Figura 8A, em que o bloco 0 802 da Figura 8A está localizado à esquerda do bloco atual 800 da Figura 8A e compartilha um limite esquerdo do bloco atual 800 da Figura 8A (por exemplo, o bloco vizinho à esquerda é um bloco à esquerda do primeiro bloco tendo um limite inferior que é o mesmo que um limite inferior do primeiro bloco). Em alguns exemplos, os circuitos de processamento podem ser configurados para construir a primeira lista de preditores de vetor de movimento tendo informações de vetor de movimento na seguinte ordem com base na disponibilidade de informações de vetor de movimento: informações de vetor de movimento do bloco vizinho à esquerda (por exemplo, bloco 0 802), informações de vetor de movimento de um bloco vizinho acima (por exemplo, bloco 1 804), informações de vetor de movimento de um bloco vizinho acima à direita (por exemplo, bloco 2 806), informações de vetor de movimento de um bloco vizinho abaixo à esquerda (por exemplo, bloco 3 808), e informações de vetor de movimento de um bloco vizinho acima à esquerda (por exemplo, bloco 4 810).

[0258] Em alguns exemplos, para construir a primeira lista de preditores de vetor de movimento, os circuitos de processamento podem ser configurados para determinar se informações de vetor de movimento para um bloco colocalizado central (por exemplo, bloco 812) localizado em uma imagem, que não a primeira imagem, estão

disponíveis antes de determinar se informações de vetor de movimento em qualquer outro bloco colocalizado localizado na imagem, que não a primeira imagem, estão disponíveis, e com base nas informações de vetor de movimento para o bloco colocalizado central estando disponíveis, adicionar as informações de vetor de movimento para o bloco colocalizado central à primeira lista de preditores de vetor de movimento. Em alguns exemplos, para construir a primeira lista de preditores de vetor de movimento, os circuitos de processamento são configurados para determinar que informações de vetor de movimento para um bloco colocalizado central localizado em uma imagem, que não a primeira imagem, não estão disponíveis, após determinar que as informações de vetor de movimento para o bloco colocalizado central não estão disponíveis, determinar se informações de vetor de movimento para um bloco colocalizado inferior (por exemplo, bloco 6 814 que é inferior e mais próximo à direita do bloco atual 800 da Figura 8A) localizado na imagem, que não a primeira imagem, estão disponíveis, e com base nas informações de vetor de movimento para o bloco colocalizado inferior estando disponíveis, adicionar as informações de vetor de movimento para o bloco colocalizado inferior à primeira lista de preditores de vetor de movimento. Em alguns exemplos, um limite direito do bloco colocalizado inferior (por exemplo, bloco 6 814) está no mesmo local na imagem, que não a primeira imagem, que um limite direito do primeiro bloco na primeira imagem.

[0259] Em alguns exemplos, para construir a

primeira lista de preditores de vetor de movimento, os circuitos de processamento são configurados para determinar que informações de vetor de movimento para um bloco colocalizado central localizado em uma imagem, que não a primeira imagem, não estão disponíveis, após determinar que as informações de vetor de movimento para o bloco colocalizado central não estão disponíveis, determinar que informações de vetor de movimento para um bloco colocalizado inferior localizado na imagem, que não a primeira imagem, não estão disponíveis, após determinar que as informações de vetor de movimento para o bloco colocalizado inferior não estão disponíveis, determinar se informações de vetor de movimento para um bloco colocalizado direito (por exemplo, bloco 7 816 que está à direita e mais próximo à parte inferior do bloco atual 800 da Figura 8A) localizado na imagem, que não a primeira imagem, estão disponíveis, e com base nas informações de vetor de movimento para o bloco colocalizado direito estando disponíveis, adicionar as informações de vetor de movimento para o bloco colocalizado direito à primeira lista de preditores de vetor de movimento. Em alguns exemplos, um limite inferior do bloco colocalizado direito (por exemplo, bloco 7 816) está no mesmo local na imagem, que não a primeira imagem, que um limite inferior do primeiro bloco na primeira imagem.

[0260] Os circuitos de processamento podem ser configurados para codificar o primeiro bloco na primeira imagem com base na primeira lista de preditores de vetor de movimento (402). Como um exemplo, para codificar o primeiro

bloco na primeira imagem com base na primeira lista de preditores de vetor de movimento, os circuitos de processamento podem ser configurados para decodificar o primeiro bloco na primeira imagem com base na primeira lista de preditores de vetor de movimento. Para decodificar o primeiro bloco, os circuitos de processamento podem ser configurados para receber informações indicativas de uma entrada na primeira lista de preditores de vetor de movimento, determinar um vetor de movimento para o primeiro bloco com base na entrada, determinar um bloco de predição para o primeiro bloco com base no vetor de movimento, e reconstruir o primeiro bloco com base no bloco de predição e informações recebidas indicativas de dados residuais representando uma diferença entre o bloco de predição e o primeiro bloco. Como outro exemplo, para codificar o primeiro bloco na primeira imagem com base na primeira lista de preditores de vetor de movimento, os circuitos de processamento podem ser configurados para codificar o primeiro bloco na primeira imagem com base na primeira lista de preditores de vetor de movimento. Para codificar o primeiro bloco, os circuitos de processamento podem ser configurados para determinar um bloco de predição para o primeiro bloco, determinar um vetor de movimento para o primeiro bloco que identifica o bloco de predição, determinar uma entrada na primeira lista de preditores de vetor de movimento com base no vetor de movimento, sinalizar informações indicativas da entrada, e sinalizar informações indicativas de dados residuais representando uma diferença entre o primeiro bloco e o bloco de predição.

[0261] Para um segundo bloco, em uma segunda imagem tendo uma ordem de codificação da direita para a esquerda, os circuitos de processamento podem ser configurados para construir uma segunda lista de preditores de vetor de movimento (404). Uma primeira entrada na segunda lista de preditores de vetor de movimento podem ser com base em informações de vetor de movimento de um bloco vizinho à direita do segundo bloco armazenado na memória. Um exemplo do bloco vizinho à direita é o bloco 0 818 da Figura 8B, em que o bloco 0 818 da Figura 8B está localizado à direita do bloco atual 800 da Figura 8B e compartilha um limite direito do bloco atual 800 da Figura 8B (por exemplo, o bloco vizinho à direita é um bloco à direita do segundo bloco tendo um limite inferior que é o mesmo que um limite inferior do segundo bloco). Em alguns exemplos, os circuitos de processamento podem ser configurados para construir a segunda lista de preditores de vetor de movimento tendo informações de vetor de movimento na seguinte ordem com base na disponibilidade de informações de vetor de movimento: informações de vetor de movimento do bloco vizinho à direita (por exemplo, bloco 0 818), informações de vetor de movimento de um bloco vizinho acima (por exemplo, bloco 1 820), informações de vetor de movimento de um bloco vizinho acima à esquerda (por exemplo, bloco 2 822), informações de vetor de movimento de um bloco vizinho abaixo à direita (por exemplo, bloco 3 824), e informações de vetor de movimento de um bloco vizinho acima à direita (por exemplo, bloco 4 826).

[0262] Em alguns exemplos, para construir a

segunda lista de preditores de vetor de movimento, os circuitos de processamento podem ser configurados para determinar se informações de vetor de movimento para um bloco colocalizado central (por exemplo, bloco 5 828) localizado em uma imagem, que não a segunda imagem, estão disponíveis antes de determinar se informações de vetor de movimento em qualquer outro bloco colocalizado localizado na imagem, que não a segunda imagem, estão disponíveis, e com base nas informações de vetor de movimento para o bloco colocalizado central estando disponíveis, adicionar as informações de vetor de movimento para o bloco colocalizado central à segunda lista de preditores de vetor de movimento. Em alguns exemplos, para construir a segunda lista de preditores de vetor de movimento, os circuitos de processamento são configurados para determinar que informações de vetor de movimento para um bloco colocalizado central localizado em uma imagem, que não a segunda imagem, não estão disponíveis, após determinar que as informações de vetor de movimento para o bloco colocalizado central não estão disponíveis, determinar se informações de vetor de movimento para um bloco colocalizado inferior (por exemplo, bloco 6 830 que é inferior e mais próximo à esquerda do bloco atual 800 da Figura 8B) localizado na imagem, que não a segunda imagem, estão disponíveis, e com base nas informações de vetor de movimento para o bloco colocalizado inferior estando disponíveis, adicionar as informações de vetor de movimento para o bloco colocalizado inferior à segunda lista de preditores de vetor de movimento. Em alguns exemplos, um

limite esquerdo do bloco colocalizado inferior (por exemplo, bloco 6 830) está no mesmo local na imagem, que não a segunda imagem, que um limite esquerdo do segundo bloco na segunda imagem.

[0263] Em alguns exemplos, para construir a segunda lista de preditores de vetor de movimento, os circuitos de processamento são configurados para determinar que informações de vetor de movimento para um bloco colocalizado central localizado em uma imagem, que não a segunda imagem, não estão disponíveis, após determinar que as informações de vetor de movimento para o bloco colocalizado central não estão disponíveis, determinar que informações de vetor de movimento para um bloco colocalizado inferior localizado na imagem, que não a segunda imagem, não estão disponíveis, após determinar que as informações de vetor de movimento para o bloco colocalizado inferior não estão disponíveis, determinar se informações de vetor de movimento para um bloco colocalizado esquerdo (por exemplo, bloco 7 832 que está à esquerda e mais próximo à parte inferior do bloco atual 800 da Figura 8B) localizado na imagem, que não a segunda imagem, estão disponíveis, e com base nas informações de vetor de movimento para o bloco colocalizado esquerdo estando disponíveis, adicionar as informações de vetor de movimento para o bloco colocalizado esquerdo à segunda lista de preditores de vetor de movimento. Em alguns exemplos, um limite inferior do bloco colocalizado esquerdo (por exemplo, bloco 7 832) está no mesmo local na imagem, que não a segunda imagem, que um limite inferior do segundo

bloco na segunda imagem.

[0264] Os circuitos de processamento podem ser configurados para codificar o segundo bloco na segunda imagem com base na segunda lista de preditores de vetor de movimento (406). Como um exemplo, para codificar o segundo bloco na segunda imagem com base na segunda lista de preditores de vetor de movimento, os circuitos de processamento podem ser configurados para decodificar o segundo bloco na segunda imagem com base na segunda lista de preditores de vetor de movimento. Para decodificar o segundo bloco, os circuitos de processamento podem ser configurados para receber informações indicativas de uma entrada na segunda lista de preditores de vetor de movimento, determinar um vetor de movimento para o segundo bloco com base na entrada, determinar um bloco de predição para o segundo bloco com base no vetor de movimento, e reconstruir o segundo bloco com base no bloco de predição e informações recebidas indicativas de dados residuais representando uma diferença entre o bloco de predição e o segundo bloco. Como outro exemplo, para codificar o segundo bloco na segunda imagem com base na segunda lista de preditores de vetor de movimento, os circuitos de processamento podem ser configurados para codificar o segundo bloco na segunda imagem com base na segunda lista de preditores de vetor de movimento. Para codificar o segundo bloco, os circuitos de processamento podem ser configurados para determinar um bloco de predição para o segundo bloco, determinar um vetor de movimento para o segundo bloco que identifica o bloco de predição,

determinar uma entrada na segunda lista de preditores de vetor de movimento com base no vetor de movimento, sinalizar informações indicativas da entrada, e sinalizar informações indicativas de dados residuais representando uma diferença entre o segundo bloco e o bloco de predição.

[0265] A seguir estão algumas técnicas exemplificativas que podem ser realizadas em conjunto e/ou em combinação com qualquer uma das técnicas descritas nesta divulgação. Os exemplos a seguir são meramente alguns exemplos e não devem ser considerados limitantes ou necessários em todos os exemplos.

[0266] Exemplo 1. Um método de codificar dados de vídeo, o método compreendendo construir uma lista de preditores de vetor de movimento compreendendo cinco candidatos espaciais e até três candidatos temporais colocalizados e codificar um bloco de dados de vídeo usando a lista de preditores de vetor de movimento.

[0267] Exemplo 2. O método do exemplo 1, em que construir a lista de preditores de vetor de movimento compreende verificar a disponibilidade de candidatos espaciais antes de verificar a disponibilidade de candidatos temporais colocalizados.

[0268] Exemplo 3. O método do exemplo 1, em que os cinco candidatos espaciais e os três candidatos temporais colocalizados estão localizados em um padrão mostrado na Figura 8A.

[0269] Exemplo 4. O método do exemplo 1, em que os cinco candidatos espaciais e os três candidatos temporais colocalizados estão localizados em um padrão

mostrado na Figura 8B.

[0270] Exemplo 5. O método do exemplo 1, em que os cinco candidatos espaciais e os três candidatos temporais colocalizados estão localizados em um padrão mostrado na Figura 8C.

[0271] Exemplo 6. O método do exemplo 1, compreendendo ainda determinar os cinco candidatos espaciais e os três candidatos temporais colocalizados com base em um ou mais de um modo de codificação para o bloco, um tipo de particionamento do bloco, ou uma direção de codificação do bloco.

[0272] Exemplo 7. O método do exemplo 1, em que construir a lista de preditores de vetor de movimento compreende verificar a disponibilidade de candidatos espaciais em paralelo com verificação da disponibilidade de candidatos temporais colocalizados.

[0273] Exemplo 8. O método do exemplo 7, compreendendo ainda determinar a validade de uma posição respectiva dos candidatos espaciais e dos candidatos temporais colocalizados e adicionar informações de movimento à lista de preditores de vetor de movimento de candidatos que são considerados válidos.

[0274] Exemplo 9. O método do exemplo 1, em que construir a lista de preditores de vetor de movimento compreende determinar um tamanho efetivo máximo da lista de preditores de vetor de movimento com base em um tamanho do bloco de dados de vídeo e construir a lista de preditores de vetor de movimento com base no tamanho efetivo máximo.

[0275] Exemplo 10. O método de qualquer

combinação das técnicas dos exemplos 1 a 9.

[0276] Exemplo 11. O método de qualquer um dos exemplos 1 a 10, em que codificar compreende decodificar.

[0277] Exemplo 12. O método de qualquer um dos exemplos 1 a 10, em que codificar (*coding*) compreende codificação (*encoding*).

[0278] Exemplo 13. Um dispositivo para codificar dados de vídeo, o dispositivo compreendendo um ou mais meios para executar o método de qualquer um dos exemplos 1 a 10.

[0279] Exemplo 14. O dispositivo do exemplo 13, em que os um ou mais meios compreendem um ou mais processadores implementados em circuitos.

[0280] Exemplo 15. O dispositivo de qualquer um dos exemplos 13 e 14, compreendendo ainda uma memória para armazenar os dados de vídeo.

[0281] Exemplo 16. O dispositivo de qualquer combinação dos exemplos 13 a 15, compreendendo ainda um visor configurado para exibir dados de vídeo decodificados.

[0282] Exemplo 17. O dispositivo de qualquer combinação dos exemplos 13 a 16, em que o dispositivo compreende um ou mais de uma câmera, um computador, um dispositivo móvel, um dispositivo receptor de radiodifusão ou um conversor (*set-top box*).

[0283] Exemplo 18. O dispositivo de qualquer combinação dos exemplos 13 a 17, em que o dispositivo compreende um decodificador de vídeo.

[0284] Exemplo 19. O dispositivo de qualquer combinação dos exemplos 13 a 18, em que o dispositivo

compreende um codificador de vídeo.

[0285] Exemplo 20. Um meio de armazenamento legível por computador tendo nele armazenadas instruções que, quando executadas, fazem com que um ou mais processadores executem o método de qualquer um dos exemplos 1 a 10.

[0286] Exemplo 21. Qualquer combinação das técnicas descritas nesta divulgação.

[0287] Deve-se reconhecer que, dependendo do exemplo, certos atos ou eventos de qualquer uma das técnicas verificadas aqui podem ser realizados em uma sequência diferente, podem ser adicionados, mesclados ou totalmente excluídos (por exemplo, nem todos os atos ou eventos descritos são necessários para a prática das técnicas). Além disso, em certos exemplos, os atos ou eventos podem ser realizados concomitantemente, por exemplo, através de processamento multiencadeado, processamento de interrupção, ou múltiplos processadores, em vez de sequencialmente.

[0288] Em um ou mais exemplos, as funções descritas podem ser implementadas em hardware, software, firmware ou qualquer combinação destes. Se implementadas em software, as funções podem ser armazenadas em ou transmitidas como uma ou mais instruções ou código em um meio legível por computador e executadas por uma unidade de processamento baseada em hardware. Meios legíveis por computador podem incluir meios de armazenamento legíveis por computador, que correspondem a um meio tangível, tal como meios de armazenamento de dados, ou meios 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. Dessa forma, os meios legíveis por computador geralmente podem corresponder a (1) meios de armazenamento legíveis por computador tangíveis que são não transitórios ou (2) um meio de comunicação, tal como um sinal ou onda portadora. Os meios de armazenamento de dados podem ser quaisquer meios disponíveis que podem ser acessados por um ou mais computadores ou um ou mais processadores para recuperar instruções, código e/ou estruturas de dados para implementação das técnicas descritas nesta divulgação. Um produto de programa de computador pode incluir um meio legível por computador.

[0289] A título de exemplo, e não de 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 magnético, memória flash, ou qualquer outro meio que pode ser usado para armazenar código do programa desejado na forma de instruções ou estruturas de dados e que pode ser acessado por um computador. Além disso, qualquer conexão é apropriadamente chamada de meio legível por computador. Por exemplo, se as instruções são transmitidas de um site, servidor ou outra fonte remota usando um cabo coaxial, cabo de fibra óptica, par trançado, linha de assinante digital (DSL) ou tecnologias sem fio, tais como infravermelho, 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-se entender, no entanto, que os meios de armazenamento legíveis por computador e os meios de armazenamento de dados não incluem conexões, ondas portadoras, sinais ou outros meios transitórios, mas são direcionados a meios de armazenamento tangíveis não transitórios. Disco (disk) e disco (disc), conforme usado neste documento, incluem disco compacto (CD), disco laser, disco óptico, disco versátil digital (DVD), disco flexível e disco Blu-ray, em que os discos (disks) geralmente reproduzem dados magneticamente, enquanto os discos (discs) reproduzem dados opticamente com lasers. As combinações dos itens acima também devem ser incluídas no escopo de meios legíveis por computador.

[0290] 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 (ASICs), matrizes de portas programáveis em campo (FPGAs), ou outros circuitos de lógica discretos ou integrados equivalentes. Por conseguinte, o termo "processador", conforme usado neste documento, pode se referir a qualquer estrutura acima ou qualquer outra estrutura adequada para a implementação das técnicas descritas neste documento. Além disso, em alguns aspectos, a funcionalidade descrita neste documento pode ser fornecida em módulos de hardware e/ou software dedicados configurados para codificar e decodificar ou incorporados em um codec combinado. Além disso, as técnicas

podem ser totalmente implementadas em um ou mais circuitos ou elementos lógicos.

[0291] As técnicas desta divulgação podem ser implementadas em uma ampla variedade de dispositivos ou aparelhos, incluindo um telefone sem fio, um circuito integrado (IC) ou um conjunto de ICs (por exemplo, um *chipset*). Vários componentes, módulos ou unidades são descritos nesta divulgação para enfatizar aspectos funcionais de dispositivos configurados para executar as técnicas divulgadas, mas não necessariamente requerem a realização por diferentes unidades de hardware. Em vez disso, conforme mencionado acima, várias unidades podem ser combinadas em uma unidade de hardware de codec ou fornecida por uma coleção de unidades de hardware interoperativas, incluindo um ou mais conhecida como acima, em conjunto com software e/ou firmware adequados.

[0292] Vários exemplos foram descritos. Estes e outros exemplos estão dentro do escopo das reivindicações a seguir.

REIVINDICAÇÕES

1. Método de codificar dados de vídeo, o método compreendendo:

para um primeiro bloco, em uma primeira imagem tendo uma ordem de codificação da esquerda para a direita, construir uma primeira lista de preditores de vetor de movimento, em que uma primeira entrada na primeira lista de preditores de vetor de movimento baseia-se em informações de vetor de movimento de um bloco vizinho à esquerda do primeiro bloco;

codificar o primeiro bloco na primeira imagem com base na primeira lista de preditores de vetor de movimento;

para um segundo bloco, em uma segunda imagem tendo uma ordem de codificação da direita para a esquerda, construir uma segunda lista de preditores de vetor de movimento, em que uma primeira entrada na segunda lista de preditores de vetor de movimento baseia-se em informações de vetor de movimento de um bloco vizinho à direita do segundo bloco; e

codificar o segundo bloco na segunda imagem com base na segunda lista de preditores de vetor de movimento.

2. Método, de acordo com a reivindicação 1, em que construir a primeira lista de preditores de vetor de movimento compreende construir a primeira lista de preditores de vetor de movimento tendo informações de vetor de movimento na seguinte ordem com base na disponibilidade de informações de vetor de movimento:

informações de vetor de movimento do bloco vizinho à esquerda;

informações de vetor de movimento de um bloco

vizinho acima;

informações de vetor de movimento de um bloco vizinho acima à direita;

informações de vetor de movimento de um bloco vizinho abaixo à esquerda; e

informações de vetor de movimento de um bloco vizinho acima à esquerda.

3. Método, de acordo com a reivindicação 1, em que construir a segunda lista de preditores de vetor de movimento compreende construir a segunda lista de preditores de vetor de movimento tendo informações de vetor de movimento na seguinte ordem com base na disponibilidade de informações de vetor de movimento:

informações de vetor de movimento do bloco vizinho à direita;

informações de vetor de movimento de um bloco vizinho acima;

informações de vetor de movimento de um bloco vizinho acima à esquerda;

informações de vetor de movimento de um bloco vizinho abaixo à direita; e

informações de vetor de movimento de um bloco vizinho acima à direita.

4. Método, de acordo com a reivindicação 1, em que o bloco vizinho à esquerda do primeiro bloco compreende um bloco à esquerda do primeiro bloco tendo um limite inferior que é o mesmo que um limite inferior do primeiro bloco, e em que o bloco vizinho à direita compreende um bloco à direita do segundo bloco tendo um limite inferior que é o mesmo que um limite inferior do segundo bloco.

5. Método, de acordo com a reivindicação 1, em que construir a primeira lista de preditores de vetor de movimento compreende verificar informações de vetor de movimento para uma pluralidade de blocos espacialmente vizinhos usados para construir a primeira lista de preditores de vetor de movimento em paralelo.

6. Método, de acordo com a reivindicação 1, em que construir a primeira lista de preditores de vetor de movimento compreende:

determinar se informações de vetor de movimento para um bloco colocalizado central localizado em uma imagem, que não a primeira imagem, estão disponíveis antes de determinar se informações de vetor de movimento em qualquer outro bloco colocalizado localizado na imagem, que não a primeira imagem, estão disponíveis; e

com base nas informações de vetor de movimento para o bloco colocalizado central estando disponíveis, adicionar as informações de vetor de movimento para o bloco colocalizado central à primeira lista de preditores de vetor de movimento.

7. Método, de acordo com a reivindicação 1, em que construir a primeira lista de preditores de vetor de movimento compreende:

determinar que informações de vetor de movimento para um bloco colocalizado central localizado em uma imagem, que não a primeira imagem, não estão disponíveis;

após determinar que as informações de vetor de movimento para o bloco colocalizado central não estão disponíveis, determinar se informações de vetor de movimento para um bloco colocalizado inferior localizado na

imagem, que não a primeira imagem, estão disponíveis; e

com base nas informações de vetor de movimento para o bloco colocalizado inferior estando disponíveis, adicionar as informações de vetor de movimento para o bloco colocalizado inferior à primeira lista de preditores de vetor de movimento.

8. Método, de acordo com a reivindicação 7, em que um limite direito do bloco colocalizado inferior está no mesmo local na imagem, que não a primeira imagem, que um limite direito do primeiro bloco na primeira imagem.

9. Método, de acordo com a reivindicação 1, em que construir a segunda lista de preditores de vetor de movimento compreende:

determinar que informações de vetor de movimento para um bloco colocalizado central localizado em uma imagem, que não a segunda imagem, não estão disponíveis;

após determinar que as informações de vetor de movimento para o bloco colocalizado central não estão disponíveis, determinar se informações de vetor de movimento para um bloco colocalizado inferior localizado na imagem, que não a segunda imagem, estão disponíveis; e

com base nas informações de vetor de movimento para o bloco colocalizado inferior estando disponíveis, adicionar as informações de vetor de movimento para o bloco colocalizado inferior à segunda lista de preditores de vetor de movimento.

10. Método, de acordo com a reivindicação 9, em que um limite esquerdo do bloco colocalizado inferior está no mesmo local na imagem, que não a segunda imagem, que um limite esquerdo do segundo bloco na segunda imagem.

11. Método, de acordo com a reivindicação 1, em que construir a primeira lista de preditores de vetor de movimento compreende:

determinar que informações de vetor de movimento para um bloco colocalizado central localizado em uma imagem, que não a primeira imagem, não estão disponíveis;

após determinar que as informações de vetor de movimento para o bloco colocalizado central não estão disponíveis, determinar que informações de vetor de movimento para um bloco colocalizado inferior localizado na imagem, que não a primeira imagem, não estão disponíveis;

após determinar que as informações de vetor de movimento para o bloco colocalizado inferior não estão disponíveis, determinar se informações de vetor de movimento para um bloco colocalizado direito localizado na imagem, que não a primeira imagem, estão disponíveis; e

com base nas informações de vetor de movimento para o bloco colocalizado direito estando disponíveis, adicionar as informações de vetor de movimento para o bloco colocalizado direito à primeira lista de preditores de vetor de movimento.

12. Método, de acordo com a reivindicação 11, em que um limite inferior do bloco colocalizado direito está no mesmo local na imagem, que não a primeira imagem, que um limite inferior do primeiro bloco na primeira imagem.

13. Método, de acordo com a reivindicação 1, em que construir a segunda lista de preditores de vetor de movimento compreende:

determinar que informações de vetor de movimento para um bloco colocalizado central localizado em uma

imagem, que não a segunda imagem, não estão disponíveis;

após determinar que as informações de vetor de movimento para o bloco colocalizado central não estão disponíveis, determinar que informações de vetor de movimento para um bloco colocalizado inferior localizado na imagem, que não a segunda imagem, não estão disponíveis;

após determinar que as informações de vetor de movimento para o bloco colocalizado inferior não estão disponíveis, determinar se informações de vetor de movimento para um bloco colocalizado esquerdo localizado na imagem, que não a segunda imagem, estão disponíveis; e

com base nas informações de vetor de movimento para o bloco colocalizado esquerdo estando disponíveis, adicionar as informações de vetor de movimento para o bloco colocalizado esquerdo à segunda lista de preditores de vetor de movimento.

14. Método, de acordo com a reivindicação 13, em que um limite inferior do bloco colocalizado esquerdo está no mesmo local na imagem, que não a segunda imagem, que um limite inferior do segundo bloco na segunda imagem.

15. Método, de acordo com a reivindicação 1, em que construir a primeira lista de preditores de vetor de movimento compreende construir a primeira lista de preditores de vetor de movimento, tendo um primeiro tamanho máximo, com base no primeiro bloco tendo um tamanho maior do que um tamanho limite, o método compreendendo ainda:

construir uma terceira lista de preditores de vetor de movimento, tendo um segundo tamanho máximo que é inferior ao primeiro tamanho máximo, para um terceiro bloco na primeira imagem com base no terceiro bloco tendo um

tamanho inferior ao tamanho limite.

16. Método, de acordo com a reivindicação 15, em que o primeiro tamanho máximo é igual a seis, em que o segundo tamanho máximo é igual a quatro, e em que o tamanho limite é igual a $N \times 4$ ou $4 \times N$, em que N é menor do que ou igual a oito.

17. Método, de acordo com a reivindicação 1, em que codificar o primeiro bloco na primeira imagem com base na primeira lista de preditores de vetor de movimento compreende decodificar o primeiro bloco na primeira imagem com base na primeira lista de preditores de vetor de movimento, em que decodificar o primeiro bloco compreende:

receber informações indicativas de uma entrada na primeira lista de preditores de vetor de movimento;

determinar um vetor de movimento para o primeiro bloco com base na entrada;

determinar um bloco de predição para o primeiro bloco com base no vetor de movimento; e

reconstruir o primeiro bloco com base no bloco de predição e informações recebidas indicativas de dados residuais representando uma diferença entre o bloco de predição e o primeiro bloco.

18. Método, de acordo com a reivindicação 1, em que codificar o primeiro bloco na primeira imagem com base na primeira lista de preditores de vetor de movimento compreende codificar o primeiro bloco na primeira imagem com base na primeira lista de preditores de vetor de movimento, em que codificar o primeiro bloco compreende:

determinar um bloco de predição para o primeiro

bloco;

determinar um vetor de movimento para o primeiro bloco que identifica o bloco de predição;

determinar uma entrada na primeira lista de preditores de vetor de movimento com base no vetor de movimento;

sinalizar informações indicativas da entrada; e

sinalizar informações indicativas de dados residuais representando uma diferença entre o primeiro bloco e o bloco de predição.

19. Dispositivo para codificar dados de vídeo, o dispositivo compreendendo:

memória configurada para armazenar informações de vetor de movimento; e

circuitos de processamento acoplados à memória, em que os circuitos de processamento são configurados para:

para um primeiro bloco, em uma primeira imagem tendo uma ordem de codificação da esquerda para a direita, construir uma primeira lista de preditores de vetor de movimento, em que uma primeira entrada na primeira lista de preditores de vetor de movimento baseia-se em informações de vetor de movimento de um bloco vizinho à esquerda do primeiro bloco armazenado na memória;

codificar o primeiro bloco na primeira imagem com base na primeira lista de preditores de vetor de movimento;

para um segundo bloco, em uma segunda imagem tendo uma ordem de codificação da direita para a esquerda, construir uma segunda lista de preditores de vetor de movimento, em que uma primeira entrada na segunda lista de preditores de vetor de movimento baseia-se em informações

de vetor de movimento de um bloco vizinho à direita do segundo bloco armazenado na memória; e

codificar o segundo bloco na segunda imagem com base na segunda lista de preditores de vetor de movimento.

20. Dispositivo, de acordo com a reivindicação 19, em que, para construir a primeira lista de preditores de vetor de movimento, os circuitos de processamento são configurados para construir a primeira lista de preditores de vetor de movimento tendo informações de vetor de movimento na seguinte ordem com base na disponibilidade de informações de vetor de movimento:

informações de vetor de movimento do bloco vizinho à esquerda;

informações de vetor de movimento de um bloco vizinho acima;

informações de vetor de movimento de um bloco vizinho acima à direita;

informações de vetor de movimento de um bloco vizinho abaixo à esquerda; e

informações de vetor de movimento de um bloco vizinho acima à esquerda.

21. Dispositivo, de acordo com a reivindicação 19, em que, para construir a segunda lista de preditores de vetor de movimento, os circuitos de processamento são configurados para construir a segunda lista de preditores de vetor de movimento tendo informações de vetor de movimento na seguinte ordem com base na disponibilidade de informações de vetor de movimento:

informações de vetor de movimento do bloco vizinho à direita;

informações de vetor de movimento de um bloco vizinho acima;

informações de vetor de movimento de um bloco vizinho acima à esquerda;

informações de vetor de movimento de um bloco vizinho abaixo à direita; e

informações de vetor de movimento de um bloco vizinho acima à direita.

22. Dispositivo, de acordo com a reivindicação 19, em que, o bloco vizinho à esquerda do primeiro bloco compreende um bloco à esquerda do primeiro bloco tendo um limite inferior que é o mesmo que um limite inferior do primeiro bloco, e em que o bloco vizinho à direita compreende um bloco à direita do segundo bloco tendo um limite inferior que é o mesmo que um limite inferior do segundo bloco.

23. Dispositivo, de acordo com a reivindicação 19, em que, para construir a primeira lista de preditores de vetor de movimento, os circuitos de processamento são configurados para verificar informações de vetor de movimento para uma pluralidade de blocos espacialmente vizinhos usados para construir a primeira lista de preditores de vetor de movimento em paralelo.

24. Dispositivo, de acordo com a reivindicação 19, em que, para construir a primeira lista de preditores de vetor de movimento, os circuitos de processamento são configurados para:

determinar se informações de vetor de movimento para um bloco colocalizado central localizado em uma imagem, que não a primeira imagem, estão disponíveis antes

de determinar se informações de vetor de movimento em qualquer outro bloco colocalizado localizado na imagem, que não a primeira imagem, estão disponíveis; e

com base nas informações de vetor de movimento para o bloco colocalizado central estando disponíveis, adicionar as informações de vetor de movimento para o bloco colocalizado central à primeira lista de preditores de vetor de movimento.

25. Dispositivo, de acordo com a reivindicação 19, em que, para construir a primeira lista de preditores de vetor de movimento, os circuitos de processamento são configurados para:

determinar que informações de vetor de movimento para um bloco colocalizado central localizado em uma imagem, que não a primeira imagem, não estão disponíveis;

após determinar que as informações de vetor de movimento para o bloco colocalizado central não estão disponíveis, determinar se informações de vetor de movimento para um bloco colocalizado inferior localizado na imagem, que não a primeira imagem, estão disponíveis; e

com base nas informações de vetor de movimento para o bloco colocalizado inferior estando disponíveis, adicionar as informações de vetor de movimento para o bloco colocalizado inferior à primeira lista de preditores de vetor de movimento,

em que um limite direito do bloco colocalizado inferior está no mesmo local na imagem, que não a primeira imagem, que um limite direito do primeiro bloco na primeira imagem.

26. Dispositivo, de acordo com a reivindicação

19, em que, para construir a segunda lista de preditores de vetor de movimento, os circuitos de processamento são configurados para:

determinar que informações de vetor de movimento para um bloco colocalizado central localizado em uma imagem, que não a segunda imagem, não estão disponíveis;

após determinar que as informações de vetor de movimento para o bloco colocalizado central não estão disponíveis, determinar que informações de vetor de movimento para um bloco colocalizado inferior localizado na imagem, que não a segunda imagem, não estão disponíveis;

após determinar que as informações de vetor de movimento para o bloco colocalizado inferior não estão disponíveis, determinar se informações de vetor de movimento para um bloco colocalizado esquerdo localizado na imagem, que não a segunda imagem, estão disponíveis; e

com base nas informações de vetor de movimento para o bloco colocalizado esquerdo estando disponíveis, adicionar as informações de vetor de movimento para o bloco colocalizado esquerdo à segunda lista de preditores de vetor de movimento,

em que um limite inferior do bloco colocalizado esquerdo está no mesmo local na imagem, que não a segunda imagem, que um limite inferior do segundo bloco na segunda imagem.

27. Dispositivo, de acordo com a reivindicação 19, em que, para construir a primeira lista de preditores de vetor de movimento, os circuitos de processamento são configurados para construir a primeira lista de preditores de vetor de movimento, tendo um primeiro tamanho máximo,

com base no primeiro bloco tendo um tamanho maior do que um tamanho limite, e em que os circuitos de processamento são configurados para:

construir uma terceira lista de preditores de vetor de movimento, tendo um segundo tamanho máximo que é inferior ao primeiro tamanho máximo, para um terceiro bloco na primeira imagem com base no terceiro bloco tendo um tamanho inferior ao tamanho limite.

28. Dispositivo, de acordo com a reivindicação 27, em que o primeiro tamanho máximo é igual a seis, em que o segundo tamanho máximo é igual a quatro, e em que o tamanho limite é igual a $N \times 4$ ou $4 \times N$, em que N é menor do que ou igual a oito.

29. Dispositivo, de acordo com a reivindicação 19, em que o dispositivo compreende um dispositivo de comunicação sem fio.

30. Meio de armazenamento legível por computador nele armazenando instruções que, quando executadas, levam um ou mais processadores a:

para um primeiro bloco, em uma primeira imagem tendo uma ordem de codificação da esquerda para a direita, construir uma primeira lista de preditores de vetor de movimento, em que uma primeira entrada na primeira lista de preditores de vetor de movimento baseia-se em informações de vetor de movimento de um bloco vizinho à esquerda do primeiro bloco;

codificar o primeiro bloco na primeira imagem com base na primeira lista de preditores de vetor de movimento;

para um segundo bloco, em uma segunda imagem tendo uma ordem de codificação da direita para a esquerda,

construir uma segunda lista de preditores de vetor de movimento, em que uma primeira entrada na segunda lista de preditores de vetor de movimento baseia-se em informações de vetor de movimento de um bloco vizinho à direita do segundo bloco; e

codificar o segundo bloco na segunda imagem com base na segunda lista de preditores de vetor de movimento.

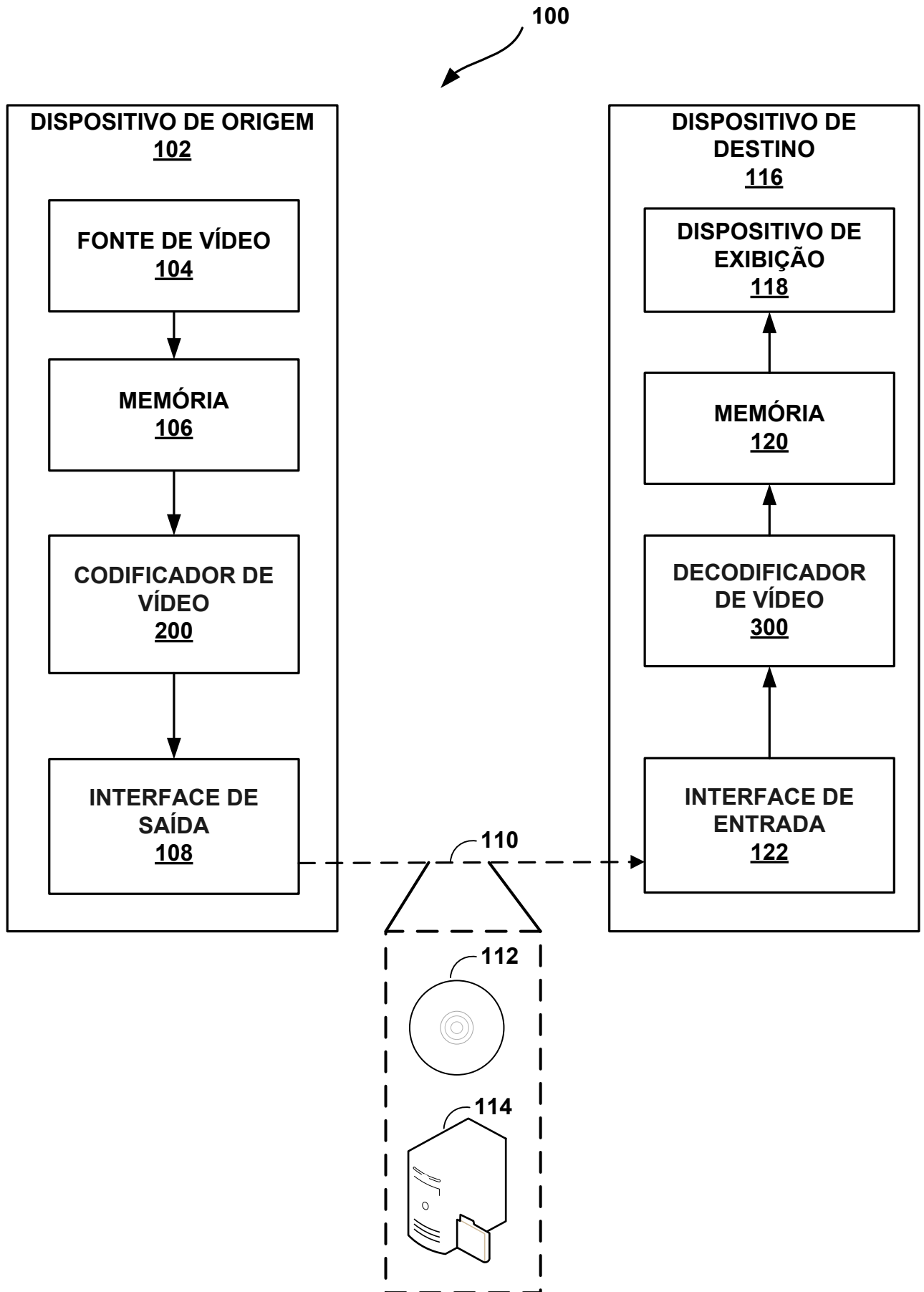


FIG. 1

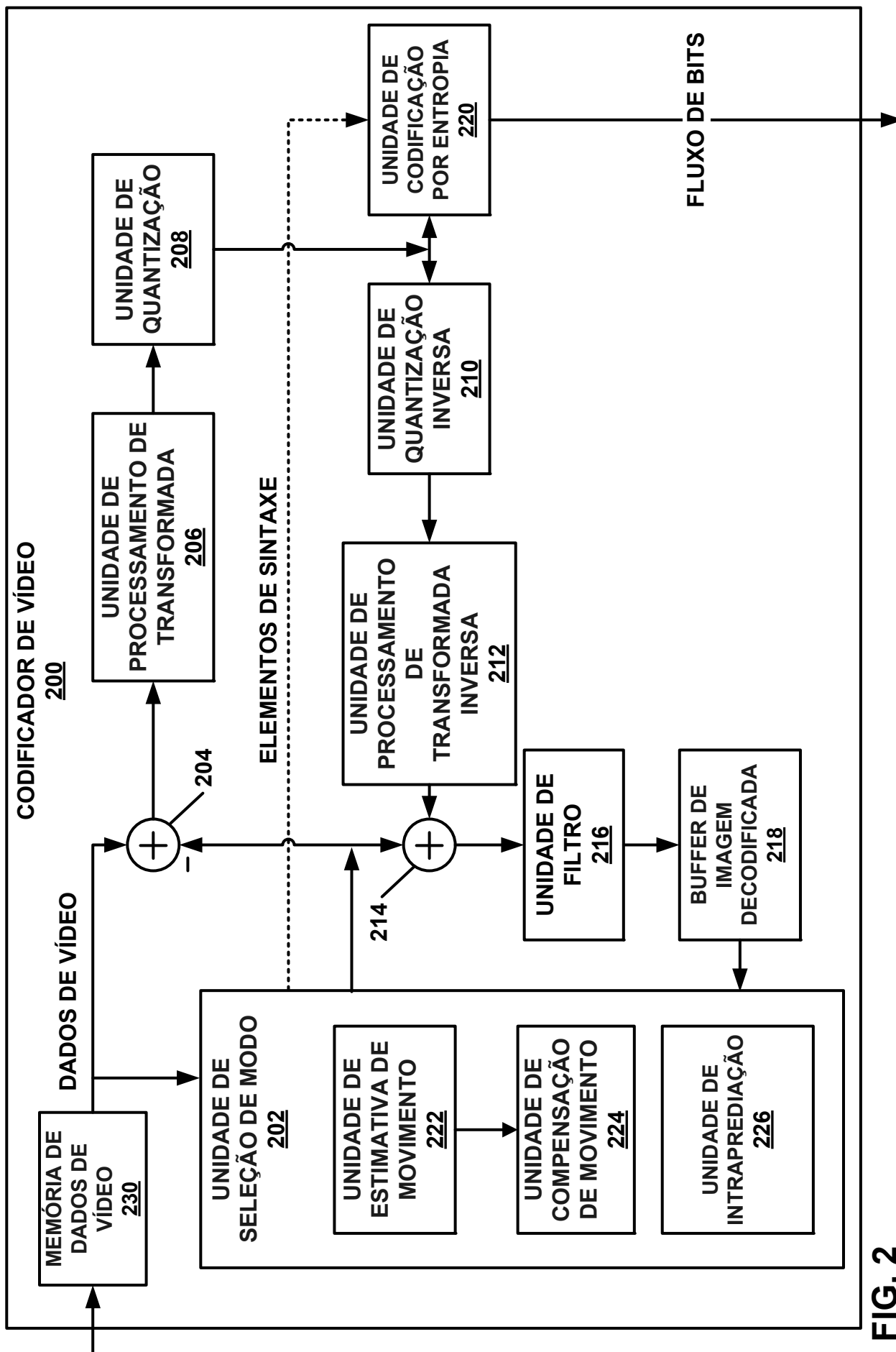


FIG. 2

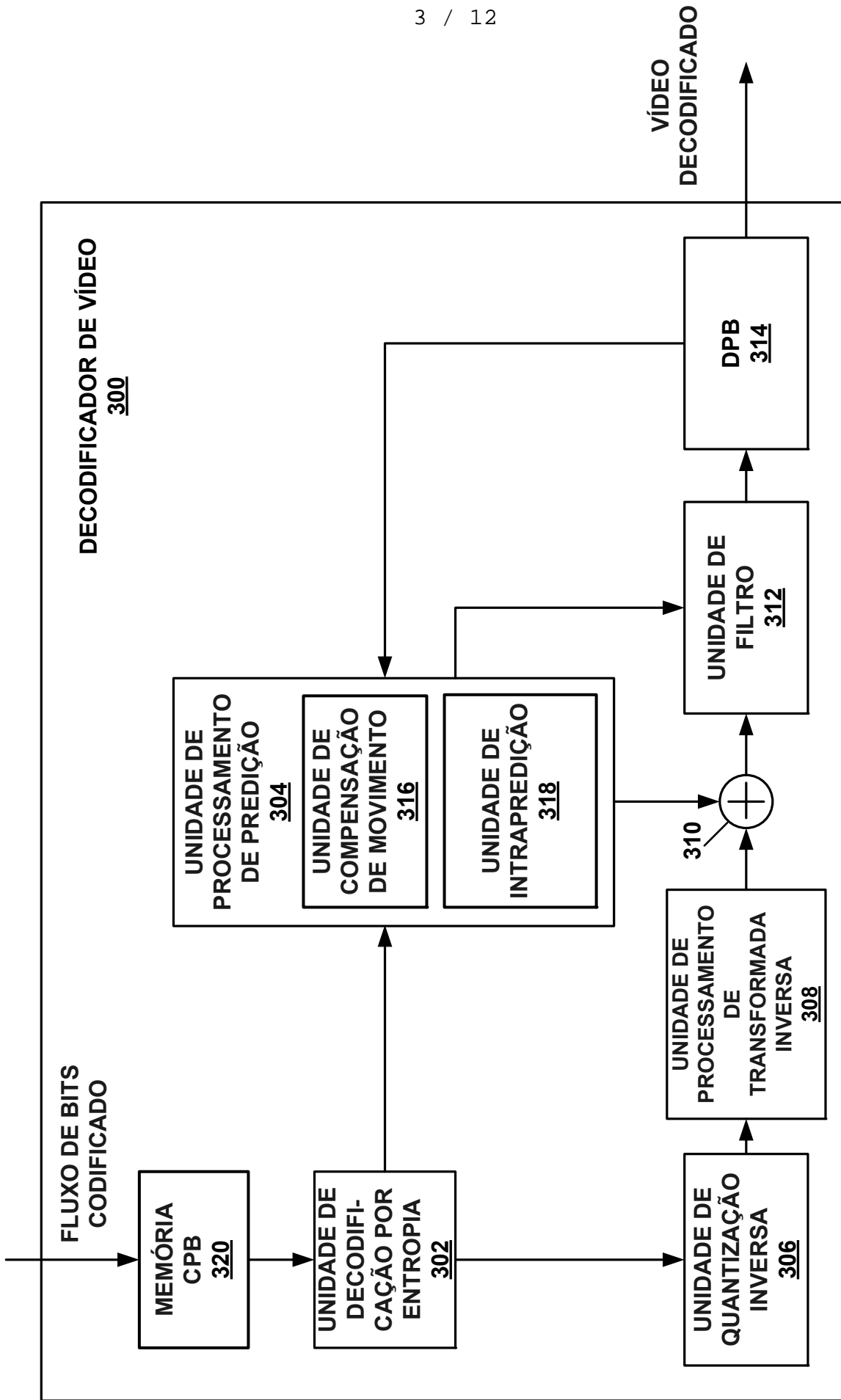


FIG. 3

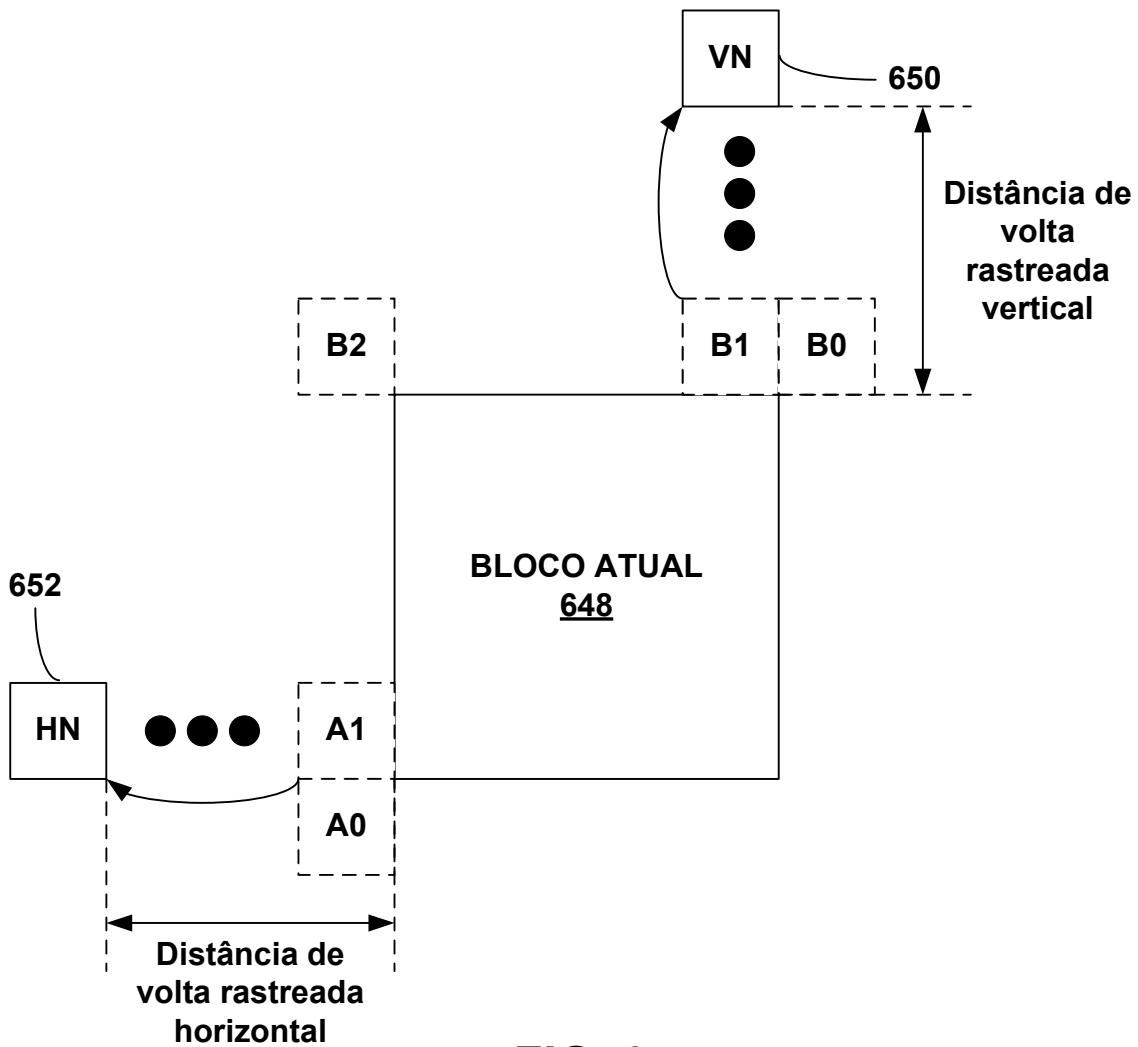


FIG. 6

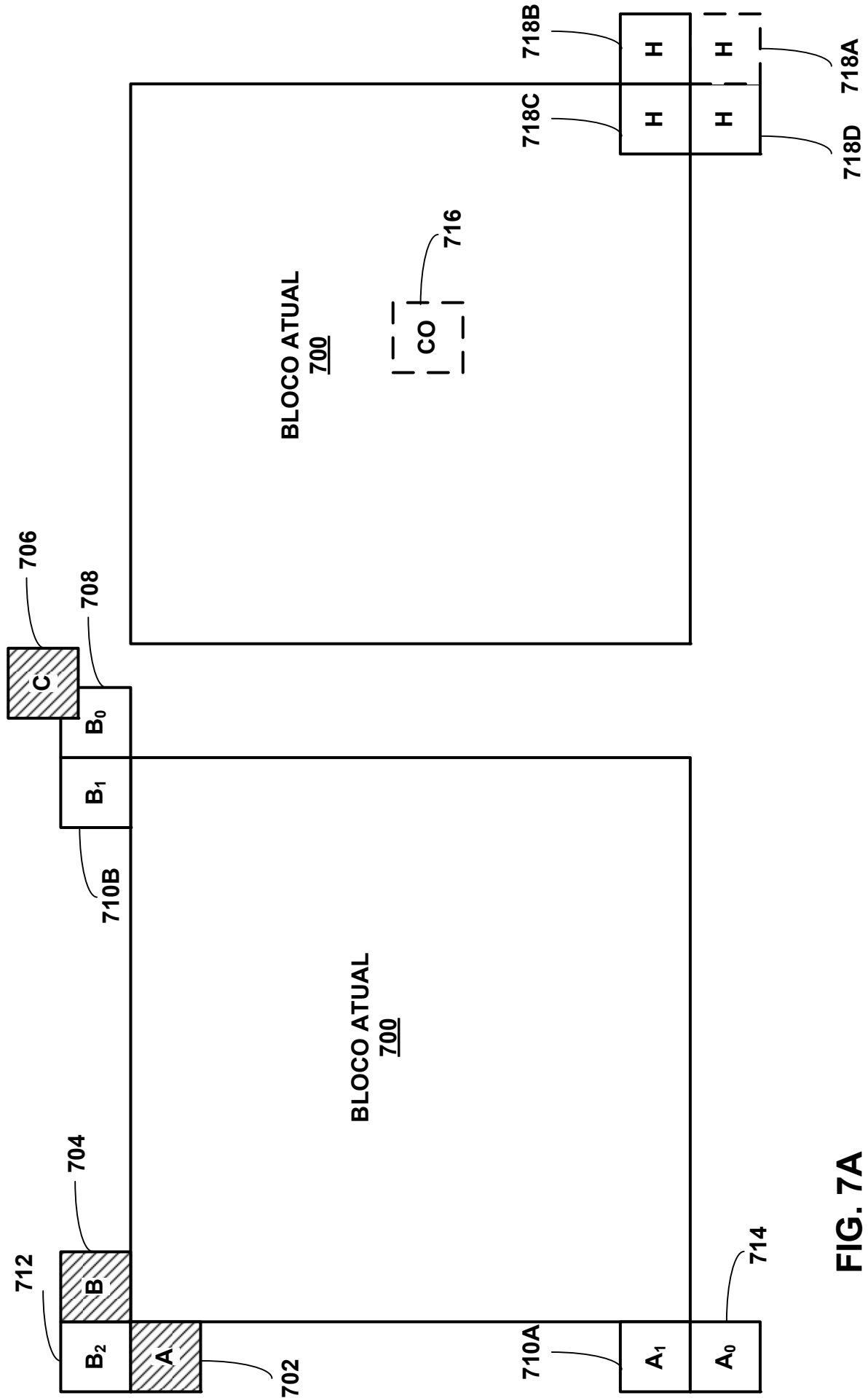


FIG. 7A

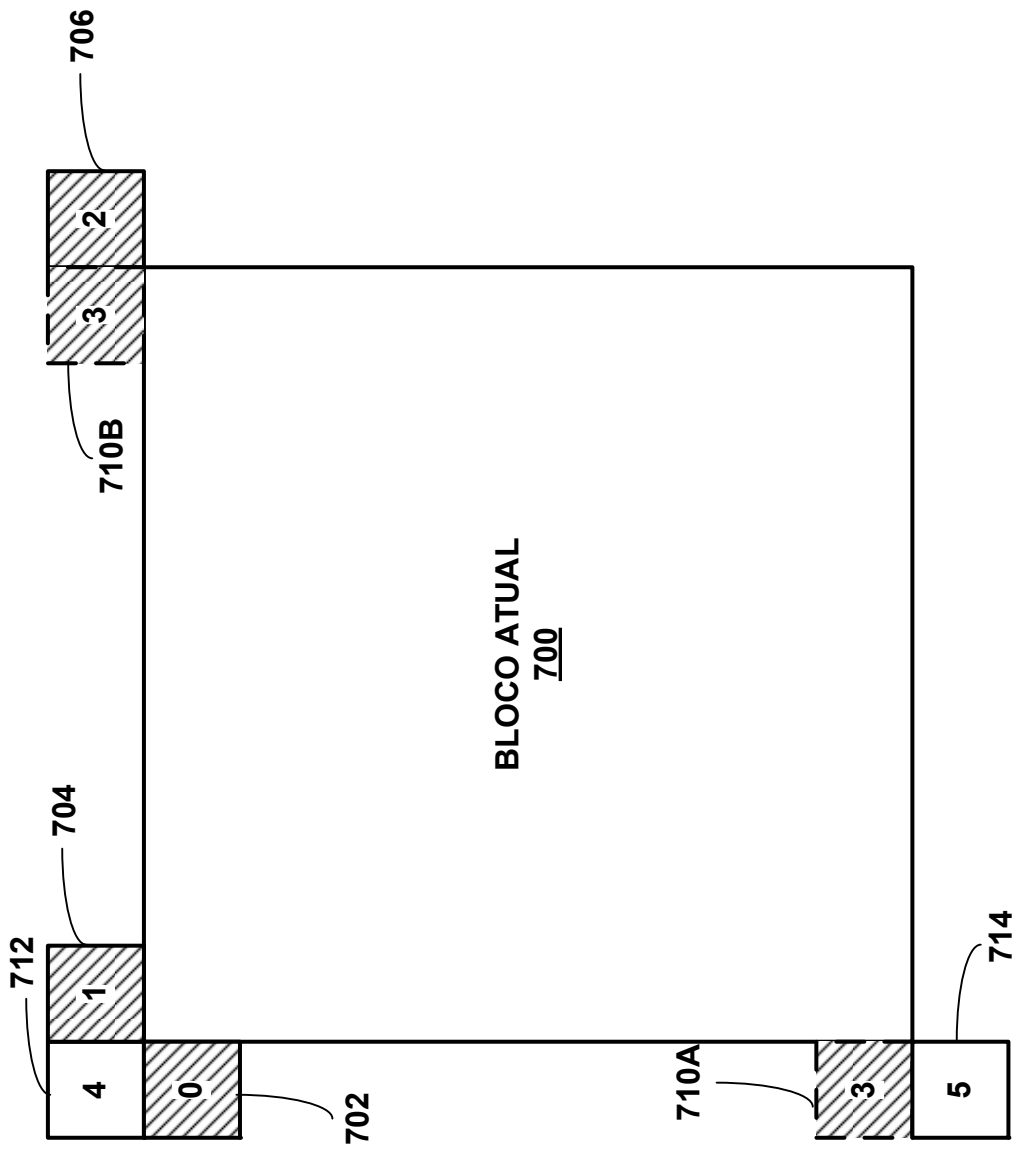


FIG. 7B

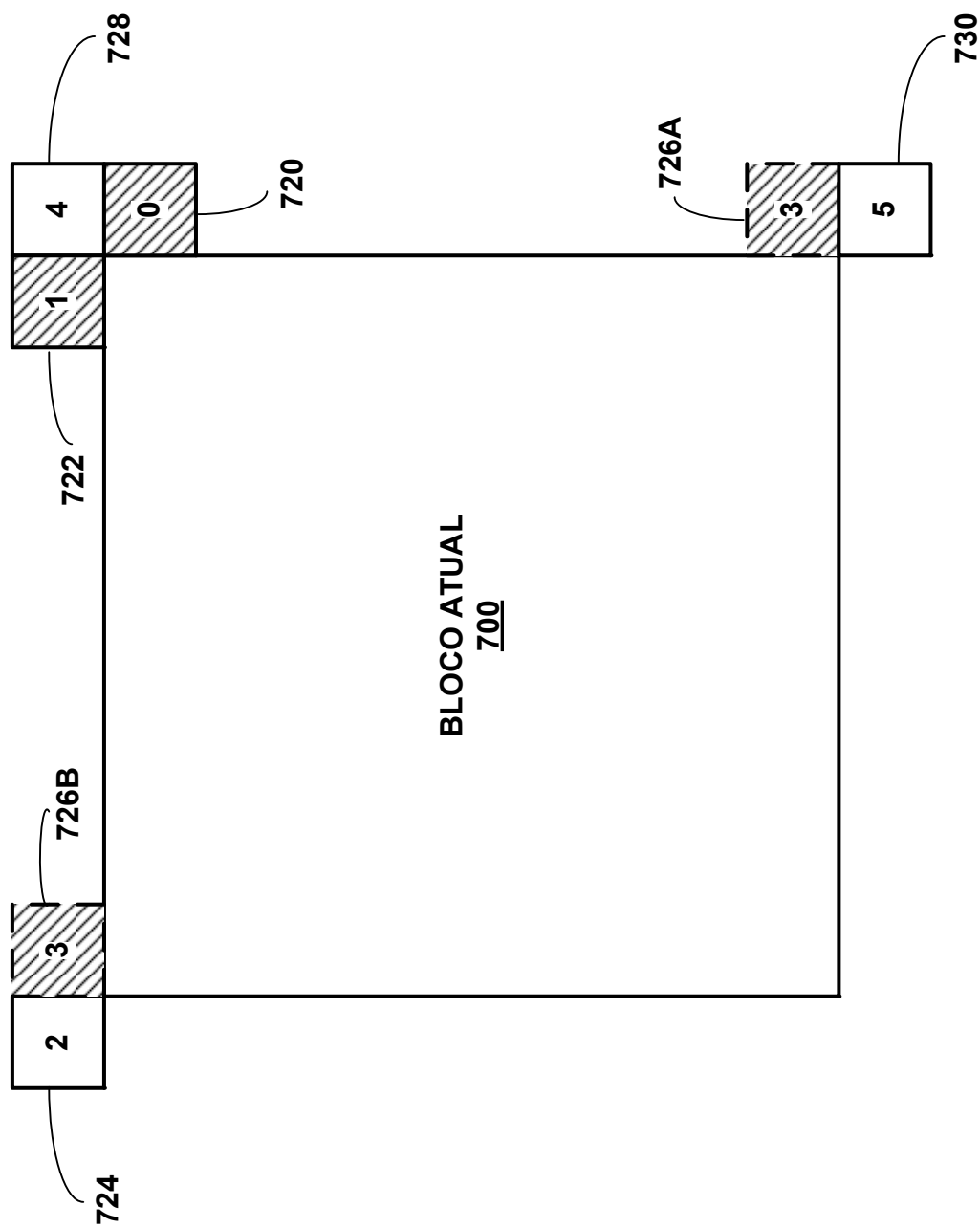


FIG. 7C

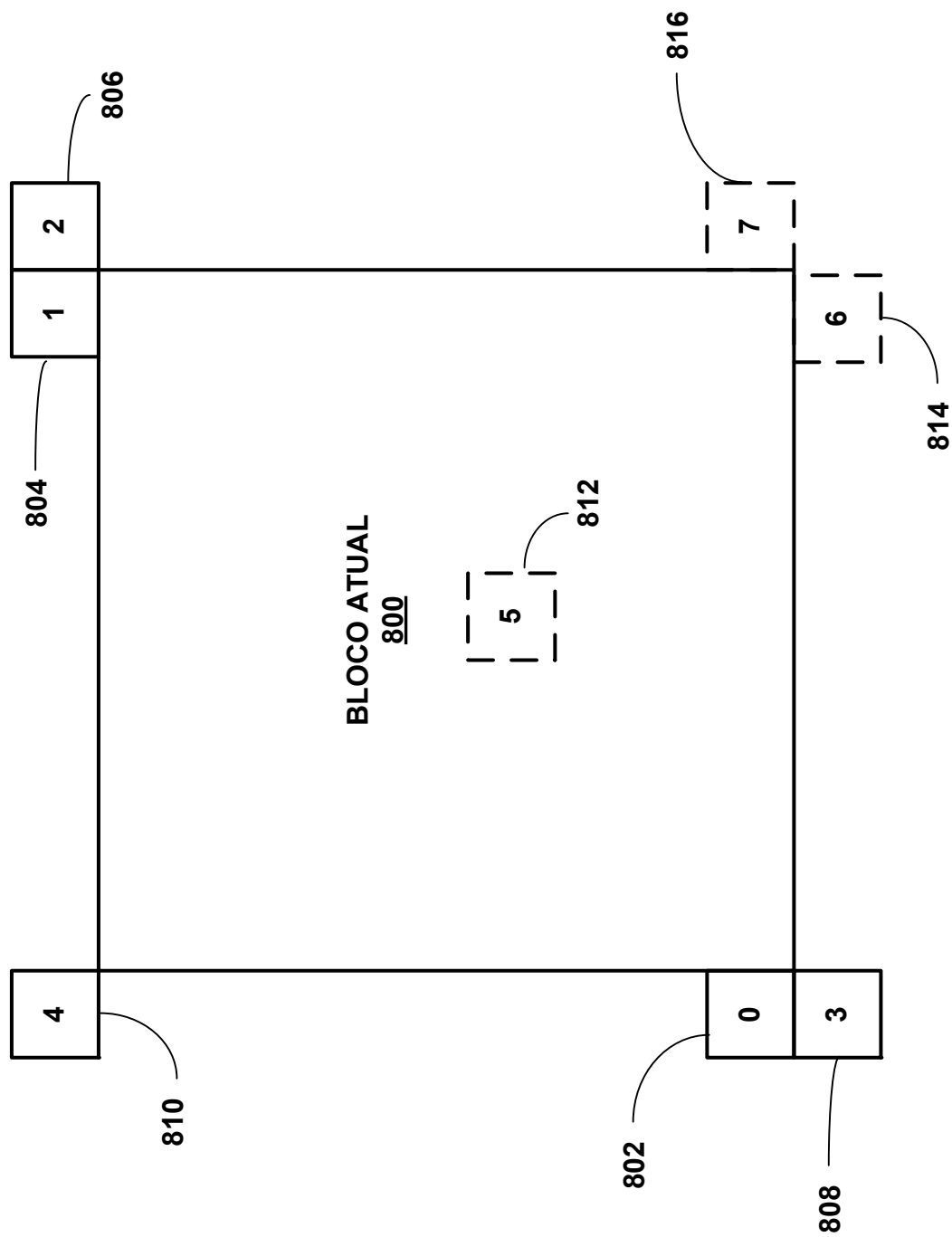


FIG. 8A

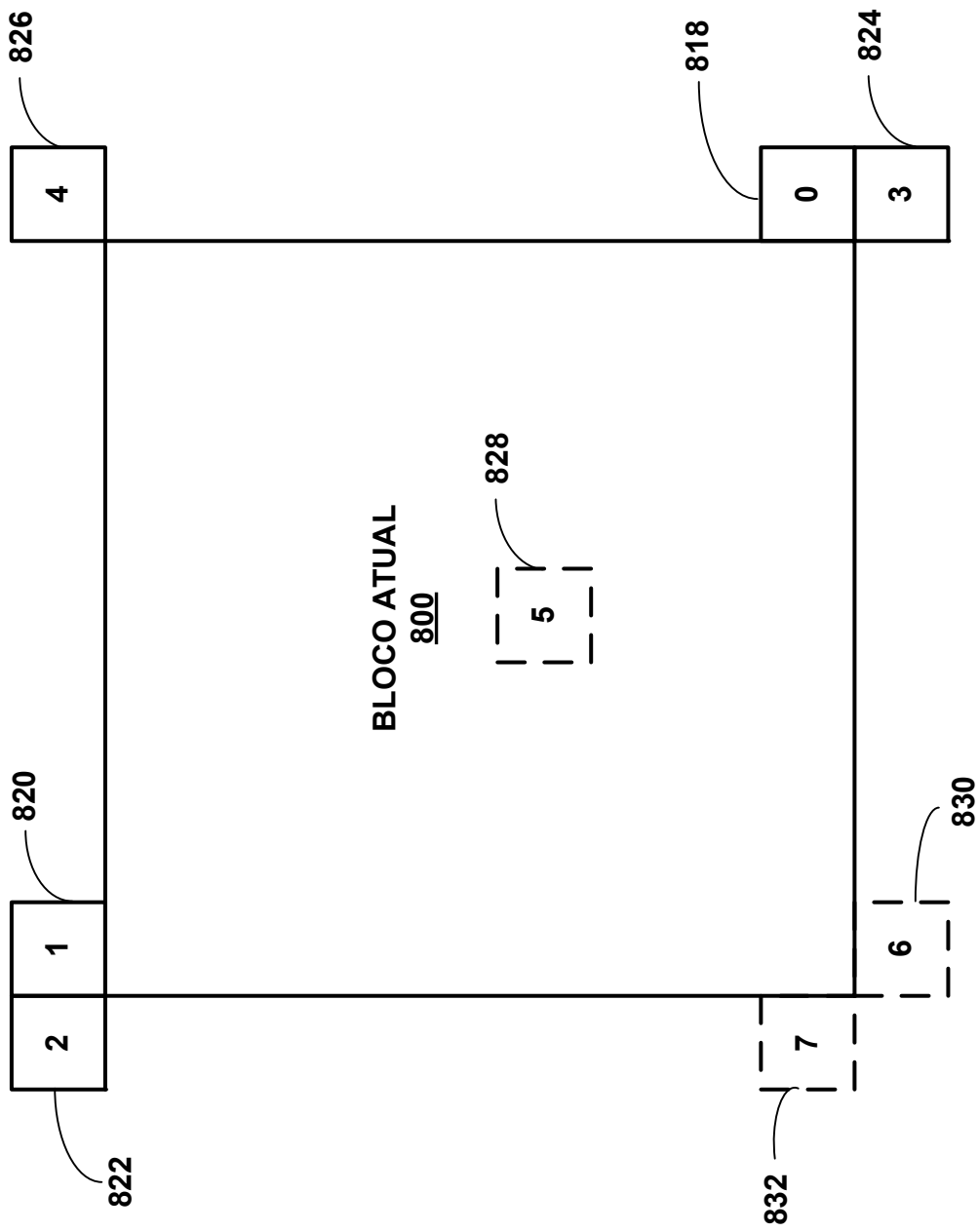


FIG. 8B

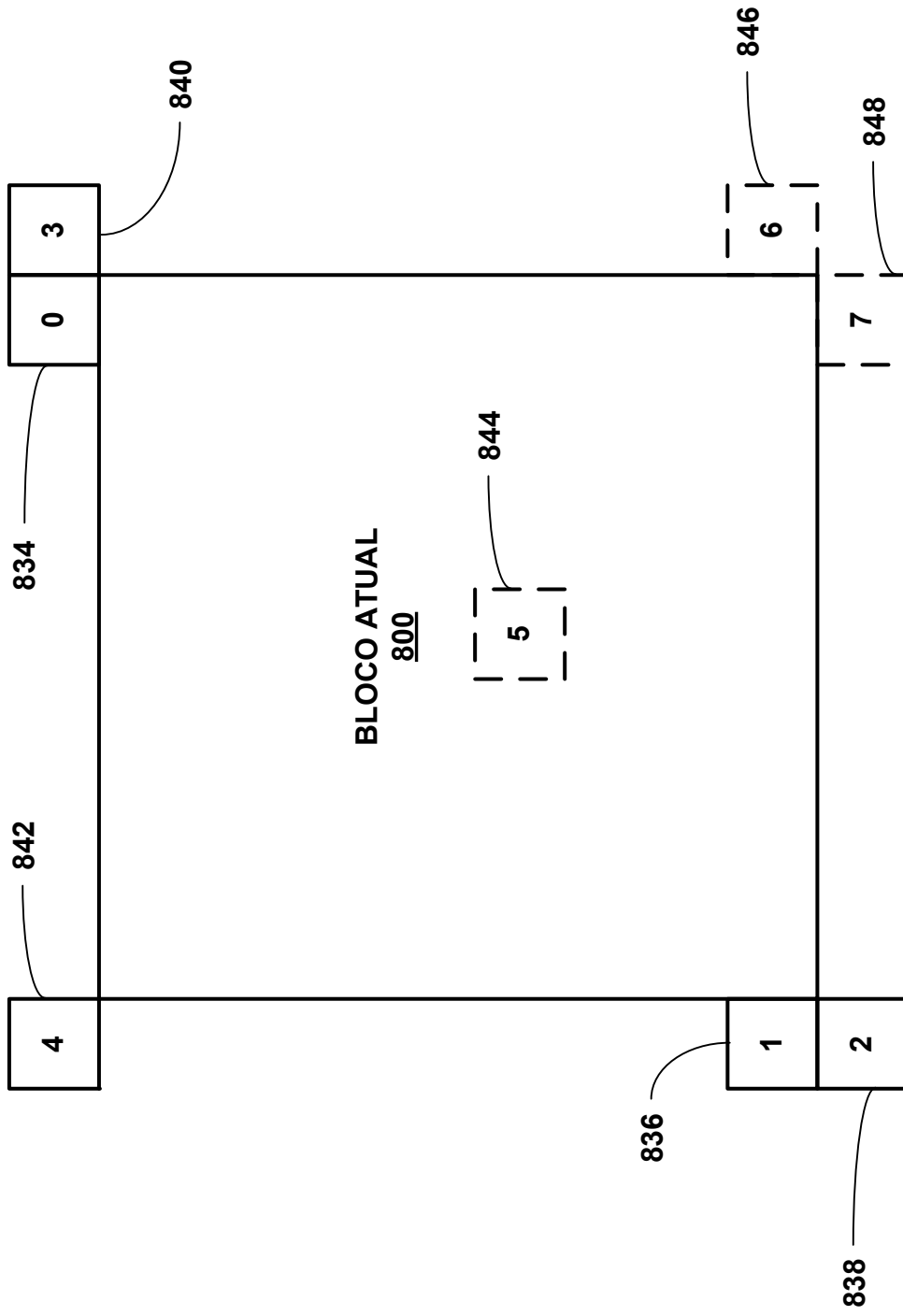


FIG. 8C

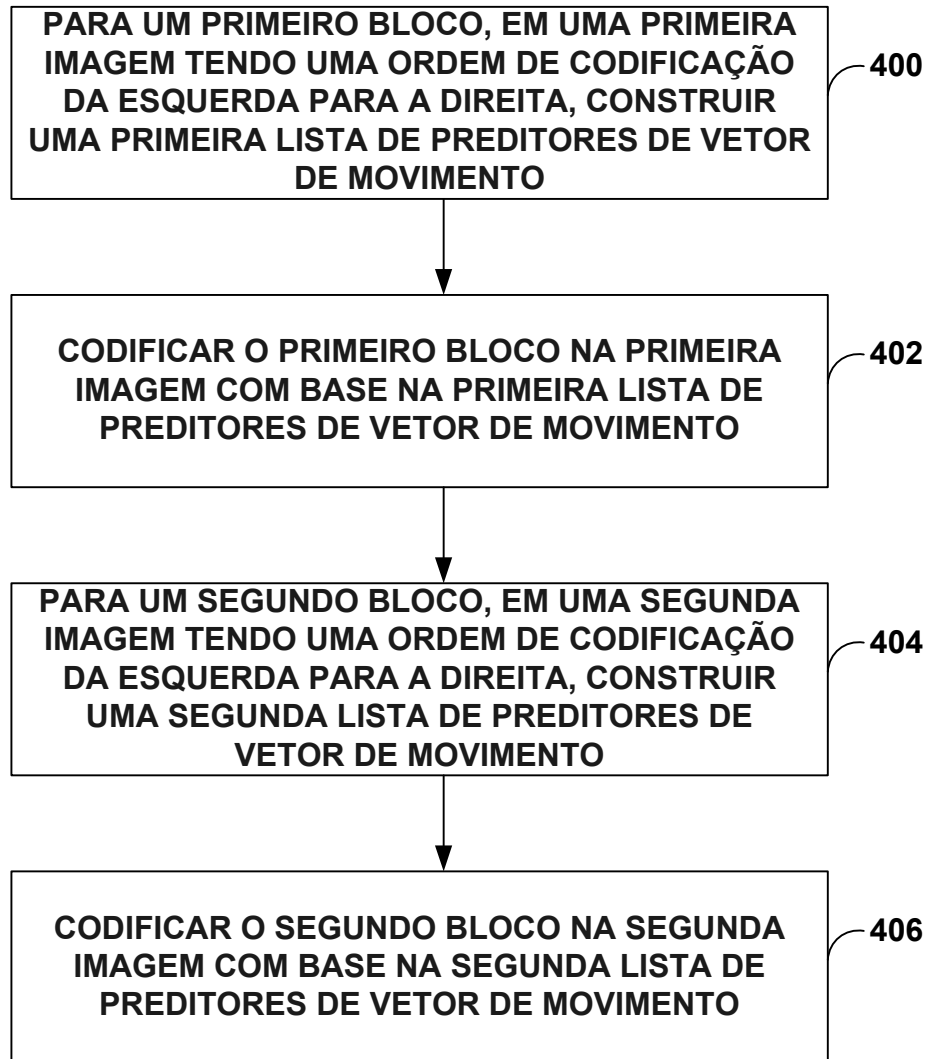


FIG. 9

RESUMO**"PADRÕES DE PREDIÇÃO DE VETOR DE MOVIMENTO ESPAÇO-TEMPORAL
PARA CODIFICAÇÃO DE VÍDEO"**

A presente invenção refere-se a técnicas para construir listas de preditores de vetor de movimento com base em blocos espacialmente vizinhos e blocos colocalizados. Um método de codificar dados de vídeo inclui, para um primeiro bloco, em uma primeira imagem tendo uma ordem de codificação da esquerda para a direita, construir uma primeira lista de preditores de vetor de movimento, em que uma primeira entrada na primeira lista de preditores de vetor de movimento baseia-se em informações de vetor de movimento de um bloco vizinho à esquerda do primeiro bloco, codificar o primeiro bloco na primeira imagem com base na primeira lista de preditores de vetor de movimento, para um segundo bloco, em uma segunda imagem tendo uma ordem de codificação da direita para a esquerda, construir uma segunda lista de preditores de vetor de movimento, em que uma primeira entrada na segunda lista de preditores de vetor de movimento baseia-se em informações de vetor de movimento de um bloco vizinho à direita do segundo bloco, e codificar o segundo bloco na segunda imagem com base na segunda lista de preditores de vetor de movimento.