



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

(21) BR 112016008358-0 A2



(22) Data do Depósito: 19/09/2014

(43) Data da Publicação Nacional: 03/08/2021

(54) Título: CANDIDATOS A FUSÃO BI-PREDITIVA COMBINADA PARA CODIFICAÇÃO DE VÍDEO 3D

(51) Int. Cl.: H04N 19/597; H04N 19/56; H04N 19/52; H04N 19/70; H04N 19/577.

(30) Prioridade Unionista: 18/09/2014 US 14/489,679; 20/09/2013 US 61/880,737.

(71) Depositante(es): QUALCOMM INCORPORATED.

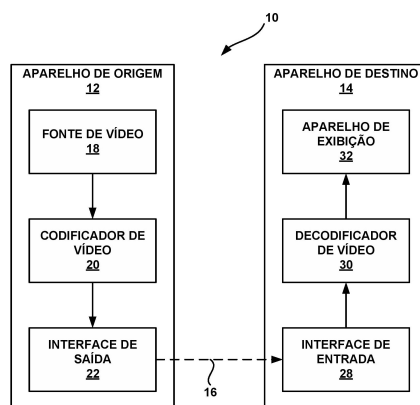
(72) Inventor(es): LI ZHANG; YING CHEN.

(86) Pedido PCT: PCT US2014056557 de 19/09/2014

(87) Publicação PCT: WO 2015/042399 de 26/03/2015

(85) Data da Fase Nacional: 21/03/2016

(57) Resumo: CANDIDATOS À FUSÃO BI-PREDITIVA COMBINADA PARA CODIFICAÇÃO DE VÍDEO 3D. Um codificador de vídeo gera uma lista de candidatos à fusão para codificar um bloco de vídeo de dados de vídeo tridimensionais (3D). O número máximo de candidatos à fusão na lista de candidatos à fusão pode ser igual a 6. Como parte da geração da lista de candidatos à fusão, o codificador de vídeo determina se o número de candidatos à fusão na lista de candidatos à fusão é menor que 5. Se este for o caso, o codificador de vídeo deriva um ou mais candidatos à fusão bi-preditiva combinada. O codificador de vídeo inclui o candidato ou candidatos à fusão bi-preditiva combinada na lista de candidatos à fusão.



CANDIDATOS À FUSÃO BI-PREDITIVA COMBINADA PARA CODIFICAÇÃO
DE VÍDEO 3D

[0001] Este pedido reivindica o benefício do pedido de patente provisório norte-americano No. 61/880 737, depositado a 20 de setembro de 2013 cujo conteúdo inteiro é aqui incorporado à guisa de referência.

CAMPO TÉCNICO

[0002] Esta revelação refere-se à codificação e compactação de vídeo e, mais especificamente, a técnicas de codificação que podem ser utilizadas na codificação de vídeo tridimensional (3D).

ANTECEDENTES

[0003] Capacidades de vídeo digital podem ser incorporadas a uma ampla faixa de aparelhos, que incluem televisões digitais, sistemas de broadcast direto digitais, sistemas de broadcast sem fio, assistentes digitais pessoais (PDAs), computadores laptop ou de mesa, câmeras digitais, aparelhos de gravação digitais, tocadores de meios digitais, aparelhos para jogos de vídeo, consoles para jogos de vídeo, telefones celulares ou de rádio-satélite, aparelhos de teleconferência de vídeo e semelhante. Os aparelhos de vídeo digital implementam técnicas de compactação de vídeo, tais como as descritas nos padrões definidos pelo MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificação Avançada de Vídeo (AVC), o padrão de Codificação de Vídeo de Alta Eficácia (HEVC) e extensões de tais padrões, para transmitir, receber, e armazenar informações de vídeo digital de maneira mais eficaz.

[0004] As técnicas de compactação de vídeo efetuam predição espacial (intra-imagem) e/ou predição temporal (inter-imagem) para reduzir ou remover a redundância inerente a sequências de vídeo. Para

codificação de vídeos baseada em blocos, uma fatia de vídeo pode ser particionada em blocos de vídeo que podem ser também referidos como blocos de árvore, unidade de codificação (CUs) e/ou nós de codificação. Os blocos de vídeo em uma fatia intra-codificada (I) de uma imagem são codificados utilizando-se predição espacial com relação a amostras de referência em blocos vizinhos na mesma imagem. Os blocos de vídeo em uma fatia inter-codificada (P ou B) de uma imagem podem utilizar predição espacial com relação às amostras de referência em blocos vizinhos na mesma imagem ou fatia ou predição temporal com relação 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.

[0005] Um fluxo de bits de codificação de várias vistas pode ser gerado por vistas de codificação, de várias perspectivas, por exemplo. A codificação de várias vistas pode permitir que um decodificador escolha entre vistas diferentes ou renderize possivelmente várias vistas. Além do mais, algumas técnicas e padrões de vídeo tridimensionais (3D) que foram desenvolvidos ou que estão em desenvolvimento, fazem uso de aspectos de codificação de várias vistas. O vídeo tridimensional é também referido como "3DV".

[0006] Por exemplo, vistas diferentes podem transmitir vistas de olho esquerdo e direito para suportar vídeo 3D. Alternativamente, alguns processos de codificação 3D podem aplicar a chamada codificação de várias vistas mais profundidade. Na codificação de várias vistas mais profundidade, um fluxo de bits 3D pode conter não só componentes de vista de textura, mas também componentes de vista de profundidade. Por exemplo, cada vista pode

compreender um componente de vista de textura e um componente de vista de profundidade.

[0007] Atualmente, uma Equipe de Colaboração Conjunta em Codificação de Vídeo 3D (JCT-3C) do VCEG e MPEG está desenvolvendo um padrão de vídeo 3D com base no padrão emergente referido como "codificação de vídeo de alta eficácia (HEVC)", para o qual parte dos esforços de padronização inclui a padronização do codec de vídeo de várias vistas baseada na HEVC (MV-HEVC) e outra parte para codificação de vídeo 3D baseada na HEVC (3D-HEVC). 3D-HEVC podem incluir e suportar novas ferramentas de codificação, inclusive aquelas no nível de unidade de codificação/unidade de predição, para vistas tanto de textura quanto de profundidade.

SUMÁRIO

[0008] Em geral, esta revelação refere-se à codificação de vídeo tridimensional (3D) baseadas em codecs avançados, inclusive a codificação de duas ou mais vistas com o codec de Codificação de Vídeo de Alta Eficácia (HEVC) - 3D. Por exemplo, alguns exemplos desta revelação descrevem técnicas relacionadas com candidatos à fusão bi-preditiva combinada. Em alguns de tais exemplos, como parte da geração de uma lista de candidatos à fusão, um codificador de vídeo determina se o número de candidatos à fusão na lista é menor que 5. Se este for o caso, o codificador de vídeo deriva um ou mais candidatos à fusão bi-preditiva combinada. O codificador de vídeo inclui o candidato ou candidatos à fusão bi-preditiva combinada na lista de candidatos à fusão.

[0009] Sob um aspecto, esta revelação descrita um método para codificar dados de vídeo 3D. O método compreende gerar uma lista de candidatos à fusão para codificar um bloco de vídeo de dados de vídeo 3D. O número

máximo de candidatos à fusão na lista de candidatos à fusão é igual a 6 e gerar a lista de candidatos à fusão compreende: determinar se o número de candidatos à fusão na lista de candidatos à fusão é menor que 5; e, em resposta, à determinação de que o número de candidatos à fusão na lista de candidatos à fusão é menor que 5: derivar um ou mais candidatos à fusão bi-preditiva combinada, em que cada respectivo candidato à fusão bi-preditiva combinada do candidato ou candidatos à fusão bi-preditiva combinada corresponde a um respectivo par de candidatos à fusão já na lista de candidatos à fusão, em que o respectivo candidato à fusão bi-preditiva combinada é uma combinação de um vetor de movimento de um primeiro candidato à fusão de um respectivo par e um vetor de movimento de um segundo candidato à fusão do respectivo par, em que o vetor de movimento do primeiro candidato à fusão e o vetor de movimento do segundo candidato à fusão referem-se à imagens em listas de imagens de referência diferentes. O método compreende também incluir o candidato ou candidatos à fusão bi-preditiva combinada na lista de candidatos à fusão.

[0010] Sob outro aspecto, esta revelação descreve um aparelho de codificação de vídeo que compreende um meio de armazenamento de dados configurado para armazenar dados de vídeo 3D; e um ou mais processadores configurados para: gerar uma lista de candidatos à fusão para codificar um bloco de vídeo dos dados de vídeo 3D, no qual o número máximo de candidatos à fusão na lista de candidatos à fusão é igual a 6 e, como parte da geração da lista de candidatos à fusão, o processador ou processadores: determinam se o número de candidatos à fusão na lista de candidatos à fusão é menor que 5; e, em resposta à determinação de que um número de candidatos à fusão na lista de candidatos à fusão é menor que 5: derivam

um ou mais candidatos à fusão bi-preditiva combinada, no qual cada respectivo candidato à fusão bi-preditiva combinada do candidato ou candidatos à fusão bi-preditiva combinada corresponde a um respectivo par de candidatos à fusão já na lista de candidatos à fusão, no qual o respectivo candidato à fusão bi-preditiva combinada é uma combinação de um vetor de movimento de um primeiro candidato à fusão do respectivo par e um vetor de movimento do segundo candidato à fusão do respectivo par no qual o vetor de movimento do primeiro candidato à fusão e o vetor de movimento do segundo candidato à fusão referem-se a imagens em listas de imagens de referência diferentes. O processador ou processadores são configurados para incluir o candidato ou candidatos à fusão na lista de candidatos à fusão.

[0011] Sob outro aspecto, esta revelação descreve um aparelho de codificação de vídeo que compreende: um dispositivo para gerar uma lista de candidatos à fusão para codificar um bloco de vídeo dos dados de vídeo 3D. O número máximo de candidatos à fusão na lista de candidatos à fusão é igual a 6 e o dispositivo para gerar a lista de candidatos à fusão compreende: um dispositivo para determinar se o número de candidatos à fusão na lista de candidatos à fusão é menor que 5; um dispositivo para derivar, em resposta à determinação de que um número de candidatos à fusão na lista de candidatos à fusão é menor que 5, um ou mais candidatos à fusão bi-preditiva combinada, no qual cada respectivo candidato à fusão bi-preditiva combinada do candidato ou candidatos à fusão bi-preditiva combinada corresponde a um respectivo par de candidatos à fusão já na lista de candidatos à fusão, no qual o respectivo candidato à fusão bi-preditiva combinada é uma combinação de um vetor de movimento de um

primeiro candidato à fusão do respectivo par e um vetor de movimento do segundo candidato à fusão do respectivo par no qual o vetor de movimento do primeiro candidato à fusão e o vetor de movimento do segundo candidato à fusão referem-se a imagens em listas de imagens de referência diferentes. O aparelho de codificação de vídeo compreende também um dispositivo para incluir o candidato ou candidatos à fusão bi-preditiva combinada na lista de candidatos à fusão.

[0012] Sob outro aspecto, esta revelação descreve um meio de armazenamento de dados passível de leitura por computador que tem, instruções armazenadas nele, que, quando executadas, fazem com que um aparelho de codificação de vídeo para dados de vídeo, 3D, as instruções fazendo com que o aparelho de codificação de vídeo: gere uma lista de candidatos à fusão para codificar um bloco de vídeo dos dados de vídeo 3D. O número máximo de candidatos à fusão na lista de candidatos à fusão é igual a 6. Gerar a lista de candidatos à fusão compreende: determinar se o número de candidatos à fusão na lista de candidatos à fusão é menor que 5; e, em resposta à determinação de que um número de candidatos à fusão na lista de candidatos à fusão é menor que 5: derivar um ou mais candidatos à fusão bi-preditiva combinada, no qual cada respectivo candidato à fusão bi-preditiva combinada do candidato ou candidatos à fusão bi-preditiva combinada corresponde a um respectivo par de candidatos à fusão já na lista de candidatos à fusão, no qual o respectivo candidato à fusão bi-preditiva combinada é uma combinação de um vetor de movimento de um primeiro candidato à fusão do respectivo par e um vetor de movimento do segundo candidato à fusão do respectivo par no qual o vetor de movimento do primeiro candidato à fusão e o vetor de movimento do segundo candidato à fusão referem-se a imagens em listas de imagens de referência diferentes; e

incluir o candidato ou candidatos à fusão bi-preditiva combinada na lista de candidatos à fusão.

[0013] Os detalhes de um ou mais exemplos são apresentados nos desenhos anexos e na descrição que se segue. Outros recursos, objetos e vantagens serão evidentes a partir da descrição, dos desenhos e das reivindicações.

DESCRIÇÃO RESUMIDA DOS DESENHOS

[0014] A Figura 1 é um diagrama de blocos que mostra um sistema de codificação de vídeo exemplar que pode utilizar as técnicas desta revelação.

[0015] A Figura 2 é uma ilustração conceitual que mostra vizinhos espaciais que são os candidatos potenciais para uma lista de fusão.

[0016] A Figura 3 é um diagrama conceitual que mostra blocos vizinhos espaciais e temporais relativos à unidade de condição atual.

[0017] A Figura 4 mostra um exemplo de um processo de derivação de um candidato a vetor de movimento predito por inter-vista.

[0018] A Figura 5 é um diagrama conceitual que mostra derivação de blocos de profundidade de uma vista de referência para efetuar predição por síntese de vista para trás-deformação (PVSP).

[0019] A Figura 6 é um diagrama conceitual que mostra quatro pixels de canto do bloco de profundidade 8x8.

[0020] A Figura 7 é uma tabela que apresenta uma especificação exemplar do 10CandIdx e 11CandIdx na 3D-HEVC.

[0021] A Figura 8 é um diagrama de blocos que mostra um codificador de vídeo exemplar que pode implementar as técnicas desta revelação.

[0022] A Figura 9 é um diagrama de blocos que mostra um decodificador de vídeo exemplar que pode implementar as técnicas desta revelação.

[0023] A Figura 10A é um fluxograma que mostra uma operação exemplar de um codificador de vídeo para codificar dados associados a vídeo 3D, de acordo com uma ou mais técnicas desta revelação.

[0024] A Figura 10B é um fluxograma que mostra uma operação exemplar de um decodificador de vídeo para decodificar dados associados a vídeo 3D, de acordo com algumas técnicas desta revelação.

[0025] A Figura 11 é um fluxograma que mostra uma primeira parte de uma operação exemplar para construir uma lista de candidatos à fusão, de acordo com uma ou mais técnicas desta revelação.

[0026] A Figura 12 é um fluxograma que mostra uma segunda parte da operação exemplar da Figura 11 para construir uma lista de candidatos à fusão para o bloco atual, de acordo com uma ou mais técnicas desta revelação.

[0027] A Figura 13 é um fluxograma que mostra um processo de derivação exemplar para candidatos à fusão bi-preditiva combinada, de acordo com uma ou mais técnicas desta revelação.

[0028] A Figura 14A é um fluxograma que mostra uma operação exemplar de um codificador de vídeo para codificar um bloco de vídeo, de acordo com uma ou mais técnicas desta revelação.

[0029] A Figura 14B é um fluxograma que mostra uma operação exemplar de um decodificador de vídeo para decodificar um bloco de vídeo, de acordo com uma ou mais técnicas desta revelação.

DESCRIÇÃO DETALHADA

[0030] A codificação de vídeo é um processador de transformar dados de vídeo em dados de vídeo codificados. Em geral, a decodificador de vídeo inverte a transformação, reconstruindo assim os dados de vídeo. A codificação de vídeo e a decodificação de vídeo podem ser ambas referidas como codificação de vídeo. A codificação de vídeo baseada em blocos é um tipo de codificação de vídeo que funciona, pelo menos em parte, em blocos de dados de vídeo dentro de imagens.

[0031] A inter-predição é uma técnica de codificação de vídeo na qual um codificador de vídeo determina, com base em amostras de uma imagem de referência, um bloco preditivo para o bloco atual (isto é, um bloco que o codificador de vídeo está codificando atualmente). A imagem de referência é uma imagem que não a imagem que o codificador de vídeo está codificando atualmente. O codificador de vídeo pode incluir, em um fluxo de bits, dados que representam dados residuais para o bloco. Os dados residuais para o bloco indicam diferenças entre o bloco atual e o bloco preditivo. Um vetor de movimento para o bloco pode indicar um deslocamento espacial entre o bloco atual e o bloco preditivo. Um índice de referência pode indicar a localização da imagem de referência dentro uma lista de imagens de referência disponíveis para utilização na codificação da imagem atual. Os índices de referência podem ser também referidos como "índices de imagens de referência". Um decodificador de vídeo pode utilizar o vetor de movimento do bloco atual para determinar o bloco preditivo para o bloco atual. Além disto, o decodificador de vídeo pode combinar o bloco preditivo com os dados residuais para o bloco atual de modo a reconstruir o bloco atual.

[0032] Na inter-predição bidirecional, o codificador de vídeo determina dois blocos preditivos para o bloco atual. Por conseguinte, o codificador de vídeo determina também dois vetores de movimento para o bloco atual. Os dois blocos preditivos para o bloco atual podem estar em imagens de referência diferentes. Consequentemente, na inter-predição bidirecional, o codificador de vídeo pode determinar dois índices de referência para o bloco atual (isto é, um primeiro índice de referência e um segundo índice de referência). Os primeiro e segundo índices de referência indicam as localizações de imagens de referência dentro de uma primeira e uma segunda listas de imagens de referência, respectivamente. Os dados residuais para o bloco atual pode indicar diferenças entre o bloco atual e um bloco preditivo sintetizado que é baseado nos dois blocos preditivos para o bloco atual.

[0033] Os vetores de movimento do bloco atual podem ser semelhantes aos vetores de movimento de blocos que são espacial ou temporalmente vizinhos ao bloco atual (isto é, blocos vizinhos). Consequentemente, pode ser desnecessário que um codificador de vídeo sinalize explicitamente os vetores de movimento e os índices de referência do bloco atual. Em vez disso, o codificador de vídeo pode determinar uma lista de candidatos à fusão para o bloco atual (isto é, uma "lista de candidatos à fusão"). Cada um dos candidatos à fusão especifica um conjunto de informações de movimento (como, por exemplo, um ou mais vetores de movimento, um ou mais índices de referência, etc.). A lista de candidatos à fusão pode incluir um ou mais candidatos à fusão que especificam respectivamente informações de movimento de blocos diferentes dos blocos vizinhos. Os blocos vizinhos podem incluir blocos vizinhos

espaciais e/ou blocos vizinhos temporais. Esta revelação pode referir-se a candidatos à fusão com base em blocos vizinhos espaciais como candidatos à fusão espaciais. Esta revelação pode referir-se a candidatos à fusão com base em blocos vizinhos temporais como candidatos à fusão temporais. Em alguns exemplos, dois candidatos à fusão na lista de candidatos à fusão podem ter informações de movimento idênticas. O codificador de vídeo pode selecionar um dos candidatos à fusão e pode sinalizar um elemento de sintaxe que indica uma posição dentro da lista de candidatos à fusão do candidato à fusão selecionado.

[0034] O decodificador de vídeo pode gerar a mesma lista de candidatos à fusão (isto é, uma lista de candidatos à fusão duplicadora da lista de candidatos à fusão determinada pelo codificador de vídeo) e pode determinar, com base no recebimento do elemento de sintaxe sinalizado, o candidato à fusão selecionado. O decodificador de vídeo pode utilizar então as informações de movimento do candidato à fusão selecionado como as informações de movimento do bloco atual. Desta maneira, o bloco atual pode herdar as informações de movimento de um dos blocos vizinhos.

[0035] Em algumas circunstâncias, as informações de movimento de um bloco vizinho podem não estar disponíveis. Por exemplo, o bloco vizinho pode ser codificado utilizando-se intra-predição, o bloco vizinho pode estar em uma fatia diferente, ou o bloco vizinho pode simplesmente não existir. Consequentemente, pode haver menos que o número necessário de candidatos à fusão (o número máximo de candidatos à fusão, que pode ser indicado em um cabeçalho de fatia, por exemplo) na lista de candidatos à fusão para o bloco atual. Por conseguinte, quando um codificador de vídeo (um codificador de vídeo ou

um decodificador de vídeo, por exemplo) gera a lista de candidatos à fusão para o bloco atual, o codificador de vídeo pode assegurar que a lista de candidatos à fusão para o bloco atual inclua o número desejado de candidatos à fusão pela inclusão de um ou mais candidatos à fusão artificiais na lista de candidatos à fusão para o bloco atual. Os candidatos à fusão artificiais são candidatos à fusão que não especificam necessariamente as informações de movimento de qualquer bloco vizinho espacial ou temporal.

[0036] Os candidatos à fusão artificiais podem incluir um ou mais candidatos à fusão bi-preditiva combinada. Conforme indicado acima, um candidato à fusão pode especificar dois vetores de movimento e dois índices de referência. Um candidato à fusão bi-preditiva combinada corresponde a um respectivo par de candidatos à fusão já na lista de candidatos à fusão para o bloco atual. Especificamente, o candidato à fusão bi-preditiva combinada é uma combinação de vetor de movimento e índice de referência de um primeiro candidato à fusão do respectivo par, se disponível, e vetor de movimento e índice de referência de um segundo candidato à fusão do respectivo par, se disponível. O vetor de movimento do primeiro candidato à fusão e o vetor de movimento do segundo candidato à fusão referem-se a imagens em listas de imagens de referência diferentes. Assim, os candidatos à fusão bi-preditiva combinada podem corresponder a combinações diferentes de vetores de movimento/índices de referência de candidatos à fusão existentes diferentes (outros candidatos à fusão que não candidatos à fusão bi-preditiva combinados, tais como candidatos à fusão especiais ou temporais, por exemplo). Por exemplo, quando as informações de movimento RefPicList0 de um primeiro candidato à fusão e as informações de movimento RefPicList1 de um segundo

candidato à fusão estão ambas disponíveis e não são idênticas (isto é, ou as imagens de referência são diferentes ou os vetores de movimento são diferentes) um candidato à fusão bi-preditiva combinado é construído um candidato à fusão bi-preditiva combinada. Caso contrário é verificado o respectivo par seguinte.

[0037] Em algumas versões da especificação HEVC, o valor máximo do número necessário de candidatos à fusão em uma lista de candidatos à fusão é 5. Além disto, em alguns casos, o número desejado de candidatos à fusão em uma lista de candidatos à fusão é 5. Consequentemente, se houver menos que 5 candidatos à fusão na lista de candidatos à fusão antes da inclusão de candidatos à fusão bi-preditiva combinada na lista de candidatos à fusão, há até 12 (isto é, 4×3) combinações possíveis de vetores de movimento utilizáveis nos candidatos à fusão bi-preditiva combinada. A seleção de um respectivo par (isto é, qual candidato é o primeiro candidato é o segundo candidato) é pré-definida na HEVC, conforme mostrado na tabela seguinte:

| combIdx | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|
| l0CandIdx | 0 | 1 | 0 | 2 | 1 | 2 | 0 | 3 | 1 | 3 | 2 | 3 |
| l1CandIdx | 1 | 0 | 2 | 0 | 2 | 1 | 3 | 0 | 3 | 1 | 3 | 2 |

Na tabela acima, l0CandIdx representa o índice do primeiro candidato à fusão existente selecionado, l1CandIdx representa o índice do segundo candidato à fusão existente e combIdx representa o índice de candidato bi-preditivo combinado construído.

[0038] A codificação de vídeo de várias camadas permite a codificação de vídeo através de várias camadas. A codificação de vídeo de várias camadas pode ser utilizada para implementar codificação escalonável de vídeo, codificação de vídeo de várias camadas e codificação

de vídeo 3-dimensional (3D). Na codificação de vídeo de várias vistas e na codificação de vídeo 3D, cada uma das camadas pode corresponder a um ponto de vista diferente. Em alguns padrões de codificação de vídeo, o número necessário de candidatos à fusão em uma lista de candidatos à fusão é maior quando se utiliza codificação de vídeo de várias camadas do que quando se utiliza codificação de vídeo de camada única. O número maior de candidatos à fusão pode ser permitido de modo a acomodar candidatos à fusão que especificam informações de movimento de blocos em vistas diferentes.

[0039] Como no caso da codificação de vídeo de camada única, quando um codificação de vídeo está utilizando codificação de várias camadas e o número de candidatos à fusão em um lista de candidatos à fusão é menor que o número desejado de candidatos à fusão, o codificação de vídeo pode gerar um ou mais candidatos à fusão bi-preditiva combinada. Entretanto, devido ao número maior de candidatos à fusão quando se utiliza codificação de várias camadas, há um número maior de combinações de vetores de movimento utilizáveis nos candidatos à fusão bi-preditiva combinada. Por exemplo, se o número necessário de candidatos à fusão for 6, há até 20 (5×4) combinações possíveis de vetores de movimento utilizáveis nos candidatos à fusão bi-preditiva combinada.

[0040] Um codificador de vídeo pode não ser capaz de gerar um candidato à fusão bi-preditiva combinada a partir de pares específicos de candidatos à fusão. Por exemplo, o codificador de vídeo pode não ser capaz de gerar um candidato à fusão bi-preditiva combinada se um dos candidatos à fusão tiver apenas um único vetor de movimento e um único índice de referência. De modo a determinar se um candidato à fusão bi-preditiva combinada pode ser gerado a

partir de informações de movimento de um par específico de candidatos à fusão, o codificador de vídeo pode precisar recuperar informações sobre o par de candidatos à fusão de uma memória.

[0041] A recuperação de informações da memória pode ser um processo comparativamente lento com relação a outros processos de codificação. Além do mais, o acesso à memória exige energia. Portanto, pode ser desejável a limitação do número de acessos à memória. À medida que aumenta o número de combinações de vetores de movimento utilizáveis em candidatos à fusão bi-preditiva combinada, aumenta a quantidade de informações que é necessário recuperar da memória. Assim o aumento no número necessário de candidatos à fusão associados à codificação de vídeo de várias vistas pode tornar significativamente lento o processo de codificação de vídeo e pode utilizar mais energia da que seria utilizada de outro modo.

[0041] Consequentemente, de acordo com o exemplo desta revelação, um codificação de vídeo pode gerar uma lista de candidatos à fusão para codificar um bloco de vídeo 3D de maneira que possa limitar os acessos à memória. Além disto, neste exemplo, como parte da geração da lista de candidatos à fusão, o codificador de vídeo pode determinar se o número de candidatos à fusão na lista é menor que 5. Em resposta à determinação de que o número de candidatos à fusão na lista é menor que 5, o codificação de vídeo pode derivar um ou mais candidatos à fusão bi-preditiva combinada. Neste exemplo, cada respectivo candidato à fusão bi-preditiva combinada do candidato ou candidatos à fusão broadcast corresponde a um respectivo par de candidatos à fusão já na lista. Além disto, neste exemplo, o respectivo candidato à fusão bi-preditiva combinada é uma combinação de vetor de movimento de um

primeiro candidato à fusão do respectivo par e vetor de movimento de um segundo candidato à fusão do respectivo par. Neste exemplo, o vetor de movimento do primeiro candidato à fusão e o vetor de movimento do segundo candidato à fusão referem-se a imagens em listas de imagens de referência diferentes. O codificador de vídeo pode incluir o candidato ou candidatos à fusão bi-preditiva combinada na lista. Em alguns exemplos, o número máximo de candidatos à fusão na lista é maior que 5 (igual a 6, por exemplo). Um efeito do processo deste exemplo é que o número de combinações permanece limitado a 12, embora o número máximo de candidatos à fusão na lista seja 6 ou mais. Isto pode ajudar a acelerar o processo de codificação pela redução da quantidade de informações recuperadas da memória e pode também economizar energia.

[0043] A Figura 1 é um diagrama de blocos que mostra um sistema de codificação de vídeo 10 exemplar que pode utilizar as técnicas desta revelação. Conforme aqui descrito, o termo "codificador de vídeo" refere-se genericamente tanto a codificadores de vídeo quanto a decodificadores de vídeo. Nesta revelação, os termos "codificação de vídeo" e "codificação" pode referir-se genericamente à codificação de vídeo ou decodificação de vídeo.

[0044] Conforme mostrado na Figura 1, o sistema de codificador de vídeo 10 inclui um aparelho de origem 12 e um aparelho de destino 14. O aparelho de origem 12 gera dados de vídeo codificados. Por conseguinte, o aparelho de origem 12 pode ser referido como aparelho de codificação de vídeo ou equipamento de codificação de vídeo. O aparelho de destino 14 pode decodificar os dados de vídeo codificados gerados pelo aparelho de origem 12. Por conseguinte, o aparelho de destino 14 pode ser referido

como aparelho de decodificação de vídeo ou equipamento de decodificação de vídeo. O aparelho de origem 12 e o aparelho de destino 14 podem ser exemplos de aparelhos de codificação de vídeo ou equipamentos de codificação de vídeo.

[0045] O aparelho de origem 12 e o aparelho de destino 14 podem compreender uma ampla faixa de aparelhos, que inclui computadores de mesa, aparelhos de computação móveis, computadores notebook (laptop, por exemplo) computadores tablet, conversores set-top-box, aparelhos telefônicos, tais como os chamados telefones "inteligentes", televisões, câmaras, aparelhos de exibição, tocadores de meios digitais, consoles para jogos de vídeo, computadores em carros ou semelhantes.

[0046] O aparelho de destino 14 pode receber os dados de vídeo codificados do aparelho de origem 12 por meio de um canal 16. O canal 16 pode compreender um ou mais meios ou aparelhos capazes de mover os dados de vídeo codificados do aparelho de origem 12 para o aparelho de destino 14. Em um exemplo, o canal 16 pode compreender um ou mais meios de comunicação que permitem que o aparelho de origem 12 transmita dados de vídeo codificados diretamente para o aparelho de destino 14 em tempo real. Neste exemplo, o aparelho de origem 12 pode modular os dados de vídeo codificados de acordo com um padrão de comunicação, tal como um protocolo de comunicação sem fio, e pode transmitir os dados de vídeo modulados para o aparelho de destino 14. O meio ou meios de comunicação podem incluir meios de comunicação sem fio ou cabeados, tal como um espectro de radiofrequência (RF) ou uma ou mais linhas de transmissão físicas. O meio ou meios de comunicação podem fazer parte de uma rede baseada e pacotes, tal como uma rede de área local, uma rede de área estendida ou uma rede

global, como a Internet. O canal 16 pode incluir diversos tipos de aparelho, tais como roteadores, comutadores, estações base ou outro equipamento que facilite a comunicação do aparelho de origem 12 para o aparelho de destino 14.

[0047] Em outro exemplo, o canal 16 pode incluir a um meio de armazenamento que armazena os dados de vídeo codificados gerados pelo aparelho de origem 12. Neste exemplo, o aparelho de destino 14 pode acessar o meio de armazenamento por meio de acesso a disco ou acesso a cartão. O meio de armazenamento pode incluir diversos meios de armazenamento de dados distribuídos ou acessados localmente, tais discos Blu-ray, DVDs, CD-ROMs, memória flash ou outros meios de armazenamento digitais para armazenar dados de vídeo codificados.

[0048] Em outro exemplo, o canal 16 pode incluir um servidor de arquivos ou outro aparelho de armazenamento intermediário que armazena dados de vídeo codificado gerados pelo aparelho de origem 12. Neste exemplo, o aparelho de destino 14 pode acessar dados de vídeo codificados armazenados no servidor de arquivos ou outro aparelho de armazenamento intermediário por meio de fluxo contínuo ou download. O servidor de arquivos pode ser um tipo de servidor capaz de armazenar dados de vídeo codificados e transmitir os dados de vídeo codificados para o aparelho de destino 14. Servidores de arquivos exemplares incluem servidores da Web (para um site da Web, por exemplo), servidores de protocolo de transferência de arquivos (FTP), aparelhos de armazenamento anexados à rede (NAS) e unidades de disco locais.

[0049] O aparelho de destino pode acessar os dados de vídeo codificados tal como uma conexão com a Internet. Tipos exemplares de conexão de dados podem

incluir canais sem fio (uma conexão Wi-Fi, por exemplo) conexões cabeadas (como, por exemplo, DSL, modem a cabo, etc.), ou uma combinação de ambos que sejam adequadas para acessar dados de vídeo codificados armazenados em um servidor de arquivos. A transmissão de dados de vídeo codificados do servidor de arquivos pode ser uma transmissão de fluxo contínuo, uma transmissão de download ou uma combinação de ambas.

[0050] As técnicas desta revelação não estão limitadas a aplicativos ou configurações sem fio. As técnicas podem ser aplicadas à codificação de vídeo em suporte a diversos aplicativos multimídia tais como broadcasts de televisão através do ar, transmissões de televisão a cabo, transmissões de televisão por satélite, transmissões de vídeo de fluxo contínuo, como, por exemplo, por meio da Internet, codificação de dados vídeo para armazenamento em um meio de armazenamento de dados, decodificação de dados de vídeo armazenados em um meio de armazenamento de dados ou outros aplicativos. Em alguns exemplos, o sistema de codificação de vídeo 10 pode ser configurado para suportar transmissão de vídeo unidirecional ou bidirecional para suportar aplicativos tais como fluxo contínuo de vídeo, repetição de vídeo, execução de broadcasts de vídeo e/ou telefonia com vídeo.

[0051] No exemplo da Figura 1, o aparelho de origem 12 inclui uma fonte de vídeo 18, um codificador de vídeo 20 e uma interface de saída 22. Em alguns exemplos, a interface de saída 22 pode incluir um modulador/demodulador (modem) e/ou um transmissor. A fonte de vídeo 18 pode incluir um aparelho de captação de vídeo, como, por exemplo, uma câmera de vídeo, um arquivo de vídeo que contém dados de vídeo previamente captados, uma interface de alimentação de vídeo para receber vídeo de um provedor

de conteúdos de vídeo e/ou um sistema de gráfico de computador para gerar dados de vídeo ou uma combinação de tais fontes de dados de vídeo.

[0052] O codificador de vídeo de vídeo 20 pode codificar dados de vídeo da fonte de vídeo 18. Em alguns exemplos, o aparelho de origem 12 transmite diretamente os dados de vídeo codificados para o aparelho de destino 14 por meio da interface de saída 22. Em outros exemplos, os dados de vídeo codificados podem ser também armazenados em um meio de armazenamento ou em um servidor de arquivos para acesso posterior pelo aparelho de destino 14 para decodificação e/ou repetição.

[0053] No exemplo da Figura 1, o aparelho de destino 14 inclui uma interface de entrada 28, um decodificador de vídeo 30 e um aparelho de exibição 32. Em alguns exemplos, a interface de entrada 28 inclui um receptor e/ou um modem. A interface de entrada 28 pode receber dados de vídeo codificados através do canal 16. O aparelho de exibição 32 pode ser integrado com o ou pode ser externo ao aparelho de destino 14. Em geral, o aparelho de exibição 32 exibe dados de vídeo decodificados. O aparelho de exibição 32 pode corresponder diversos aparelhos de exibição, tais como uma tela de cristal líquido (LCD), uma tela de plasma ou uma tela de diodo emissor de luz orgânico (OLED) ou outro tipo de aparelho de exibição. De acordo com esta revelação, o codificador de vídeo 20 e o decodificador de vídeo 30 podem executar um ou mais técnicas aqui descrita como parte de um processo de codificação de vídeo (codificação de vídeo ou decodificação de vídeo, por exemplo).

[0054] A Figura 1 é meramente um exemplo e as técnicas desta revelação podem aplicar-se a configurações de codificação de vídeo (codificação de vídeo ou

decodificação de vídeo, por exemplo) e não incluem necessariamente qualquer comunicação de dados entre o aparelho de codificação de vídeo e o aparelho de decodificação de vídeo. Em outros exemplos, os dados são recuperados de uma memória local, postos em fluxo contínuo através de uma rede ou semelhante. Um aparelho de codificação de vídeo pode codificar e armazenar dados na memória, e/ou um aparelho de decodificação de vídeo pode recuperar e decodificar dados da memória. Em muitos exemplos, a codificação de vídeo e a decodificação de vídeo são efetuadas por aparelhos que não se comunicam entre si, mas simplesmente codificam dados da memória e/ou recuperam e decodificam dados da memória.

[0055] O codificador de vídeo 20 e o decodificador de vídeo 30 podem ser, cada um, implementados como qualquer um de diversos circuitos adequados, tais como um ou mais processadores, processadores de sinais digitais (DSPs), circuitos integrados específicos de aplicativo (ASICs), arranjos de portas programável no campo (FPGAs), lógica discreta, hardware ou quaisquer combinações deles. Se as técnicas forem implementadas parcialmente em software, um aparelho pode armazenar instruções para o software em um meio de armazenamento passível de leitura por computador não transitório adequado e pode executar as instruções no hardware utilizando um ou mais processadores para executar as técnicas desta revelação. Qualquer um dos elementos precedentes (inclusive hardware, software, uma combinação de software e hardware, etc.) pode ser considerado como sendo um ou mais processadores. Cada um do codificador de vídeo 20 e do decodificador de vídeo 30 pode ser incluído em um ou mais codificadores ou decodificadores, ambos podendo ser integrados como parte de

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

[0056] Esta revelação pode referir-se de maneira geral à "sinalização" pelo codificador de vídeo 20, de determinadas informações. O termo "sinalização" pode referir-se de maneira geral à comunicação de elementos de sintaxe e/ou outros dados utilizados para decodificar os dados de vídeo compactados. Tal comunicação pode ocorrer em tempo real ou quase real. Alternativamente, tal comunicação pode ocorrer dentro de uma duração de tempo, tal como pode ocorrer quando se armazena elementos de sintaxe em um meio de armazenamento passível de leitura por computador em um fluxo de bits codificado no momento da codificação, que um aparelho de decodificação de vídeo pode então recuperar em qualquer momento depois de serem armazenados neste meio. Em alguns exemplos, da perspectiva de um codificador, a sinalização pode incluir a geração de um fluxo de bits codificado e, da perspectiva de um decodificador, a sinalização pode incluir o recebimento e a execução de parse em um fluxo de bits codificado.

[0057] Em alguns exemplos, o codificador de vídeo 20 e o decodificador de vídeo 30 funcionam de acordo com um padrão de compactação de vídeo, tal como o ISO/IEC MPEG4 Visual e o ITU-T H.264 (também conhecido como ISO-IEC MPEG-4 AVC) que inclui suas extensões de Codificação Escalonável de Vídeo (SVC) e Codificação de Vídeo de Várias Vistas (MVC). O rascunho conjunto mais recente da MVC é descrito na recomendação ITU-T H.264, "Codificação Avançada de Vídeo para Serviços Áudio Visuais Genéricos" março de 2010. Em outros exemplos, o codificador de vídeo 20 e o decodificador de vídeo 30 podem funcionar de acordo com outros padrões de codificação de vídeo, que incluem o ITU-T H.261, o ISO/IEC MPEG11 Visual, o ITU-T H.262 ou o ISO/IEC

MPEG-2 Visual, o ITU-T H.263, e assim por diante. As técnicas desta revelação, contudo, não estão limitadas a qualquer padrão ou técnica de codificação específica.

[0058] Em outros exemplos, o codificador de vídeo 20 e o decodificador de vídeo 30 podem funcionar de acordo com outros padrões de compactação de vídeo, que incluem o padrão de Codificação de Vídeo de Alta Eficácia (HEVC), desenvolvido pela Equipe de Colaboração Conjunta de Codificação de Vídeo (JCT-VC) do Grupo de Especialistas em Codificação de Vídeo (VCEG), ITU-T, e do Grupo de Especialistas em Imagens em Movimento (MPEG), ISO/IEC. Um rascunho do padrão HEVC referido como "Rascunho Operacional HEVC9", é descrito em Bross et alii, "Rascunho de Especificação de Texto de Codificação de Vídeo de Alta Eficácia (HEVC) 9", Equipe de Colaboração Conjunta de Codificação de Vídeo do ITU-T H. SG16 WP3 e ISO/IEC JCT1/SC29/WG11°. Encontro: Xangai, China, outubro, 2012, pode ser baixado de http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v8.zip. Outro rascunho recente do padrão HEVC, referido como "Rascunho Operacional HEVC 10" ou "WD10" é descrito no documento JCTVC-L1003v34, Bross et alii, "rascunho de especificação de texto de Codificação de Vídeo de Alta Eficácia (HEVC) 10 (para FDIS & Last Call)", "Equipe de Colaboração Conjunta de Codificação de Vídeo (JCT-VC) do ITU-T SG16 WP3 e ISO/IEC JCT1/SC29/WG11, 12°. Encontro Genebra, CH, 14-23 de janeiro de 2013, que pode ser baixado de http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-K1003-v34.zip. Ainda outro rascunho do padrão HEVC é aqui referido como "Revisões WD10", descrito em Bross et alii, "Correções Propostas pelos Editores na Versão HEVC1", Equipe de Colaboração Conjunta de Codificação de Vídeo

(JCT-VC) do ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/G11 13. Encontro, Incheon, KR, abril 2103, que está disponível de http://phenix.int-evry.fr/jct/doc_end_user/documents/13_Incheon/wg11/JCTVC-M0432-v3.zip.

[0059] Atualmente uma Equipe de Colaboração Conjunta de Codificação de Vídeo 3D (JCT-3C) do VCEG e MPEG está desenvolvendo um padrão 3DV com base na HEVC para o qual parte dos esforços de padronização inclui a padronização do CODEC de vídeo de várias vistas baseada na HEVC (MV-HEVC) e outra parte para Codificação de Vídeo 3D baseada na HEVC (3D-HEVC) para a 3D-HEVC, podem ser incluídas e suportadas novas ferramentas de codificação, inclusive ao nível de unidade de codificação/unidade de predição para vistas tanto de textura quanto de profundidade. O software para 3D-HEVC (isto é, 3D-HTM) pode ser baixado do link [3D-HTM versão 8.0]: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSsoftware/tags/HTM-8.0/ um rascunho operacional da 3D-HEVC (isto é, Tech1, "Texto de Rascunho 3D-HEVC", Equipe de Colaboração Conjunta em Desenvolvimento de Extensões de Codificação de Vídeo 3D do ITU-T SG 16 WP3 e ISO/IEC JTC1-SC 29/WG11, 5°. Encontro, Viena, AT, 27 de julho, 2 de agosto de 21013, número de documento JCT3V-E1001-v2 (daqui por diante, "JCTeV-E1001" ou "Texto de Rascunho 3D-HEVC1") está disponível de: http://phenix.it.sud-paris.eu/jct2/doc_end_user/documents/5_Vienna/wg11/JCT3V-E1001-v3.zip. Uma descrição de software da 3D-HEVC (Zangh, et alii, "Modelo de Teste 3D-HEVC 3", Equipe de Colaboração Conjunta em Desenvolvimento de Extensões de Codificação de Vídeo do ITU-T SG16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11 3°. Encontro, Genebra, CH, 12-23 de janeiro de 2013, número de documento: JCT3V-C1005_d0 (daqui por diante "JCT3V-C1005"

ou "Modelo de Teste 3D-HEVC 3") está disponível de http://phenix.int-evry.fr/jct/doc_end_user/documents/3_Geneva/wg11/JCT3V-C1005-v2.zip. Outra descrição de software da 3D-HEVC (Zangh et alii "Modelo de Teste 3D-HEVC 5", Equipe de Colaboração Conjunta em Desenvolvimento de Extensões de Codificação de Vídeo do ITU-T SG 16 WP 3 e ISO/IEC MPEG/IEC JTC 1/SC 29 WG 11, 5º Encontro, Vienna AT, 27 de julho-2 de agosto de 2013, número de documento JCT 3V-E1005 (daqui por diante "JCT3V-E1005") está disponível de http://phenix.it.sud-paris.eu/jct2/doc_end_user/current_documents.php?id=1360.

[0060] Conforme mencionado brevemente acima, o codificador de vídeo 20 codifica dados de vídeo. Os dados de vídeo podem corresponder uma ou mais imagens. Cada uma das imagens é uma imagem parada que faz parte de um vídeo. Quando o codificador de vídeo 20 codifica os dados de vídeo, o codificador de vídeo 20 pode gerar um fluxo de bits. O fluxo de bits pode incluir uma sequência de bits que formam uma representação codificada dos dados de vídeo. O fluxo de bits pode incluir imagens codificadas e dados conexos. Uma imagem codificada é uma representação codificada de uma imagem. Os dados conexos podem incluir conjuntos de parâmetros de sequência (SPSs), conjuntos de parâmetros de imagens (PPSs), conjuntos de parâmetros de vídeo (VPSs) conjuntos de parâmetros adaptativos (APSs), cabeçalhos de fatia, cabeçalhos de blocos e outras estruturas de sintaxe.

[0061] Uma imagem pode incluir três arranjos de amostras, denotados como S_L , S_{Cb} , e S_{Cr} . S_L é um arranjo bidimensional (isto é, um bloco) de amostras luma. As amostras luma podem ser também aqui referidas como amostras "Y". S_{Cb} é um arranjo bidimensional de amostras de crominância Cb. S_{Cr} é arranjo bidimensional de amostras de

crominância Cr. As amostras de crominância podem ser também aqui referidas como amostras "croma". As amostras de crominância Cb podem ser aqui referidas como "amostras U". As amostras de crominância Cr podem ser aqui referidas como "amostras V".

[0062] Em alguns exemplos, o codificador de vídeo 20 pode sub-amostrar os arranjos croma de uma imagem (isto é, S_{Cb} e S_{Cr}). Por exemplo, o codificador de vídeo 20 pode utilizar um formato de vídeo YUV 4:2:0, um formato de vídeo YUV 4:2:2 ou um formato de vídeo 4:4:4. No formato de vídeo YUV 4:2:0, o codificador de vídeo 20 pode sub-amostrar os arranjos croma de modo que os arranjos croma sejam metade da altura e metade da largura do arranjo croma. No formato de vídeo YUV 4:2:2, o codificador de vídeo 20 pode sub-amostrar os arranjos croma de modo que os arranjos croma sejam metade da largura e da mesma altura do arranjo luma.. No formato de vídeo YUV 4:4:4, o codificador de vídeo 20 não sub-amostra os arranjos croma.

[0063] Para gerar uma representação codificada de uma imagem, o codificador de vídeo 20 pode gerar um conjunto de unidades de árvore de codificação (CTUs). Cada uma das CTUs pode ser um bloco de árvore de codificação de amostras luma, dois blocos de árvore de codificação correspondentes de amostras croma e estruturas de sintaxe utilizadas para codificar as amostras do bloco de árvore de codificação. Em uma imagem monocromática ou em uma imagem que tem três planos coloridos separados, uma CTU pode compreender um único bloco de árvore de codificação e estruturas de sintaxe utilizadas para codificar as amostras do bloco de árvore de codificação. Um bloco de árvore de codificação (CTB), pode ser um bloco NxN de amostras. Uma CTU pode ser também referida como "bloco de árvore" ou "maior unidade de codificação" (LCU). As CTUs da HEVC podem

ser amplamente análogas aos macroblocos de outros padrões, tais como o H.264/AVC. Entretanto, uma CTU não está necessariamente limitada a um tamanho específico e pode incluir uma ou mais unidades de codificação (CUs).

[0064] Como parte da codificação de uma imagem, o codificador de vídeo 20 pode gerar representações codificadas de cada fatia da imagem (isto é, fatias codificadas). Para gerar uma fatia codificada, o codificador de vídeo 20 pode codificar uma série de CTUs. Esta revelação pode referir-se a uma representação codificada de uma CTU como CTU codificada. Em alguns exemplos, cada uma das fatias inclui um número inteiro de CTUs codificadas.

[0065] Para gerar uma CTU codificada, o codificador de vídeo 20 pode efetuar recursivamente particionamento de transformação quad-tree nos blocos de árvore de codificação de uma CTU de modo a dividir os blocos de árvore de codificação em blocos de codificação, daí o nome "unidades de árvore de codificação". Um bloco de codificação é um bloco NxN de amostras. Uma CU pode ser um bloco de codificação de amostras luma e dois blocos de codificação correspondente de amostras croma de uma imagem que tem um arranjo de amostras luma, um arranjo de amostras Cb e um arranjo de amostras Cr e estruturas de sintaxe utilizadas para codificar as amostras dos blocos de codificação. Em uma imagem monocromática ou uma imagem que tem três planos coloridos separados, uma CU pode compreender um único bloco de codificação e estruturas de sintaxe utilizadas para codificar as amostras do bloco de codificação.

[0066] O codificador de vídeo 20 pode particionar um bloco de codificação de uma CU em um ou mais blocos de predição. Um bloco de predição pode ser um bloco

retangular (isto é, quadrado ou não quadrado) de amostras no qual a mesma predição é aplicada. Uma unidade de predição (PU) de uma CU pode ser um bloco de predição de amostras luma, dois blocos de predição correspondentes de amostras croma de uma imagem e estruturas de sintaxe utilizadas para predizer as amostras do bloco de predição. Uma imagem monocromática ou uma imagem que tem três planos coloridos separados, uma PU pode compreender um único bloco de predição e estruturas de sintaxe utilizadas para predizer as amostras do bloco de predição. O codificador de vídeo 20 pode gerar um bloco preditivo para cada bloco de predição de uma PU. Por exemplo, o codificador de vídeo 20 pode gerar luma preditiva, blocos Cb e Cr para a luma, blocos de predição Cb e Cr de cada PU da CU. Os blocos preditivos podem ser também referidos como blocos de amostras preditivos.

[0067] O codificador de vídeo 20 pode utilizar intra-predição ou inter-predição para gerar os blocos preditivos para uma PU. Se o codificador de vídeo 20 utilizar intra-predição para gerar os blocos preditivos de uma PU, o codificador de vídeo 20 pode gerar os blocos preditivos da PU com base em amostras decodificadas da imagem associada à PU.

[0068] Se o codificador de vídeo 20 utilizar inter-predição para gerar os blocos preditivos de uma PU, o codificador de vídeo 20 pode gerar os blocos preditivos da PU com base em amostras decodificadas de uma ou mais imagens outras que não a imagem associada à PU. O codificador de vídeo 20 pode utilizar uni-predição ou bi-predição para gerar os blocos preditivos de uma PU. Quando o codificador de vídeo 20 utiliza uni-predição para gerar os blocos preditivos para uma PU, a PU pode ter um único vetor de movimento. Quando o codificador de vídeo 20

utiliza uni-predição para gerar os blocos preditivos para uma PU, a PU pode ter dois vetores de movimento.

[0069] Depois que o codificador de vídeo 20 gera blocos de predição (luma preditiva, blocos Cb e Cr, por exemplo) para uma ou mais PUs de uma CU, o codificador de vídeo 20 pode gerar um ou mais blocos residuais para a CU. Cada amostra no bloco residual para a CU pode indicar a diferença entre uma amostra em um bloco preditivo de uma PU da CU e uma amostra correspondente no bloco de codificação da CU. Por exemplo, o codificador de vídeo 20 pode gerar um bloco residual luma para a CU. Cada amostra em um bloco residual luma da CU pode indicar a diferença entre uma amostra luma em um blocos luma preditivos da PU da CU e uma amostra correspondente no bloco de codificação Cb original da CU. Além disto, o codificador de vídeo 20 pode gerar um bloco residual Cb para a CU. Cada amostra em um bloco residual Cb de uma CU pode indicar a diferença entre uma amostra Cb em um de um bloco Cb preditivo de uma PU da CU e uma amostra correspondente em um bloco de codificação Cb original da CU. O codificador de vídeo 20 pode gerar também um bloco residual Cr para a CU. Cada amostra em um bloco residual Cr da CU pode indicar a diferença entre uma amostra Cr em um bloco Cr preditivo de uma PU da CU e uma amostra correspondente em um bloco de codificação Cr original da CU.

[0070] Além disso, o codificador de vídeo 20 pode utilizar particionamento de transformação quad-tree para decompor os blocos residuais (luma, blocos residuais Cb e Cr de uma CU, por exemplo) em uma mais luma, blocos de transformada Cb e Cr, por exemplo. Um bloco de transformada pode ser um bloco retangular de amostras no qual a mesma transformada é aplicada. Uma unidade de transformada (TU) de uma CU pode ser um bloco de transformada de amostras

luma, dos blocos de transformada correspondentes de amostras croma e estruturas de sintaxe utilizadas para transformar as amostras dos blocos de transformada. Assim, cada TU de uma CU pode ser associada a um bloco de transformada luma, um bloco de transformada Cb e um bloco de transformada Cr. Em uma imagem monocromática ou uma imagem que tem três planos coloridos separados, uma TU pode compreender um único bloco de transformada e estruturas de sintaxe utilizadas para transformar as amostras de o bloco de transformada. Assim, cada TU de uma CU pode estar associada a um bloco de transformada luma, um bloco de transformada Cb e um bloco de transformada Cr. Em uma imagem monocromática ou uma imagem que tem três planos coloridos separados, uma TU pode compreender um único bloco de transformada e estruturas de sintaxe utilizadas para transformar as amostras do bloco de transformada. O bloco de transformada luma de (isto é, associado a) uma TU de uma CU pode ser um sub-bloco de um bloco residual luma da CU. O bloco de transformada Cb de uma TU de uma CU pode ser um sub-bloco de um bloco residual Cb da CU. O bloco de transformada Cr de uma TU de uma CU pode ser um sub-bloco de um bloco residual Cr da CU.

[0071] Para codificação 3D, os valores de profundidade em blocos de profundidade podem ser igualmente representados como valores de amostra (valores luma, por exemplo) cada um deles indicando o nível de profundidade associado a uma determinada localização de pixel. Uma ou mais das técnicas desta revelação são aplicáveis à codificação de blocos de profundidade, particularmente em modos tais como o modo por saltos ou o modo de fusão, onde uma lista de candidatos é gerada para herdar ou utilizar as informações de movimento de um candidato selecionado na codificação do bloco de profundidade.

[0072] O codificador de vídeo 20 pode aplicar uma ou mais transformadas a um bloco de transformada uma TU de modo a gerar um bloco de coeficientes para a TU. Um bloco de coeficientes pode ser um arranjo bidimensional de coeficientes de transformada. Um coeficiente de transformada pode ser uma quantidade escalar. Por exemplo, o codificador de vídeo 20 pode aplicar uma ou mais transformadas a um bloco de transformada luma de uma TU de modo a gerar um bloco de coeficientes luma para a TU. O codificador de vídeo 20 pode aplicar uma ou mais transformadas a um bloco de transformada Cb de uma TU de modo a gerar um bloco de coeficientes Cb para a TU. O codificador de vídeo 20 pode aplicar uma ou mais transformadas a um bloco de transformada Cr de uma TU de modo a gerar um bloco de coeficientes Cr para a TU.

[0073] Depois de gerar um bloco de coeficientes (um bloco de coeficientes luma, um bloco de coeficientes Cb ou um bloco de coeficientes Cr, por exemplo), o codificador de vídeo 20 pode quantificar o bloco de coeficientes. A quantificação refere-se de maneira geral a um processo no qual coeficientes de transformada são quantificados para reduzir possivelmente a quantidade de dados utilizados para representar os coeficientes de transformada, obtendo-se compactação adicional. Depois que o codificador de vídeo 20 quantifica um bloco de coeficientes, o codificador de vídeo 20 pode codificar por entropia elementos de sintaxe que indicam os coeficientes de transformada quantificados. Por exemplo, o codificador de vídeo 20 pode efetuar Codificação Aritmética Binária Adaptativa ao Contexto (CABAC) nos elementos de sintaxe que indicam os coeficientes de transformada quantificados. O codificador de vídeo 20 pode transmitir os elementos de sintaxe codificados por entropia em um fluxo de bits. O

fluxo de bits pode incluir também elementos de sintaxe que não são codificados por entropia.

[0074] O decodificador de vídeo 30 pode receber um fluxo de bits gerado pelo codificador de vídeo 20. Além disto, o decodificador de vídeo 30 pode efetuar parse no fluxo de bits de modo a obter (decodificar, por exemplo) elementos de sintaxe do fluxo de bits. O decodificador de vídeo 30 pode reconstruir as imagens dos dados de vídeo com base, pelo menos em parte, nos elementos de sintaxe decodificados, ou senão obtidos do fluxo de bits. O processo para reconstruir os dados de vídeo pode ser geralmente complementar ao processo executado pelo codificador de vídeo 20. Por exemplo, o decodificador de vídeo 30 pode utilizar vetores de movimento de PUs para determinar blocos preditivo. Além disto, o decodificador de vídeo 30 pode quantificar por inversão blocos de coeficientes de transformada associados a TUs da CU atual. O decodificador de vídeo 30 pode efetuar transformada inversa nos blocos de coeficientes de transformada de modo a reconstruir blocos de transformadas associados às TU da CU atual. Em alguns exemplos, o decodificador de vídeo 30 pode reconstruir os blocos de codificação da CU atual adicionando as amostras dos blocos preditivos para as PUs da CU atual a amostras correspondentes dos blocos de transformadas das TUs da CU atual. Pela reconstrução dos blocos de codificação para cada CU de uma imagem, o decodificador de vídeo 30 pode reconstruir a imagem.

[0075] Em alguns casos, o codificador de vídeo 20 pode sinalizar as informações de movimento de uma PU utilizando o modo de fusão ou o modo por saltos, ou possivelmente um modo de predição avançada de vetores de movimento (AMVP). Em outras palavras, no padrão HEVC há dois modos de inter-predição para a PU, a saber o modo de

fusão (o salto é considerado como um caso especial de fusão e o modo AMVP, respectivamente) ou no modo AMVP, um codificador de vídeo mantém uma lista de candidatos a vetor de movimento para vários preditores de vetor de movimento. Para facilitar a explicação, esta revelação pode referir-se a uma lista de candidatos a vetor de movimento para o modo de fusão como "lista de candidatos à fusão". Da mesma maneira, esta revelação pode referir-se a uma lista de candidatos a vetor de movimento para o modo AMVP como lista de candidatos AMVP. As informações de movimento de uma PU podem incluir o vetor ou vetores de movimento da PU e/ou índice ou índices de referência da PU.

[0076] Quando o codificador de vídeo 20 sinaliza as informações de movimento da PU atual utilizando o modo de fusão, o codificador de vídeo 20 gera uma lista de candidatos à fusão. A lista de candidatos à fusão inclui um conjunto de candidatos. Os candidatos em uma lista de candidatos à fusão podem ser referidos como "candidatos à fusão". Os candidatos podem indicar as informações de movimento de PUs que são espacialmente ou temporalmente vizinhas da PU atual. As PUs que são espacialmente vizinhas da PU atual podem ter blocos preditivos adjacentes a um bloco preditivo da PU atual na mesma imagem da PU atual. As PUs que são temporalmente vizinhas à PU atual podem estar em uma imagem diferente da PU atual. O codificador de vídeo 20 pode selecionar então um candidato da lista de candidatos e pode utilizar as informações de movimento indicadas pelo candidato selecionado como as informações de movimento da PU atual. Além disto, no modo de fusão, o codificador de vídeo 20 pode sinalizar a posição na lista de candidatos do candidato selecionado. Por exemplo, o codificador de vídeo 20 pode sinalizar um índice de fusão (merge_idx, por exemplo) que indica a posição na lista de

candidatos à fusão do candidato à fusão selecionado. O decodificador de vídeo 30 pode gerar a mesma lista de candidatos e pode determinar, com base na indicação da posição do candidato selecionado (a posição indicada pelo índice de fusão, por exemplo), o candidato selecionado. O decodificador de vídeo 30 pode utilizar então as informações de movimento do candidato selecionado para gerar um ou mais blocos preditivos (amostras preditivas, por exemplo) para a PU atual. O decodificador de vídeo 30 pode reconstruir amostras com base nos blocos preditivos (amostras preditivas, por exemplo) para a PU atual e um sinal residual. Desta maneira, um codificador de vídeo pode gerar um vetor ou vetores de movimento, assim como índices de referência no modo de fusão, da PU atual tirando um candidato da lista de candidatos a vetor de movimento.

[0077] O modo por saltos é semelhante ao modo de fusão no sentido de que o codificador de vídeo 20 gera uma lista de candidatos e seleciona um candidato da lista de candidatos. Entretanto, quando o codificador de vídeo 20 sinaliza as informações de movimento da PU atual (um bloco de profundidade, por exemplo) utilizando o modo por saltos, o codificador de vídeo 20 pode evitar a geração de qualquer sinal residual. Uma vez que o modo por saltos tem o mesmo processo de derivação de vetor de movimento do modo de fusão, as técnicas descritas neste documento podem aplicar-se tanto ao modo de fusão quanto ao modo por saltos. Um ou mais aspectos desta revelação podem ser utilizados no modo AMVP ou em outros modos que utilizam listas de candidatos.

[0078] O modo AMVP é semelhante ao modo de fusão no sentido de que o codificador de vídeo 20 gera uma lista de candidatos e seleciona um candidato da lista de candidatos. Entretanto, quando o codificador de vídeo

sinaliza as informações de movimento da PU atual (um bloco de profundidade, por exemplo) utilizando o modo AMVP, o codificador de vídeo 20 pode sinalizar uma diferença entre vetores de movimento (MVD) para a PU atual e um índice de referência além da sinalização da posição do candidato selecionado na lista de candidatos. Um MVD para a PU atual pode indicar a diferença entre um vetor de movimento da PU e o vetor de movimento do candidato a vetor de movimento selecionado. Na uni-predição o codificador de vídeo 20 pode sinalizar uma MVD e um índice de referência para a PU atual. Na bi-predição, o codificador de vídeo 20 pode sinalizar dois MVDs e dois índices de referência para a PU atual. Em alguns exemplos, o codificador de vídeo 20 pode sinalizar tipicamente uma MVD e um índice de referência para a PU atual, embora a predição de blocos de profundidade possa utilizar também técnicas semelhantes à bi-predição, nas quais são sinalizados dois MVDs e dois índices de referência.

[0079] Além disto, quando as informações de movimento da PU atual são sinalizadas utilizando-se o modo AMVP, o decodificador de vídeo 30 pode gerar a mesma lista de candidatos e pode determinar, com base na indicação da posição do candidato selecionado, o candidato selecionado. O decodificador de vídeo 30 pode recuperar o vetor de movimento da PU atual adicionando uma MVD ao vetor de movimento do candidato selecionado. O decodificador de vídeo 30 pode utilizar então o vetor de movimento ou vetores de movimento recuperados da PU atual para gerar blocos preditivos para a PU atual.

[0080] Em alguns exemplos, a lista de candidatos a vetor de movimento contém até cinco candidatos para o modo de fusão e apenas dois candidatos para o modo AMVP. Em outras palavras, uma lista de candidatos à fusão

pode incluir até cinco candidatos, enquanto uma lista de candidatos AMVP só pode incluir dois candidatos. Um candidato à fusão (isto é, um candidato em uma lista de candidatos a vetor de movimento para o modo de fusão) pode conter vetores de movimento que correspondem tanto a listas de índices de referência (lista 0 e lista 1) quanto aos índices de referência. Se um candidato à fusão for identificado por um índice de fusão, as imagens de referência utilizadas para a predição dos blocos atuais, assim como os vetores de movimento conexos são determinados. Entretanto, sob um modo AMVP para cada direção de predição potencial ou da lista 0 ou da lista 1 é explicitamente sinalizado um índice de referência, juntamente com um índice de preditor de vetor de movimento da lista de candidatos a vetor de movimento uma vez que o candidato a AMVP contém apenas um vetor de movimento. No modo AMVP, os vetores de movimento preditos podem ser também refinados.

[0081] Conforme indicado acima, um codificador de vídeo pode derivar candidatos para o modo de fusão de blocos vizinhos espaciais e temporais. O codificador de vídeo pode derivar o número máximo de candidatos do elemento de sintaxe codificado `five_minus_max_num_merge_cand`, que é incorporado em um cabeçalho de fatia para uma fatia. O elemento de sintaxe `five_minus_max_num_merge_cand` especifica o número máximo de candidato à fusão suportados na fatia, subtraído de 5. O codificador de vídeo pode derivar o número máximo de candidato à fusão `MaxNumMergeCand`, da maneira seguinte:

$$\text{MaxNumMergeCand} = 5 -$$

$$\text{five_minus_max_num_merge_cand} \quad (7-39)$$

O valor de `MaxNumMergeCand` está na faixa de 1 a 5, inclusive.

[0082] Um codificador de vídeo pode construir a lista de candidatos à fusão com as etapas seguintes. Em primeiro lugar, o codificador de vídeo pode derivar até 4 candidatos a vetor de movimento espaciais de 5 blocos vizinhos espaciais mostrados na Figura 1. A Figura 2 é uma ilustração conceitual que mostra vizinhos espaciais que são os candidatos potenciais para a lista de fusão. As setas indicam qual ou quais candidatos espaciais serão comparados. O codificador de vídeo pode derivar os candidatos a vetor de movimento espaciais na ordem seguinte: esquerda (A1), acima (B1), acima à direita (B0), abaixo à esquerda (A0) e acima à esquerda (B2), conforme mostrado na Figura 2. Além disto, o codificador de vídeo pode aplicar um processo de podar para remover candidatos a vetor de movimento espaciais idênticos. Por exemplo, o codificador de vídeo pode comparar B1 com A1, comparar B0 com B1, comparar A0 com A1 e comparar B2 tanto com B1 quanto A1, se já houver quatro candidatos à fusão disponíveis depois do processo de podar, o codificador de vídeo não insere B2 na lista de candidatos à fusão.

[0083] Em segundo lugar, o codificador de vídeo pode determinar candidatos à fusão temporais. Por exemplo, o codificador de vídeo pode adicionar um candidato a preditor de vetor de movimento temporal (TMVP) de uma imagem de referência co-localizada (se habilitada e disponível) à lista de candidatos à fusão (isto é, a lista de candidatos a vetor de movimento) depois de candidatos a vetor (isto é, a lista de candidatos a vetor de movimento) não estiver completa, candidatos a vetor de movimento espaciais.

[0084] Em terceiro lugar, se a lista de candidatos à fusão, o codificador de vídeo pode gerar e inserir candidatos a vetor de movimento artificiais no

final da lista de candidatos à fusão até que a lista de candidatos à fusão tenha todos os candidatos (isto é, todos os candidatos indicados por MaxNumMergeCand). Em outras palavras, o codificador de vídeo pode inserir candidatos a vetor de movimento artificiais na lista de candidatos à fusão se o número de candidatos à fusão na lista de candidatos à fusão for menor que MaxNumMergeCand. Há dois tipos de candidatos a vetor de movimento artificiais: candidatos à fusão bi-preditiva combinada (que são derivados apenas para fatias B) e zero candidatos à fusão de vetor de movimento. A lista de candidatos à fusão pode incluir um ou mais zero candidatos à fusão de vetor de movimento se o primeiro tipo (isto é, candidatos à fusão bi-preditiva combinada) não apresentar candidatos artificiais suficientes.

[0085] Quando a fatia atual (isto é, a fatia que um codificador de vídeo está codificando atualmente) é uma fatia B, o codificador de vídeo pode chamar um processo de derivação para candidatos à fusão bi-preditiva combinada. Em pelo menos alguns exemplos, uma fatia B é uma fatia na qual são permitidas a intra-predição, a inter-predição unidirecional e a inter-predição bidirecional. Quando o processo de derivação é chamado, o codificador de vídeo pode, para cada par de candidatos à fusão que já estão na lista de candidatos à fusão e têm as informações de movimento necessárias, derivar candidatos a vetor de movimento bi-preditivo combinado (com índice denotado por comblIdx) por uma combinação do vetor de movimento (e, em alguns casos, do índice de referência) do primeiro candidato à fusão do par (com índice de candidato à fusão igual a 10CandIdx) referente a uma imagem na lista 0 (se disponível) e do vetor de movimento (e, em alguns casos, do índice de referência) de um segundo candidato à fusão do

par (com índice de candidato à fusão igual a 11CandIdx) referente a uma imagem na lista 1 (se disponível e ou a imagem de referência ou o vetor de movimento for diferente do primeiro candidato). O par de candidatos à fusão pode ser um par ordenado no sentido de que ordens diferentes dos mesmos dois candidatos à fusão são consideradas pares diferentes. A definições de 10CandIdx e 11CandIdx que compreendem a combIdx são mostradas na Tabela 1 abaixo:

Tabela 1: Especificação de 10CandIdx e 11CandIdx

| combIdx | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|-----------|---|---|---|---|---|---|---|---|---|---|----|----|
| 10CandIdx | 0 | 1 | 0 | 2 | 1 | 2 | 0 | 3 | 1 | 3 | 2 | 3 |
| 11CandIdx | 1 | 0 | 2 | 0 | 2 | 1 | 3 | 0 | 3 | 1 | 3 | 2 |

[0086] Na Tabela 1, a fileira para 10CandIdx indica índices de candidatos à fusão dos quais retirar informações de movimento RefPicList0 (vetores de movimento, índices de referência, por exemplo). Da mesma maneira, na Tabela 1, a fileira para 11CandIdx indica índices de candidatos à fusão dos quais retirar informações de movimento RefPicList1. Assim, a coluna para a combinação 0 (isto é, combIdx = 0) indica que um candidato a vetor de movimento e bi-preditivo combinado especifica as informações de movimento RefPicList0 do candidato à fusão 0 e especifica as informações de movimento RefPicList1 do candidato à fusão 1. Uma vez que nem todos os candidatos à fusão têm necessariamente as informações de movimento aplicáveis para uma combinação (o candidato à fusão 1 pode não ter as informações de movimento RefPicList1, por exemplo) ou as informações de movimento de RefPicList0 associadas ao candidato à fusão 0 e RefPicList1 associadas ao candidato à fusão são idênticas, um codificador de vídeo pode processar as combinações da Tabela 1 na ordem de combIdx até que não haja combinações restantes disponíveis

ou o codificador de vídeo tenha gerado um número suficiente de candidatos a vetor de movimento bi-preditivo combinado.

[0087] Para `combIdx` sendo 0 ... 11, o processo de geração de candidatos a vetor de movimento bi-preditivo combinado é terminado quando uma das condições seguintes for verdadeira:

- `combIdx` é igual a $(\text{numOrigMergeCand} * (\text{numOrigMergeCand} - 1))$, onde `numOrigMergeCand` denota o número de candidatos na lista de fusão antes da chamada deste processo.
- O número de candidatos totais, inclusive candidatos à fusão bi-preditiva combinada é recém gerados) na lista de fusão é igual a `MaxNumMergeCand`.

[0088] Conforme indicado acima, um codificador de vídeo pode incluir um ou mais zero candidatos à fusão de vetor de movimento em uma candidatos à fusão. Para cada respectivo zero candidato à fusão de vetor de movimento, o vetor de movimento do respectivo zero candidato à fusão de vetor de movimento é fixado em 0 e o índice de referência para o respectivo zero candidato à fusão de vetor de movimento é fixado de zero até o número de índices de referência disponível menos 1. Se o número de candidatos à fusão na lista de candidatos à fusão for ainda menor que `MaxNumMergeCand`, o codificador de vídeo pode inserir um ou mais zero candidatos a vetor de movimento (zero índices de referência e vetores de movimento, por exemplo) até que o número total de candidatos à fusão na lista de candidatos à fusão seja igual a `MaxNumMergeCand`.

[0089] As sub-seções seguintes desta revelação examinam as técnicas de codificação de vídeo 3D baseadas em AVC e baseadas em HEVC relacionadas com esta revelação. Na codificação de várias vistas (codificação de vídeo 3D) pode haver várias vistas na mesma cena de pontos de vistas

diferentes). O termo "unidade de acesso" pode ser utilizado para referir-se ao conjunto de imagens que correspondem à mesma ocorrência de tempo. Em outras palavras, uma unidade de acesso pode incluir imagens codificadas de todas as vistas para uma ocorrência de tempo de saída. Um "componente de vista", pode ser uma representação codificada de uma vista em uma única unidade de acesso. Em alguns exemplos, um componente de vista pode conter um componente de vista de textura e um componente de vista de profundidade. Nesta revelação, uma "vista" pode referir-se a uma sequência de componentes de vista associados ao mesmo identificador de vista. Assim, quando uma vista inclui representações tanto de textura quanto de profundidade codificada, um componente de vista pode compreender (consistir em, por exemplo) um componente de vista de textura e um componente de vista de profundidade. Em alguns exemplos, um componente de vista de textura é uma representação codificada da textura de uma vista em uma única unidade de acesso. Além disto, em alguns exemplos, um componente de vista de profundidade é uma representação codificada da profundidade de uma vista em uma única unidade de acesso. Um componente de vista de profundidade pode ser também referido como imagem de profundidade.

[0090] Cada componente de vista de textura inclui um conteúdo de imagem real a ser exibido. Por exemplo, um componente de vista de textura pode incluir componentes luma (Y) e croma (Cd e Cr). Cada componente de vista de profundidade pode indicar as profundidades relativas dos pixels em seu componente de vista de textura correspondente. Em alguns exemplos, os componente de vista de profundidade são imagens em escala cinza que inclui apenas valores luma. Em outras palavras, os componentes de vista de profundidade podem não transmitir qualquer

conteúdo de imagem, mas, em vez disso, podem fornecer medidas das profundidades relativas dos pixels em componentes de componentes de vista de textura correspondente.

[0091] Por exemplo, um pixel puramente branco em um componente de vista de profundidade pode indicar que o pixel ou pixels correspondentes do pixel no componente de vista de textura correspondente estão mais próximos, da perspectiva do observador. Neste exemplo, um pixel puramente preto no componente de vista de profundidade indica que o pixel ou pixels correspondentes do pixel no componente de vista de textura correspondente estão mais afastados, da perspectiva do observador. As diversas sombras de cinza entre preto e branco indicam níveis de profundidade diferentes. Por exemplo, um pixel cinza escuro em um componente de vista de profundidade indica que o pixel correspondente do pixel no componente de vista de textura está mais afastado que um pixel cinza claro no componente de vista de profundidade. Neste exemplo, uma vez que apenas a escala cinza é necessária para identificar a profundidade dos pixels, os componentes de vista de profundidade não precisam incluir componentes croma. Uma vez que os componentes croma para os componentes de vista de profundidade podem não servir a nenhuma finalidade. Esta revelação apresenta o exemplo de componentes de vista de profundidade que utilizam apenas valores luma (valores de intensidade, por exemplo) para identificar a profundidade para fins de exemplificação, e não deve ser considerada limitadora. Em outros exemplos, outras técnicas podem ser utilizadas para indicar as profundidades relativas dos pixels em componentes de vista de textura.

[0092] Na codificação de várias vistas, um fluxo de bits pode ter uma série de camadas. Cada uma das

camadas pode compreender a uma vista diferente. Na codificação de várias vistas, uma vista pode ser referida como "vista base" se um decodificador de vídeo (o decodificador de vídeo 30, por exemplo) puder decodificar imagens na vista sem referência a imagens a qualquer outra vista. Uma vista pode ser referida como vista não base se a decodificação da vista depender da decodificação de imagens em uma ou mais outras vistas. Quando codifica uma imagem em uma das vistas não base, um codificador de vídeo (tal como o codificador de vídeo 20 ou o decodificador de vídeo 30, por exemplo) pode adicionar uma imagem a uma lista de imagens de referência se a imagem estiver em uma vista diferente, mas dentro da mesma ocorrência de tempo (isto é, unidade de acesso da imagem que o codificador de vídeo está codificando atualmente). Como outras imagens de referência de inter-predição, o codificador de vídeo pode inserir uma imagem de referência de predição inter-vista em qualquer posição de uma lista de imagens de referência.

[0093] Na 3D-HEVC, um vetor de disparidade (DV) pode ser utilizado como estimador do deslocamento entre duas vistas. Uma vez que blocos vizinhos compartilham quase as mesmas informações de movimento/disparidade na codificação de vídeo, o bloco atual pode utilizar as informações de vetor de movimento em blocos vizinhos como um bom preditor. Seguindo esta idéia, o processo de derivação de vetor de disparidade baseado em blocos vizinhos (NBDV) utiliza as informações de vetor de movimento vizinhas para estimar o vetor de disparidade em vistas diferentes. A 3D-HEVC adotou primeiramente o método de Vetor de Disparidade (baseado em) Blocos Vizinhos (NBDV) proposto no documento seguinte: Zangh et alii, "3D-CE5.h: Resultados de geração de vetores de disparidade" Equipe de Colaboração Conjunta em Desenvolvimento de Extensões de

Codificação de Vídeo 3D do ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 1°. Encontro: Estocolmo, SE, 16-20 de julho de 2012, documento JCT3V-A0097 (daqui por diante "JCT3V-A0097").

[0094] Vários blocos vizinhos espaciais e temporais são definidos no processo NBDV. Um codificador de vídeo que executa o processo NBDV verifica cada um dos blocos vizinhos espaciais e temporais em uma ordem pré-definida determinada pela prioridade da correlação entre o bloco atual e o bloco candidato (bloco vizinho especial ou temporal). Assim, no processo NBDV, o codificador de vídeo utiliza dois conjuntos de blocos vizinhos. Um conjunto de blocos vizinhos é de blocos vizinhos espaciais e o outro conjunto é de blocos vizinhos temporais. Quando o codificador de vídeo verifica um bloco vizinho, o codificador de vídeo pode determinar se o bloco vizinho tem um vetor de movimento de disparidade (isto é, o indica uma imagem de referência inter-vista). Uma vez que o codificador de vídeo encontra um vetor de movimento de disparidade, o codificador de vídeo pode converter o vetor de movimento de disparidade em um vetor de disparidade. Por exemplo, para converter o vetor de movimento de disparidade no vetor de disparidade, o codificador de vídeo pode fixar o vetor de disparidade como igual ao vetor de movimento de disparidade. Entrementes, o índice de ordem de vistas de referência conexo é também enviado de volta. Em outras palavras, como parte da execução do processo NBDV, o codificador de vídeo pode determinar também um índice de ordem de vistas de referência.

[0095] Em algumas versões da 3D-HEVC, o codificador de vídeo utiliza dois blocos vizinhos espaciais no processo NBDV para a derivação de vetores de disparidade. Os dois blocos vizinhos espaciais estão à

esquerda e acima da CU atual, conforme denotado por A1, B1, conforme mostrado na Figura 3. A Figura 3 é um diagrama conceitual que mostra blocos vizinhos espaciais e temporais relativos à unidade de codificação atual. Deve-se observar que os blocos vizinhos espaciais utilizados no processo NBDV são idênticos aos utilizados no modo de fusão na HEVC. Portanto, pelo menos em alguns exemplos, nenhum acesso adicional à memória é necessário quando se processam blocos vizinhos espaciais no processo NBDV.

[0096] Em alguns exemplos, para verificar blocos vizinhos temporais no processo NBDV, o codificador de vídeo pode executar primeiro um processo de construção para gerar uma lista de imagens candidatas. Até duas imagens de referência da vista atual (isto é, a vista que inclui a imagem que está sendo atualmente codificada) pode ser tratada como imagens candidatas. Uma imagem de referência co-localizada (isto é, uma imagem co-localizada) é primeiro inserida na lista de imagens candidatas, seguida pelo resto das imagens candidatas (isto é, todas as imagens de referência na RefPicList0 e na RefPicList1) na ordem ascendente de índices de referência.

[0097] Se a fatia atual da imagem atual for uma fatia B (isto é, uma fatia que é autorizada a incluir PUs inter-preditas bidirecionalmente) o codificador de vídeo 20 pode sinalizar, em um cabeçalho de fatia, um elemento de sintaxe (`collocated_from_10_flag`, por exemplo) que indica se a imagem co-localizada é da RefPicList0 ou da RefPicList1. Em outras palavras, quando a utilização de TMVPs é habilitada para a fatia atual e a fatia atual é uma fatia B (uma fatia que é autorizada a incluir PUs inter-preditas bidirecionalmente) o codificador de vídeo pode sinalizar um elemento de sintaxe (`collocated_from_10_flag`, por exemplo) em um cabeçalho de fatia para indicar se a

imagem co-localizada em RefPicList0 ou em RefPicList1. Se a fatia atual não for uma fatia B, pode ser desnecessário que o codificador de vídeo 20 sinalize o elemento de sintaxe para indicar se a imagem co-localizada está em RefPicList0 ou em RefPicList1 uma vez que, se fatia atual for uma fatia I, nenhuma inter-predição é permitida e, se a fatia atual for uma fatia P só há uma lista de imagens de referência para a fatia. Depois que o decodificador de vídeo 30 identifica a lista de imagens de referência que inclui a imagem co-localizada, o decodificador de vídeo 30 pode utilizar outro elemento de sintaxe (`collocated_ref_idx`, por exemplo) que pode ser sinalizado em um cabeçalho de fatia para identificar uma imagem (a imagem co-localizada) na lista de imagens de referência identificada. Ou seja, depois que uma lista de imagens de referência `collocated_ref_idx`, que é sinalizada em um cabeçalho de fatia, pode ser utilizado para identificar a imagem na lista de imagens de referência.

[0098] Quando duas imagens de referência com o mesmo índice de referência em ambas as listas de imagens de referência estão disponíveis, a imagem de referência na mesma lista de imagens de referência da imagem co-localizada precede a outra imagem de referência. Para cada imagem candidata na lista de imagens de referência, o codificador de vídeo pode determinar o bloco da região co-localizada que cobre a posição central como bloco vizinho temporal.

[0099] Quando um bloco é codificado com predição de movimento inter-vista o codificador de vídeo pode precisar derivar um vetor de disparidade para selecionar um bloco correspondente em uma vista diferente. Um vetor de disparidade implícito (IDV ou então vetor de disparidade derivado) pode ser referido como vetor de

disparidade derivado na predição de movimento inter-vista. Embora o bloco seja codificado com predição de movimento, o vetor de disparidade derivado não é descartado para fins de codificação do bloco seguinte.

[0100] Em pelo menos alguns desenhos do 3D-HEVC, o processo NBDV verifica vetores de movimento de disparidade nos blocos vizinhos temporais, vetores de movimento de disparidade nos blocos vizinhos espaciais e então os IDVs em ordem. Uma vez que o codificador de vídeo encontra um vetor de movimento de disparidade ou IDV, o codificador de vídeo termina o processo NBDV.

[0101] Em alguns exemplos, quando um codificador de vídeo deriva um vetor de disparidade do processo NBDV, o codificador de vídeo também refina o vetor de disparidade recuperando dados de profundidade de um mapa de profundidade (isto é, um componente de vista de profundidade da vista de referência). O processo de refinamento é denominado de NBDV orientado para profundidade (DoNBDV) e pode incluir as duas etapas seguintes. Em primeiro lugar, localizar um bloco de profundidade correspondente pelo vetor de disparidade derivado na vista de profundidade de referência codificada anteriormente, tal como a vista base; o tamanho do bloco de profundidade correspondente é idêntico ao da PU atual. Em segundo lugar selecionar um valor de profundidade de quatro pixels de canto do bloco de profundidade correspondente (devido à adoção de Chang et alii, "3D-CE2.h related: Simplified DV derivation for DoNBDV and BVSP," Equipe de Colaboração Conjunta em Desenvolvimento de Extensões de Codificação de Vídeo 3D do ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 4°. Encontro: Incheon, KR, 20-26 de abril de 2013, documento JCT3V-D0138 (daqui por diante "JCT3V-D0138")) e converter o valor de profundidade selecionado no

componente horizontal do vetor de disparidade refinado. O componente vertical do vetor de disparidade fica inalterado. O JCT3V-D0138 está disponível em http://phenix.it-sudparis.eu/jct3vdoc_end_user/current_document.php?id=823.

[0102] Na 3D-HEVC, o processo de construção para listas de candidatos à fusão difere do processo de construção para listas de candidatos à fusão utilizado na HEVC. Por exemplo, com base no vetor de disparidade derivado do processo NBDV ou DoNBDV, o codificador de vídeo pode adicionar um novo candidato a vetor de movimento (isto é, um Candidato a Vetor de Movimento Predito Inter-Vista (IPMVC)) se disponível, aos modos AMVP aos saltos/de fusão. Em outras palavras, o codificador de vídeo pode incluir um IPMVC em uma lista de candidatos à fusão ou em uma lista de candidatos a AMVP. O IPMVC pode especificar as informações de movimento de um bloco de referência em uma vista de referência. Por exemplo, um IPMVC pode especificar um ou mais vetores de movimento temporais, assim como indicadores de direção de predição e índices de referência.

[0103] Para o modo de fusão/por saltos, o codificador de vídeo pode derivar um vetor de movimento predito inter-vista pelas etapas seguintes. Em primeiro lugar, o codificador de vídeo pode localizar um bloco correspondente de PU/CU atual em uma vista de referência da mesma unidade de acesso pelo vetor de disparidade. Em segundo lugar, se o bloco correspondente não for intra-codificado e não predito inter-vista e sua imagem de referência tiver um valor de total de ordens de imagem (POC) igual ao de uma entrada na mesma lista de imagens de referência da PU/CU atual, o codificador de vídeo pode derivar suas informações de movimento (direção de predição, imagens de referência e vetores de movimento) depois de

converter o índice de referência baseado em POC, de modo a ser o vetor de movimento predito inter-vista.

[0104] A Figura 4 mostra um exemplo do processo de derivação do candidato a vetor de movimento predito inter-vista. Em particular, a Figura 4 é uma ilustração conceitual que mostra a derivação de um candidato a vetor de movimento predito inter-vista para o modo de fusão/por saltos. No exemplo da Figura 4, a PU atual 40 ocorre na vista V1 em uma ocorrência de T1. Uma PU de referência 42 para a PU atual 40 ocorre em uma vista diferente da PU atual (isto é, a vista V0) e na mesma ocorrência de tempo da PU atual 40 (isto é, a ocorrência de tempo T1). No exemplo da Figura 4, a PU de referência 42 é inter-predita bidirecionalmente. Consequentemente, a PU de referência 42 tem um primeiro vetor de movimento 44 e um segundo vetor de movimento 46. O vetor de movimento 44 indica uma posição em uma imagem de referência 48. A imagem de referência 48 ocorre na vista V0 e na ocorrência de tempo T0. O vetor de movimento 46 indica uma posição na imagem de referência 50, a imagem de referência ocorre na vista V0 na ocorrência de tempo T3.

[0105] O codificador de vídeo pode gerar, com base nas informações de movimento da PU de referência 42, um IPMVC para inclusão para lista de candidatos à fusão da PU atual 40. O IPMVC pode ter um primeiro vetor de movimento 52 e um segundo vetor de movimento 54. O vetor de movimento 52 corresponde ao vetor de movimento 44 e o vetor de movimento 54 corresponde ao vetor de movimento 46. O codificador de vídeo. O codificador de vídeo gera o IPMVC de modo que um primeiro índice de referência do IPMVC indique uma posição na RefPicList0 na PU atual 40 de uma imagem de referência (isto é, a índice de referência 56) que ocorre na mesma ocorrência de tempo da imagem de

referência 48 (isto é, a ocorrência de tempo T0). No exemplo da Figura 4, a imagem de referência 56 ocorre na primeira posição (isto é, Ref0) na RefPicList0 na PU atual 40. Além disto, o codificador de vídeo gera o IPMVC de modo que um segundo índice de referência do IPMVC indique uma posição na RefPicList1 para a PU atual 40 de uma imagem de referência (isto é, a imagem de referência 58) que ocorre na mesma ocorrência de tempo da imagem de referência 50. Assim, no exemplo da Figura 4, o índice de referência RefPicList0 do IPMVC pode ser igual a 0. No exemplo da Figura 4, uma imagem de referência 59 ocorre na primeira posição (isto é, Ref0) na RefPicList1 para a PU atual 40 e a imagem de referência 58 ocorre na segunda posição (isto é, Ref1) na RefPicList1 para a PU atual 40. Por conseguinte, o índice de referência RefPicList1 do IPMVC pode ser igual a 1.

[0106] Assim, no exemplo da Figura 4, um vetor de disparidade é calculado encontrando-se o bloco correspondente 42 em uma vista diferente, (vista 0 ou V0, por exemplo) da PU atual 40 da vista atualmente codificada (vista 1 ou V1). Se o bloco correspondente 42 não for intra-codificado e não predito inter-vista, e sua imagem de referência tiver um valor POC que está na lista de imagem de referência da PU atual 40 (como, por exemplo, Ref0, Lista 0; Ref0, Lista 1; Ref1, Lista 1, conforme mostrado na Figura 4), então as informações de movimento para o bloco crt 42 são utilizadas como um vetor de movimento predito inter-vista. O codificador de vídeo pode escalonar o índice de referência com base no POC.

[0107] Além disto, quando gera uma lista de candidatos à fusão (ou em alguns exemplos, uma lista de candidatos a MVP) para um bloco (PU, por exemplo), o codificador de vídeo pode converter o vetor de disparidade

do bloco em um candidato a vetor de movimento de disparidade de inter-vista (IDM). O IDMVC pode especificar o vetor de disparidade do bloco. O codificador de vídeo pode adicionar o IDMVC à lista de candidatos à fusão (ou em alguns exemplos, lista de candidatos a AMVP) em uma posição diferente do IPMVC. Alternativamente, em alguns exemplos, o codificador de vídeo pode adicionar o IDMVC à lista de candidatos à fusão (ou em alguns exemplos, lista de candidatos a AMVP) na mesma posição do IPMVC quando o IDMVC estiver disponível. Neste contexto, ou um IPMVC ou um IDMVC pode ser chamado de "candidato inter-vista". Em alguns exemplos, no modo de fusão/por saltos o codificador de vídeo insere sempre o IPMVC, se disponível, antes de todos os candidatos à fusão espaciais e temporais na lista de candidatos à fusão. Em alguns de tais exemplos, o codificador de vídeo pode inserir o IDMVC antes do candidato à fusão especial derivado de A_0 .

[0108] Thirumalai, at alii, "Derivação de candidatos à fusão a partir de deslocamento de vetores", Equipe de Colaboração Conjunta em Extensões de Codificação de Vídeo 3D do ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 5°. Encontro: Vienna, AU, 27 de julho - 2 de agosto de 2013, documento no. JCT3V-E0126 (daqui por diante "JCT3V-E0126")) descreve derivação de candidatos à fusão a partir de deslocamento de vetores. O JCT3V-E0126 está disponível em http://phenix.it-sudparis.eu/jct3vdoc_end_user/current_document.php?id=1140. Devido à adoção do JCT3V-E0126, um candidato a mais, chamado de "candidato deslocado" ou "IvMVC deslocado" pode ser derivado com um vetor de disparidade deslocada. Tal candidato pode ser um IPMVC derivado de um bloco de referência em uma vista de referência com vetores de disparidade deslocados ou derivado do primeiro candidato à

fusão especial disponível que inclui um vetor de disparidade ou IDMC. Etapas detalhadas para gerar o candidato adicional e inserção na lista de candidatos à fusão seção descrito da maneira seguinte.

[0109] Em primeiro lugar, um codificador de vídeo desloca o vetor de disparidade DV em $(PuWidth/2*4+4)$, $(PuHeight/2*4+4)$. O codificador de vídeo utiliza o DV para derivar um candidato IvMC deslocado da vista de referência. Aqui, o tamanho da PU atual é $PuWidth \times PuHeight$. Se o IvMC deslocado estiver disponível, o codificador de vídeo pode saltar a etapa 2 (isto é, a segunda etapa descrita em seguida) e se este IvMC deslocado não for idêntico a IvMC sem deslocamento de vetor de disparidade, o codificador de vídeo insere o IvMC deslocado na lista de candidatos à fusão imediatamente antes do candidatos à fusão temporal.

[0110] Em segundo lugar, o codificador de vídeo pode derivar um candidato, denotado como Vetor de Movimento Deslocado por Disparidade (DSMV). O codificador de vídeo pode configurar o DSMV para ser o candidato adicional. Se o DSMV estiver disponível, o codificador de vídeo pode inserir o DSMV na lista de candidatos à fusão na mesma posição de um IvMC deslocado. O codificador de vídeo pode derivar o DSMV da maneira seguinte. Em primeiro lugar, o codificador de vídeo identifica o primeiro vetor de movimento de disparidade disponível (DMV) que corresponde à $RefPicList0$ a partir dos blocos vizinhos espaciais. Em segundo lugar, se o DMV estiver disponível, o codificador de vídeo configura o componente horizontal do vetor de movimento na lista 0 como DMV deslocado em quatro e o codificador de vídeo mantém o componente vertical do vetor de movimento inalterado ou reconfigura o componente vertical do vetor de movimento em zero, dependendo de BVSP ser ou não habilitado. Os índices de referência e os

vetores de movimento na Lista 1 são herdados diretamente. Caso contrário (isto é, se o DMV não estiver disponível), o codificador de vídeo configura o componente horizontal do vetor de movimento na Lista 0 e na Lista 1 como o DV deslocado em quatro e o codificador de vídeo configura ambos os componentes verticais dos vetores de movimento na Lista 0 e na Lista 1 em 0.

[0111] Tian et alii, "CE1.h: Predição por Síntese de Vista Retroativa com a Utilização de Bloco Vizinho", Equipe de Colaboração Conjunta em Extensões de Codificação de Vídeo 3D do ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 3°. Encontro: Genebra, CH, 17-23 de janeiro de 2013, documento no. JCT3V-C0152 (daqui por diante "JCT3V-C0152")) descreve a predição por síntese de vista retroativa com a utilização de blocos vizinhos. O JCT3V-C0152 está disponível em http://phenix.it-sudparis.eu/jct2vdoc_end_user/current_document.php?id=594. A abordagem VSP retroativa-de deformação proposta no JCT3V-C0152 foi adotada no 3°. Encontro JCT-3V. A idéia básica deste VSP de retroação-de deformação proposto no JCT3V-C0152 é idêntica à do VSP baseada em blocos na 3D-AVC. Ambas estas duas técnicas utilizam o VSP de retroação-deformação e baseado em blocos para evitar a transmissão das diferenças entre vetores de movimento que utilizam vetores de movimento mais precisos. Os detalhes de implementação são diferentes devido a plataformas diferentes. Os parágrafos seguintes utilizam o termo "BVSP" para indicar a abordagem VSP de retroação-deformação na 3D-HEVC.

[0112] Em alguns desenhos do 3D-HTM, o modo BVSP só é suportado para um bloco inter-código ou no modo por saltos ou no modo de fusão. O modo BVSP não é permitido para um bloco codificado no modo AMVP. Em vez da

transmissão de um indicador para indicar a utilização do modo BVSP, um candidato à fusão adicional (isto é, o candidato à fusão BVSP) é introduzido e cada candidato é associado a um indicador de BVSP. Conforme indicado acima, o codificador de vídeo 20 pode sinalizar um índice de fusão (merge_idx, por exemplo) em um fluxo de bits e o decodificador de vídeo 30 pode obter o índice de fusão do fluxo de bits. Quando o índice de fusão decodificado corresponde a um candidato à fusão BVSP, a PU atual utiliza o modo BVSP. Além disto,, quando o índice de fusão decodificado corresponde ao candidato à fusão BVSP, para cada sub-bloco dentro da PU atual, o codificador de vídeo pode derivar um vetor de movimento de disparidade para o sub-bloco convertendo um valor de profundidade em uma vista de referência de profundidade.

[0013] A configuração de indicadores de BVSP pode ser definida da maneira seguinte. Quando um bloco vizinho especial utilizado para derivar um candidato à fusão especial é codificado como o modo BVSP, as informações de movimento conexas são herdadas pelo bloco atual como no modo de fusão convencional. Além disto, este candidato à fusão especial é etiquetado com um indicador de BVSP igual a 1. Para o candidato à fusão BVSP recém-introduzido, o indicador de DVSP é fixado em 1. Para todos os outros candidatos à fusão, os indicadores de BVSP conexos são fixados em 0.

[0114] Conforme indicado acima, na 3D-HEVC, um codificador de vídeo pode derivar um novo candidato (isto é, um candidato à fusão BVSP) e pode inserir o candidato à fusão BVSP na lista de candidatos à fusão. O codificador de vídeo pode fixar os índices de referência e vetores de movimento correspondentes para o candidato à fusão BVSP pelo método seguinte. Em primeiro lugar, o codificador de

vídeo pode obter o índice de vista (denotado por `refVIdxLX`) do vetor de disparidade derivado do NBDV. Em segundo lugar, o codificador de vídeo pode obter a lista de imagens de referência `RefPicListX` (ou `RefPicList0` ou `RefPicList1`) que está associada à imagem de referência com o índice de ordem de vista igual a `refVIdxLX`). O codificador de vídeo pode utilizar o índice de referência e o vetor de disparidade correspondente do processo NBDV como as informações de movimento do candidato à fusão BVSP em `RefPicListX`.

[0115] Em terceiro lugar, se a fatia atual for uma fatia B, o codificador de vídeo pode verificar a disponibilidade de uma imagem de referência inter-vista, um índice de ordem de vistas (denotado por `refVIdxLY`) não igual a `refVIdxLX` na outra lista de imagens de referência que não a `RefPicListX` (isto é, `RefPicListY` com `Y` sendo $1-X$). Se tal imagem de referência inter-vista diferente for encontrada, o codificador de vídeo aplica VSP bi-preditiva. Entrementes, o codificador de vídeo utiliza o índice de referência correspondente da índice de referência inter-vista diferente e o vetor de disparidade escalonado de um processo NBDV como as informações de movimento do candidato à fusão DVSP na `RefPicListY`. O codificador de vídeo pode utilizar o bloco de profundidade da vista com índice de ordem de vistas igual a `refVIdxLX` como a informação de profundidade do bloco atual (no caso de primeiro ordem de codificação por textura), e o codificador de vídeo pode acessar as duas imagens de referência inter-vista diferentes (cada uma de uma lista de imagens de referência) por meio de um processo de retroação- deformação e não transitório ponderadas de modo a se obter o preditor de VSP retroativo final. Caso contrário, o codificador de vídeo aplica VSP uni-preditiva com `RefPicListX` como a lista de imagens de referência para predição.

[0116] No 3D-HTM, a primeira codificação por textura é aplicada a condições de teste comuns. Portanto, a vista de profundidade não base correspondente está indisponível quando se decodifica uma lista de textura não base. Portanto, as informações de profundidade são estimadas e utilizadas para executar BVSP. De modo a estimar as informações de profundidade para um bloco, um codificador de vídeo pode primeiro derivar um vetor de disparidade dos blocos vizinhos e em seguida utilizar o vetor de disparidade derivado para obter um bloco de profundidade de uma vista de referência. No modelo de teste 3D-HTM 8.0, existe um processo para derivar um preditor de vetor de disparidade, conhecido como MBDV (Vetor de Disparidade de Blocos Vizinhos). Que (dv_x, dv_y) denote o vetor de disparidade identificado a partir da função NBDV e que a posição do bloco atual seja $(bloco_x, bloco_y)$.

[0117] Em alguns exemplos da BVSP unipreditiva, um codificador de vídeo busca um bloco de profundidade na posição de topo à esquerda $(bloco_x + dv_x, bloco_y + bloco_x)$ na imagem de profundidade na vista de referência. O bloco atual é primeiro dividido em vários sub-blocos, cada um deles tendo o mesmo tamanho de $W \times H$. Para cada sub-bloco com o tamanho $W \times H$ o codificador de vídeo utiliza um sub-bloco de profundidade correspondente dentro do bloco de profundidade buscado e converte o valor de profundidade máximo dos quatro pixels de canto do sub-bloco de profundidade em um vetor de movimento de disparidade. O codificador de vídeo utiliza então o vetor de movimento de disparidade derivado para cada sub-bloco para compensação de movimento. A Figura 5 mostra as três etapas de como um bloco de profundidade da vista de referência é localizado e em seguida utilizado para BVSP (também chamada "predição BVSP").

[0118] Em particular, a Figura 5 é um diagrama conceitual que mostra a derivação de blocos de profundidade a partir de uma vista de referência para efetuar predição BVSP. Em alguns exemplos de BVSP de bi-predição, quando há várias imagens de referência inter-vista de vistas diferentes na RefPicList0 e na RefPicList1, o codificador de vídeo aplica VSP bi-preditiva. Ou seja, o codificador de vídeo pode gerar dois preditores VSP a partir de cada lista de referência, conforme descrito acima. O codificador de vídeo pode então dividir proporcionalmente os dois preditores VSP de modo a obter o preditor VSP final.

[0119] No exemplo da Figura 5, um codificador de vídeo está codificando a imagem de textura atual 60. A imagem de textura atual 60 é rotulada como "imagem de textura dependente", uma vez que a imagem de textura atual 60 depende de uma imagem de textura de referência sintetizada 62. Em outras palavras, pode ser necessário que o codificador de vídeo sintetize a imagem de textura de referência 62 (ou partes dela) de modo a codificar a imagem de textura atual 60. A imagem de textura de referência 62 e a imagem de textura atual 60 estão na mesma unidade de acesso, mas estão em vistas diferentes.

[0120] De modo a sintetizar a imagem de textura de referência 62 (ou partes dela), o codificador de vídeo pode processar blocos (isto é, unidades de vídeo) da imagem de textura atual 60. No exemplo da Figura 5, o codificador de vídeo está processando o bloco atual 64. Quando o codificador de vídeo processa o bloco atual 64, o codificador de vídeo pode executar o processo de derivação NBDV para derivar um vetor de disparidade para o bloco atual 64. No exemplo da Figura 5, por exemplo, o codificador de vídeo identifica um vetor de disparidade 66 de um bloco 68 que é vizinho do bloco atual 64. A

identificação do vetor de disparidade 66 é mostrada como Etapa 1 da Figura 5. Além disto, no exemplo da Figura 5, o codificador de vídeo determina, com base no vetor de disparidade 66, um vetor de disparidade 69 no bloco atual 64. Por exemplo, o vetor de disparidade 69 pode ser uma cópia do vetor de disparidade 66. A cópia do vetor de disparidade 66 é mostrada como Etapa 2 da Figura 5.

[0121] O codificador de vídeo pode identificar, com base no vetor de disparidade 69 do bloco atual 64, um bloco de referência 70 em uma imagem de profundidade de referência 72. A imagem de profundidade de referência 72, a imagem de textura atual 60 e a imagem de textura de referência 62 podem estar, cada uma, na mesma unidade de acesso. A imagem de profundidade de referência 72 e a imagem de textura de referência 62 podem estar na mesma vista. O codificador de vídeo pode determinar, com base nos valores de amostra de textura do bloco atual 64 e nos valores de amostra de profundidade do bloco de referência 70, valores de amostra de textura da imagem de textura de referência 72. O processo de determinar os valores de amostra de textura pode ser referido como deformação retroativa. A seção W.8.5.2.2.7 do Modelo de teste 3D-HEVC 3 descreve o processo de deformação retroativo. A deformação retroativa é mostrado como a Etapa 3 da Figura 5. Desta maneira, a Figura 5 mostra as três etapas de como um bloco de profundidade da vista de referência é localizado e em seguida utilizado para predição BVSP.

[0122] O tamanho de compensação de movimento (isto é, $W \times H$ conforme descrito acima) utilizado na BVSP pode ser ou 8×4 ou 4×8 . Para se determinar o tamanho da compensação de movimento, é aplicada a regra seguinte. Para

cada bloco 8x8, o codificador de vídeo verifica quatro cantos do bloco 8x8 de profundidade correspondente e:

```
if (vdepth[TL] < vdepth[BR] ? 0 : 1 != (vdepth[TR] < vdepth[BL] ? 0 : 1))
```

```
    utiliza partição 4x8 (X=4, H=80
```

```
    caso contrário
```

```
    utiliza partição 8x4 (W=8, H=4)
```

Figura 6 é um diagrama conceitual que mostra quatro pixels de canto de um bloco de profundidade 8x8.

[0123] O número máximo de candidato à fusão e o processo de construção de listas de fusão para a 3D-HEVC são descritos nos parágrafos seguintes. Em algumas versões da 3D-HEVC, o número total de candidatos na lista de fusão é até seis e `five_minus_max_num_merge_cand` é sinalizado em cabeçalho de fatia para especificar o número máximo dos candidatos à fusão subtraído de cinco. `five_minus_max_num_merge_cand` está na faixa de zero a 5, inclusive. `five_minus_max_num_merge_cand` especifica o número máximo de candidatos a preditor de vetor de movimento (MVP) de fusão (isto é, candidatos à fusão) suportados na fatia subtraído de 5. Um codificador de vídeo pode computar o número máximo de candidatos AMVP de fusão (isto é, `MaxNumMergeCand`) como:

$$\text{MaxNumMergeCand} = 5 - \text{five_minus_max_num_merge_cand} + \text{iv_mv_pred_flag[nuh_layer_id]} \quad (\text{H-1})$$

Em tais versões da 3D-HEVC, o valor de `five_minus_max_num_merge_cand` será limitado de modo que `MaxNumMergeCand` esteja na faixa de 0 a $(5 + \text{iv_mv_pred_flag[nuh_layer_id]})$, inclusive.

[0124] Além disto, em tais versões da 3D-HEVC, um elemento de sintaxe `iv_mv_pred_flag[layerId]` indica se

a predição de parâmetros de movimento inter-vista é utilizada no processo de decodificação da camada com `nuh_layer_id` igual `layerId`. `iv_mv_pred_flag [layerId]` igual a 0 especifica que a predição de parâmetros de movimento inter-vista não é utilizada para a camada com `nuh_layer_id` igual a `layerId`. Quando não presente, o valor de `iv_mv_pred_flag [layerId]` será inferido como sendo igual a 0.

[0125] O processo de construção da lista de candidatos à fusão na 3D-HEVC pode ser definido da maneira seguinte:

1. Inserção de IPMVC: Quando a predição de movimento inter-vista é aplicada, o codificador de vídeo deriva um IPMVC pelo procedimento descrito acima. Se o IPMVC estiver disponível, o codificador de vídeo insere o IPMVC na lista de fusão (isto é, a lista de candidatos à fusão).

2. Processo de derivação para candidatos à fusão espaciais e inserção de IDMVC na 3D-HEVC

O codificador de vídeo verifica as informações de movimento de PUs vizinhas espaciais na ordem seguinte: A_1 , B_1 , B_0 , A_0 ou B_2 . Além disto, o codificador de vídeo pode efetuar poda restrita pelos procedimentos seguintes:

- Se A_1 (isto é, um candidato à fusão derivado da PU vizinha especial A_1) e o IPMVC tiverem os mesmos vetores de movimento e os mesmos índices de referência, o codificador de vídeo não insere A_1 na lista de candidatos (isto é, a lista de candidatos à fusão). Caso contrário, o codificador de vídeo insere A_1 na lista (isto é, a lista de candidatos à fusão)

- Se B_1 e A_1 /IPMVC tiverem os mesmos vetores de movimento e os mesmos índices de referência, o codificador de vídeo não insere B_1 (isto é, um candidato à fusão)

derivado da PU vizinha especial B_1) na lista de candidatos (isto é, a lista de candidatos à fusão). Caso contrário, o codificador de vídeo insere B_1 na lista (isto é, a lista de candidatos à fusão).

- Se B_0 (isto é, um candidato à fusão derivado da PU vizinha especial B_0) estiver disponível, o codificador de vídeo adiciona B_0 à lista de candidatos (isto é, a lista de candidatos à fusão).

- Quando a predição de movimento inter-vista é aplicada, o codificador de vídeo deriva um IDMVC pelo procedimento descrito acima. Se o IDMVC estiver disponível, e o IDMV for diferente dos candidatos derivado de A_1 e B_1 , o codificador de vídeo insere o IDMVC na lista de candidatos (isto é, a lista de candidatos à fusão).

- Se BVSP for habilitado para toda a imagem ou para a fatia atual, então o codificador de vídeo insere o candidato à fusão BVSP na lista de candidatos à fusão.

- Se A_0 (isto é, o candidato à fusão derivado da PU vizinha especial A_0) estiver disponível, o codificador de vídeo adiciona A_0 à lista de candidatos (isto é, a lista de candidatos à fusão).

- Se B_2 (isto é, o candidato à fusão derivado da PU vizinha especial B_2) estiver disponível, o codificador de vídeo adiciona B_2 à lista de candidatos (isto é, a lista de candidatos à fusão).

- Quando a predição de movimento inter-vista é aplicada, o codificador de vídeo insere um candidato deslocado (isto é, DSMV), se disponível, conforme descrito acima.

3. Processo de derivação para candidato fusão temporal

O processo de derivação para os candidatos à fusão temporais é semelhante ao processo de derivação para

candidatos à fusão temporais na HEVC, onde as informações de movimento da PU co-localizada são utilizadas. Entretanto, o índice de referência-alvo do candidato à fusão temporal pode ser alterado em vez de ser fixado como sendo 0. O índice de referência-alvo do candidato à fusão temporal é um índice de referência de uma imagem de referência na qual o codificador de vídeo baseia o candidato à fusão temporal. Quando o índice de referência-alvo igual a zero corresponde a uma imagem de referência temporal (isto é, uma imagem de referência na mesma vista da PU atual), enquanto o vetor de movimento da PU co-localizada indica uma imagem de referência inter-vista, o codificador de vídeo altera o índice de referência-alvo para um índice que corresponde à primeira entrada de uma índice de referência inter-vista na lista de imagens de referência. Em outras palavras, o codificador de vídeo altera o índice de referência-alvo de modo que o índice de referência-alvo indique a primeira imagem de referência inter-vista na lista de imagens de referência. Entretanto, quando o índice de referência-alvo igual a 0 corresponde a uma imagem de referência inter-vista enquanto o vetor de movimento da PU co-localizada indica uma imagem de referência temporal, o codificador de vídeo altera o índice de referência-alvo para outro índice que corresponde à primeira entrada de uma imagem de referência temporal na lista de imagens de referência. Em outras palavras, o codificador de vídeo altera o índice de referência-alvo de modo que o índice de referência-alvo indique a primeira imagem de referência temporal na lista de imagens de referência.

4. Processo de derivação para candidatos à fusão bi-preditiva combinada na 3D-HEVC

Se o número total de candidatos derivado das três etapas acima for menor que o número máximo de candidatos, o codificador de vídeo executa o mesmo processo definido na HEVC com duas alterações:

- o Em primeiro lugar, as condições de obtenção de um candidato à fusão bi-preditiva combinada não alteradas adicionando-se a verificação de indicadores de BVSP associados ao primeiro/segundo candidato.

- o Em segundo lugar, é modificada a especificação do 10CandIdx e 11CandIdx. A relação entre combIdx 10CandIdx e 11CandIdx é definida na Figura 7, que é uma tabela que apresenta uma especificação de 10CandIdx e 11CandIdx na 3D-HEVC.

5. Processo de derivação para zero candidatos à fusão de vetor de movimento

- O codificador de vídeo executa o mesmo procedimento definido na HEVC (e descrito acima) para derivar os zero candidatos à fusão de vetor de movimento.

[0126] O desenho do processo de derivação de candidatos à fusão bi-preditiva combinada na 3D-HEVC pode ter um ou mais problemas potenciais. Por exemplo, o desenho atual do processo de derivação de candidatos à fusão bi-preditiva combinada na 3D-HEVC pode exigir unidades lógicas adicionais a serem adicionadas para verificar os indicadores de BVSP dos 1 e segundo candidatos à fusão existentes utilizados para construir um candidato à fusão bi-preditiva combinada. Entretanto, a verificação adicional dos indicadores de BVSP não ajuda em termos de eficácia de codificação. Assim, a verificação adicional dos indicadores de BVSP aumenta a complexidade.

[0127] Em outro exemplo dos problemas potenciais associados ao processo de derivação de candidatos à fusão bi-preditiva combinada na 3D-HEVC, a re-

utilização direta do processo de derivação HEVC de candidatos à fusão bi-preditiva combinada pode resultar em um processo de decodificação imprevisível. O processo de derivação HEVC de candidatos à fusão bi-preditiva combinada só pode aceitar até quatro candidatos à fusão para gerar novos candidatos. Entretanto, se este processo for utilizado na 3D HEVC diretamente, pode haver o caso de que cinco candidatos à fusão são utilizados como entrada para este processo. Quando há até quatro candidatos à fusão apenas doze combinações possíveis estão disponíveis e, portanto, elas são definidas neste processo em uma tabela. Entretanto, quando cinco candidatos à fusão estão disponíveis, pode haver vinte combinações possíveis, embora a tabela atual (isto é, a Tabela 1 acima) não suporte tantas combinações.

[0128] Uma ou mais das técnicas desta revelação referem-se ao processo de derivação de candidatos à fusão bi-preditiva combinada na 3D-HEVC. De acordo com uma técnica exemplar desta revelação, o desenho do processo de derivação de candidatos à fusão bi-preditiva combinada na 3D-HEVC é substituído pelo utilizado na HEVC. Portanto, não há necessidade de verificar os indicadores de PVSP no processo de derivação de candidatos à fusão bi-preditiva combinada. Em outras palavras, o processo de gerar a lista de candidatos à fusão ocorre sem a verificação de quaisquer indicadores de BVSP. A não verificação dos indicadores de BVSP no processo de derivação de candidatos à fusão bi-preditiva combinada pode reduzir a complexidade do processo de codificação/decodificação sem provocar um impacto negativo significativo sobre a eficácia de codificação.

[0129] Desta maneira, esta revelação pode proporcionar um método para codificar dados associados a vídeo 3D. Este método pode compreender gerar uma lista de

candidatos à fusão para configurar um bloco de vídeo associado a vídeo 3D de acordo com um processo de derivação de listas de fusão. A lista inclui um ou mais candidatos à fusão bi-preditiva. O processo de derivação de listas de fusão para vídeo 3D corresponde ao mesmo processo de derivação de listas de fusão e está associado a vídeo não 3D.

[0130] Além disto, de acordo com uma ou mais técnicas desta revelação, quando se chama o processo de derivação de candidatos à fusão bi-preditiva combinada na HEVC, em vez de se verificar apenas que o tipo de fatia é igual à fatia B, outra condição deve ser também satisfeita, isto é, o número de candidatos à fusão disponíveis inseridos na lista de candidatos à fusão deve ser menor do que cinco.

[0131] Assim, em alguns exemplos, um codificador de vídeo pode codificar dados associados a vídeo 3D. Como parte da codificação dos dados, o codificador de vídeo pode gerar uma lista de candidatos à fusão para codificar um bloco de vídeo (uma PU, por exemplo) do vídeo 3D. Como parte da geração da lista de candidatos à fusão, o codificador de vídeo pode determinar se o número de candidatos à fusão na lista é menor do que 5. Em resposta à determinação de que o número de candidatos à fusão na lista é menor do que 5, o codificador de vídeo pode derivar um ou mais candidatos à fusão bi-preditiva combinada. Neste exemplo, cada candidato à fusão bi-preditiva combinada do candidato ou candidatos à fusão bi-preditiva combinada corresponde a um respectivo par (um par ordenado, por exemplo) de candidatos à fusão já na lista. O respectivo candidato à fusão bi-preditiva combinada é uma combinação de um vetor de movimento de um primeiro candidato à fusão do respectivo par e vetor de movimento de

um segundo candidato à fusão do respectivo par. O vetor de movimento do primeiro candidato à fusão e o vetor de movimento do segundo candidato à fusão referem-se a imagens em lista de imagens de referência diferentes. O codificador de vídeo pode incluir o candidato ou candidatos à fusão bi-preditiva combinada na lista de candidatos à fusão.

[0132] Alternativamente, em alguns exemplos, antes que o processo de derivação de candidatos à fusão bi-preditiva combinada seja chamado, o número máximo de candidatos MVP à fusão, `MaxNumMergeCand` é fixado da maneira seguinte:

$$\text{MaxNumMergeCand} = 5 - \text{five_minus_max_num_merge_cand}.$$

Depois que o processo de derivação de candidatos à fusão bi-preditiva combinada é chamado o `MaxNumMergeCand` é fixado novamente no valor, como na 3D-HEVC:

$$\text{MaxNumMergeCand} = 5 - \text{five_minus_max_num_merge_cand} + \text{iv_mv_pred_flag[nuh_layer_id]}.$$

`nuh_layer_id` é o elemento de sintaxe que especifica um identificador de camada. Assim, em alguns de tais exemplos, antes de derivar o candidato ou candidatos à fusão bi-preditiva combinada, um codificador de vídeo pode reconfigurar o número máximo de candidatos à fusão como sendo igual a 5 menos o valor de um primeiro elemento de sintaxe. O primeiro elemento de sintaxe especifica o número máximo de candidatos à fusão suportados em uma fatia subtraído de 5. Depois de derivar o candidato ou candidatos à fusão bi-preditiva combinada. O codificador de vídeo pode fixar o número máximo de candidatos à fusão em 5 menos o valor do primeiro elemento de sintaxe mais o valor de um segundo elemento de sintaxe, em que o segundo elemento de sintaxe indica se a predição de parâmetros de movimento inter-vista é utilizada no processo de decodificação de uma camada.

[0133] É quando MaxNumMergeCand é igual a 6 e há cinco candidatos antes que o processo de derivação de candidatos à fusão bi-preditiva combinada na HEVC seja chamado, um candidato zero (com componentes de índice de referência e vetor de movimento todos sendo 0) é sempre gerado e inserido na lista de candidatos à fusão, conforme especificado no sub-parágrafo 8.5.3.2.4 do Rascunho Operacional HEVC 10.

[0134] Alternativamente, o codificador de vídeo fixa MaxNumMergeCand em 5 antes da chamada do processo para determinar candidatos à fusão bi-preditiva e o codificador de vídeo só considera os primeiros quatro candidatos como entrada deste processo. Depois que o codificador de vídeo chama o processo para determinar candidatos à fusão bi-preditiva, o codificador de vídeo põe o candidato à fusão bi-preditiva recém gerado, se disponível, no final da lista de candidatos à fusão. Assim, o candidato à fusão bi-preditiva recém gerado segue-se ao 4º. na lista de candidatos à fusão, que o codificador de vídeo não considerou como parte da entrada do processo para determinar candidatos à fusão bi-preditiva. Posteriormente, neste exemplo, o MaxNumMergeCand é fixado novamente em 6. Quando o processo para determinar candidatos à fusão bi-preditiva não apresenta um novo candidato à fusão bi-preditiva, o codificador de vídeo gera um candidato zero e insere o candidato zero na lista de candidatos à fusão., conforme especificado no sub-parágrafo 8.5.3.2.4 no Rascunho Operacional HEVC 10. O Sub-parágrafo 8.5.3.2.4 no Rascunho Operacional HEVC 10 é reproduzido em seguida.

8.5.3.2.4 no Processo de derivação para zero candidatos à fusão de vetor de movimento

Entradas para este processo são:

- uma lista de candidatos à fusão mergeCandList,

- os índices de referência `refIdxL0N` e `refIdxL1N` que cada candidato `N` em `mergeCandList`,

- os indicadores de utilização de lista de predição `predFlagL0N` e `predFlagL1N` de cada candidato `N` em `mergeCandList`,

- os vetores de movimento `mvL0N` e `mvL1N` de cada candidato `N` em `mergeCandList`,

- o número de elementos `numCurrMergeCand` dentro de `mergeCandList`,

Saídas deste processo são:

- a lista de candidatos à fusão `mergeCandList`,

- o número de elementos `numCurrMergeCand` dentro da `mergeCandList`,

- o índices de referência `refIdxL0candm` e cada novo candidatos `zeroCandm` adicionado a `mergeCandList` durante a chamada deste processo,

- os indicadores de utilização de lista de predição `predFlagL0zeroCandm` e `predFlagL1zeroCandm` de cada novo candidato `zeroCandm` adicionado à `mergeCandList` durante a chamada deste processo,

- os vetores de movimento `mvL0zeroCandm` e `mvL1zeroCandm` de cada novo candidato `zeroCandm` adicionado à `mergeCandList` durante a chamada deste processo.

A variável `numRefIdx` é derivada da maneira seguinte:

- Se `slice_type` for igual a `P`, `numRefIdx` é fixado como igual a `num_ref_idx_10_active_minus1 + 1`.

- Caso contrário (`slice_type` é igual a `B`), `numRefIdx` é fixado como igual a `Min(num_ref_idx_10_active_minus1 + 1, num_ref_idx_11_active_minus1 + 1)`.

Quando `numCurrMergeCand` é menor do que `MaxNumMergeCand`, a variável `numInputMergeCand` é fixada como igual `numCurrMergeCand`, a variável `zeroIdx` é fixada como igual a

zero e as etapas seguintes são repetidas até que numCurrMergeCand seja igual a MaxNumMergeCand:

1. Para a derivação dos índices de referência dos indicadores de utilização de lista de predição e dos vetores de movimento do zero candidato à fusão a vetor de movimento, se aplica o seguinte:

- Se slice_type for igual a P, o candidato zeroCand_m com m igual a (numCurrMergeCand 0 numInputMergeCand) é adicionado no final da mergeCandList, isto é, mergeCandList[numCurrMergeCand] é fixado como igual a zeroCand_m, e os índices de referência, os indicadores de utilização de lista de predição e os vetores de movimento de zeroCand_m são derivados da maneira seguinte e numCurrMergeCand é incrementado em 1:

$$\text{refIdxL0zeroCand}_m = (\text{zeroIdx} < \text{numRefIdx}) ? \text{zeroIdx} : 0 \quad (8-122)$$

$$\text{refIdxL1zeroCand}_m = -1 \quad (8-123)$$

$$\text{predFlagL0zeroCand}_m = 1 \quad (8-124)$$

$$\text{predFlagL1zeroCand}_m = 0 \quad (8-125)$$

$$\text{mvL0zeroCand}_m[0] = 0 \quad (8-126)$$

$$\text{mvL0zeroCand}_m[1] = 0 \quad (8-127)$$

$$\text{mvL1zeroCand}_m[0] = 0 \quad (8-128)$$

$$\text{mvL1zeroCand}_m[1] = 0 \quad (8-129)$$

$$\text{numCurrMergeCand} = \text{numCurrMergeCand} + 1 \quad (8-130)$$

- Caso contrário (slice_type é igual a B), o zeroCand_m com m igual a (numCurrMergeCand -

numInputMergeCand) é adicionado no final da mergeCandList, isto é, mergeCandList[numCurrMergeCand] é fixado como igual a zeroCand_m, e os índices de referência, os indicadores de utilização de lista de predição e os vetores de movimento de zeroCand_m são derivados da maneira seguinte e numCurrMergeCand é incrementado em 1:

$$\text{refIdxL0zeroCand}_m = (\text{zeroIdx} < \text{numRefIdx}) ? \text{zeroIdx} : 0 \quad (8-131)$$

$$\text{refIdxL1zeroCand}_m = (\text{zeroIdx} < \text{numRefIdx}) ? \text{zeroIdx} : 0 \quad (8-132)$$

$$\text{predFlagL0zeroCand}_m = 1 \quad (8-133)$$

$$\text{predFlagL1zeroCand}_m = 1 \quad (8-134)$$

$$\text{mvL0zeroCand}_m[0] = 0 \quad (8-135)$$

$$\text{mvL0zeroCand}_m[1] = 0 \quad (8-136)$$

$$\text{mvL1zeroCand}_m[0] = 0 \quad (8-137)$$

$$\text{mvL1zeroCand}_m[1] = 0 \quad (8-138)$$

$$\text{numCurrMergeCand} = \text{numCurrMergeCand} + 1 \quad (8-139)$$

2. A variável zeroIdx é incrementada em 1.

[0135] Assim, em alguns exemplos nos quais o número máximo de candidatos à fusão (MaxNumMergeCand, por exemplo) é igual a 6, um codificador de vídeo pode, em resposta à determinação de que há cinco candidatos à fusão na lista de candidatos à fusão antes do acréscimo de qualquer um do candidato ou candidatos à fusão bi-preditiva à lista, o codificador de vídeo pode incluir um candidato zero na lista. Os componentes de vetor de movimento do

candidato zero são iguais a 0 e o índice de referência do candidato zero é igual a 0.

[0136] A seção seguinte desta revelação descreve alguns detalhes de implementação exemplares compatíveis com as técnicas desta revelação no contexto da HEVC. São mostradas em seguida alterações em seções do Texto de Rascunho 3D-HEVC 1. Diversas partes mostradas em negrito sublinhado ou em itálico sublinhado podem corresponder a acréscimos a seções da HEVC, e partes mostradas com itálico circundados com colchetes quadrados duplos (`[[texto]]`, por exemplo) podem corresponder a supressões. As técnicas desta revelação podem corresponder, em alguns exemplos, nos acréscimos mostrados em negrito sublinhado e às supressões mostradas com itálico circundado por colchetes quadrados duplos.

H.8.5.3.2.1 Processo de derivação para vetores de movimento luma para modos de fusão

Este processo só é chamado quando `merge_flag[xP][xP]`, é igual a 1, onde (x_{Pb}, y_{Pb}) especificam a amostra de topo-à esquerda do bloco de predição luma atual com relação à amostra luma de topo-à esquerda da imagem atual.

As entradas para este processo são:

- a localizações luma (x_{Cb}, y_{Cb}) da amostra de topo-à esquerda do bloco de codificação luma atual com relação à amostra luma de topo-à esquerda da imagem atual,
- A localizações luma (x_{Pb}, y_{Pb}) da amostra de topo-à esquerda do bloco de predição luma atual com relação a amostra luma de topo-à esquerda da imagem atual,
- uma variável `nCbS` que especifica o tamanho do bloco de codificação luma atual,
- duas variáveis `nPbW` e `nPbH` que especifica a largura e a altura do bloco de predição luma,

- uma variável `partIdx` que especifica o índice da unidade de predição atual dentro da unidade de codificação atual.

As saídas deste processo são:

- os vetores de movimento luma `mvL0` e `mvL1`,
- os índices de referência `refIdxL0` e `refIdxL1`,
- os indicadores de utilização de lista de predição `predFlagL0` e `predFlagL1`,
- os indicadores de disponibilidade de vetor de movimento `ivpMvFlagL0` e `ivpMvFlagL1`,
- o indicador `vpsModelFlag` que especifica se a PU atual é codificada utilizando-se predição por síntese de vista,

A localização (`xOrigP`, `yOrigP`) e as variáveis `nOrigPbW` e `nOrigPbH` são derivadas para armazenar os valores de (`xPb`, `yPb`), `nPbW` e `nPbH` da maneira seguinte:

(`xOrigP`, `yOrigP`) é fixado como igual a (`xPb`, `xPb`) H-81

`nOrigPbW` = `nPbW` H-82

`nOrigPbH` = `nPbH` H-83

Quando `Log2ParMrgLevel` é maior que 2 e `nCbS` é igual a 8, (`xPb`, `yPb`), `nPbW`, `nPbH` e `partIdx` são modificados da maneira seguinte:

(`xPb`, `yPb`) = (`xCb`, `yCb`) (H-84)

`nPbW` = `nCbS` (H-85)

`nPbH` = `nCbS` (H-86)

`partIdx` = 0 (H-87)

Observação - Quando `Log2ParMrgLevel` é maior que 2 e `nCbS` é igual a 8, todas as unidades de predição da unidade de codificação atual compartilham uma única lista

de candidatos à fusão, que é idêntica à lista de candidatos à fusão da unidade de predição $2N \times 2N$.

Os vetores de movimento $mvL0$ e $mvL1$, os índices de referência $refIdxL0$ e $refIdxL1$ e os indicadores de utilização de predição $predFlagL0$ e $predFlagL1$ são derivados pelas etapas ordenadas seguintes:

1. O processo de derivação para candidatos à fusão a partir de partições de unidades de predição vizinhas no sub-parágrafo 8.5.3.2.2 é chamado com a localização do bloco de codificação luma (x_{Cb} , y_{Cb}), o tamanho do bloco de codificação n_{CbS} , a localização do bloco de predição luma (x_{Pb} , y_{Pb}), a largura do bloco de predição luma n_{PbW} , a altura do bloco de predição luma n_{PbH} e o índice de partição como entradas, e a saída sendo os indicadores de disponibilidade $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, $availableFlagB_1$, $availableFlagB_2$, os índices de referência $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, $refIdxLXB_1$, $refIdxLXB_2$, os indicadores de utilização de lista de predição $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, $predFlagLXB_1$ e $predFlagLXA_2$, e os vetores de movimento $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, $mvLXB_1$ e $mvLXB_2$ com X sendo 0 ou 1.

2. Os índices de referência para o candidato à fusão temporal, $refIdxLXCol$, com X sendo 0 ou 1, são fixados como iguais a zero.

3. O processo de derivação para predição de vetores de movimento luma temporais no sub-parágrafo 8.5.3.2.7 é chamado com a localização luma (x_{Pb} , y_{Pb}), a largura do bloco de predição luma n_{PbW} , a altura do bloco de predição luma n_{PbH} e a variável $refIdxL0Col$ como entradas, e a saída sendo o indicador de disponibilidade $availableFlagL0Col$ e o vetor de movimento temporal $mvL0Col$. As variáveis

availableFlagCol, predFlagL0Col e predFlagL1Col são derivadas da maneira seguinte:

$$\text{availableFlagCol} = \text{availableFlagL0Col} \quad (\text{H-88})$$

$$\text{predFlagL0Col} = \text{availableFlagL0Col} \quad (\text{H-89})$$

$$\text{predFlagL1Col} = 0 \quad (\text{H-90})$$

4. Quando slice_type é igual a B, o processo de derivação para predição de vetores de movimento luma temporais no sub-parágrafo H 8.5.3.2.7 é chamado com a localização luma (xPb, yPb), a largura do bloco de predição luma nPbW, a altura do bloco de predição luma nPbH e a variável de refIdxL1Col como entradas, e a saída sendo o indicador de disponibilidade availableFlagL1Col e o vetor de movimento temporal mvL1Col. As variáveis availableFlagCol e predFlagL1Col são derivadas da maneira seguinte:

$$\text{availableFlagCol} = \text{availableFlagL0Col} \mid \mid \text{availableFlagL1Col} \quad (\text{H-91})$$

$$\text{predFlagL1Col} = \text{availableFlagL1Col} \quad (\text{H-92})$$

5. Dependendo de iv_mv_pred_flag[nuh_layer_id], o seguinte se aplica.

- Se iv_mv_pred_flag [nuh_layer_id] for igual a 0, os indicadores a availableFlagIvMC, availableIvMCShift e availableFlagIvDC são fixados como iguais a 0.

- Caso contrário iv_mv_pred_flag [nuh_layer_id] é igual a 1), o processo de derivação para os candidatos à fusão inter-vista especificado no sub-parágrafo H 8.5.3.2.10 é chamado com a localização luma (xPb, yPb) as variáveis nPbW e nPbH como as entradas, e a saída sendo atribuída aos indicadores de disponibilidade availableFlagIvMC, availableIvMCShift, e availableFlagIvDC, aos índices de referência refIdxLXIvMC, refIdxLXIvMCShift,

e refIdxLXivDC, as indicadores de utilização de lista de predição predFlagLXivMC, predFlagLXivMCShift e predFlagLXivDC e aos vetores de movimento mvLXivMC, mvLXivMCShift e mvLXivDC (com X sendo 0 ou 1, respectivamente).

6. Dependendo de view_synthesis_pred_flag [nuh_layer_id], o seguinte se aplica.

- Se view_synthesis_pred_flag [nuh_layer_id] for igual a 0, o indicador availableFlagVSP é fixado como igual a 0.

- Caso contrário, (view_synthesis_pred_flag [nuh_layer_id] é igual a 1), a o processo de derivação para um candidato à fusão de predição por síntese de vista especificado no sub-parágrafo H. 8.5.3.2.13 é chamado coma as localizações luma (xCb, yCb) como entradas e saídas sendo o indicador de disponibilidade availableFlagVSP, os índices de referência refIdxL0VSP e refIdxL1VSP, os indicadores de utiliza de lista de predição predFlagL0VSP e predFlagL1VSP e os vetores de movimento mvL0VSP e mvL1VSP.

7. Dependendo de DepthFlag, o seguinte se aplica.

- Se DepthFlag for igual a 0, a variável availableFlagT é fixada como igual a 0.

- Caso contrário (DepthFlag é igual a 1) o processo de derivação para o candidato à fusão de textura especificado no sub-parágrafo H 8.5.3.2.14 é chamado como a localização luma (xPb, yPb), as variáveis nPbW e nPbH como as entradas e as saídas sendo o indicador availableFlagT, os indicadores de utilização de predição predFlagL0T e predFlagL1T, os índices de referência refIdxL0T, refIdxL1T, e os vetores de movimento mvL0T e mvL1T.

8. As listas de candidatos à fusão mergeCandList e mergCandIsVspFlag são construídas conforme especificado pelas etapas ordenadas seguintes:

a. A variável `mergCand` é fixada como igual a 0.
 b. Quando `availableFlagT` é igual a 1, a entrada `mergeCandList[numMergCand]` é fixada como igual a T, a entrada `mergCandIsVspFlag[numMergCand]` é fixada como igual a 0 e a variável `numMergCand` é aumentada e 1.

c. Quando `availableFlagIvMC` é igual a 1, a entrada `mergeCandList[numMergCand]` é fixada como igual a IvMC, a entrada `mergCandIvVspFlag[numMergCand]` é fixada como igual a 0 e a variável `numMergCand` é aumentada em 1.

d. Quando `availableFlagA1` é igual a 1, o seguinte se aplica:

- Quando a condição seguinte é verdadeira,
- `availableFlagT == 0 && availableFlagIvMC == 0`,
 ou uma ou mais das condições seguintes são verdadeiras com N sendo substituído por T e IvMC

- `availableFlagN == 1 && predFlagLXN ! = predFlagLXA1`, (com X sendo substituído por 0 e 1).

- `availableFlagN == 1 && mvLXN ! == mvLXA1` (com X sendo substituído por 0 e 1).

- `availableFlagN == 1 && refIdxLXN ! == refIdxLXA1` (com X sendo substituído por 0 e 1).

a entrada `mergeCandList[numMergCand]` é fixada como igual a A₁, e entrada `mergCandIsVspFlag[numMergCand]` é fixada como igual a `VspModeFlag[xPb - 1][yPb + nPbH - 1]` e a variável `numMergCand` é aumentada em 1.

e. Quando a variável `availableFlagB1` é igual a 1, o seguinte se aplica:

- `availableFlagT == 0 && availableFlagIvMC == 0`,

ou uma ou mais das condições seguintes são verdadeiras, com N sendo substituído por T e IvMC:

- `availableFlagT == 0 && availableFlagIvMC == 0`,
 ou uma ou mais das condições seguintes são verdadeiras com N sendo substituído por T e IvMC

- availableFlagN == 1 && predFlagLXN != predFlagLXB₁, (com X sendo substituído por 0 e 1).com

- availableFlagN == 1 && mvLXN != mvLXB₁ (com X sendo substituído por 0 e 1).

- availableFlagN == 1 && refIdxLXN != refIdxLXB₁ (com X sendo substituído por 0 e 1).

a entrada mergeCandList[numMergCand] é fixada como igual a B₁, e entrada mergeCandIsVspFlag[numMergCand] é fixada como igual a VspModeFlag[xPb - 1][yPb + nPbH - 1] e a variável numMergCand é aumentada em 1.

f. Quando availableFlagB₀ é igual a 1, a entrada mergeCandList[numMergCand] é fixada como igual a B₀, a entrada mergeCandVspFlag[numMergCand] é fixada como igual a VspModeFlag[xPb - 1][yPb + nPbH - 1] e a variável numMergCand é aumentada em 1.

g. Quando availableFlagIvDC é igual a 1, e uma ou mais das condições seguintes são verdadeiras,

- availableFlagA₁ == 0

- predFlagLXA₁ != predFlagLXIvDC, (com X sendo substituído por 0 e 1),

- mvLXA₁ != mvLXIvDC (com X sendo substituído por 0 e 1),

- availableFlagB₁ == 0,

- predFlagLXB₁ != predFlagLXIvDC, (com X sendo substituído por 0 e 1),

- mvLXB₁ != mvLXIvDC (com X sendo substituído por 0 e 1),

- refIdxLXB₁ != predFlagLXIvDC, (com X sendo substituído por 0 e 1),

a entrada mergeCandList[numMergCand] é fixada como = a IvDC, a entrada mergeCandIsVspFlag[numMergCand] é fixada como = a zero e a variável numMergCand é aumentada em 1.

h. Quando `availableFlagVSP` é igual 1, a entrada `mergeCandList[numMergCand]` é fixada como igual a VSP, a entrada

`mergeCandIsVspFlag[numMergCand]` é fixada como igual a 1 e a variável

`numMergCand` é aumentada em 1.

i. Quando `availableFlagA0` é igual a 1, a entrada `mergeCandList[numMergCand]` é fixada como `A0`, a entrada

`mergeCandIsVspFlag[numMergCand]` é fixada como igual

`VpsModelFlag[xPb - 1][yPb + nPbH]` e a variável `numMergCand` é aumentada em 1.

j. Quando `availableFlagB2` é igual 1 e `numMergCand` é menor que

$4 + \text{iv_mv_pred_flag}[\text{nuh_layer_id}] + \text{DepthFlag}$, a entrada `mergeCandList[numMergCand]` é fixada como igual a `B2`, a entrada

`mergeCandIsVspFlag[numMergCand]` é fixada como igual a

`VpsModelFlag[xPb - 1][yPb + nPbH]` e a variável `numMergCand` é aumentada em 1.

k. Quando `availableFlagIvMCShift` é igual a 1 e `numMergCand` é menor que 6, e uma ou mais das condições seguintes são verdadeiras,

- `availableFlagIvMC == 0`

- `predFlagLXMC != predFlagLXMCSHIFT` (com X sendo substituído por 0 e 1),

- `mvLXMC != mvLXMCSHIFT` (com X sendo substituído por 0 e 1),

- `refIdxLXMC != refIdxLXMCSHIFT` (com X sendo substituído por 0 e 1),

a entrada `mergeCandList[numMergCand]` é fixada como igual a 0 e a `IvMCShift`, e a entrada `mergeCandIsVspFlag[numMergCand]` é fixada como igual a 0 e a variável `numMergCand` é aumentada em 1.

1. A variável `availableFlagIvMCShift` é fixada em 0 e quando todas as condições seguintes são verdadeiras,

- `DepthFlag` é igual a 0, [`Ed(CY)`: esta condição já está assumida e, portanto, pode ser removida].
- `availableFlagIvMCShift` é igual a 0,
- `numMergCand` é menor que 6.

o processo de derivação para o candidato à fusão de disparidade deslocada especificado no sub-parágrafo H, 8.5.3.2.15 é chamado com os indicadores de disponibilidade `availableFlagN`, os índices de referência `refIdxL0N` e `refIdxL1N`, os indicadores de indicação de lista de predição `predFlagL0N` e `predFlagL1N`, os vetores de movimento `mvL0N` e `mvL1N`, de cada candidato `N` estando na `mergeCandList`, na `mergeCandList` e na `mergeCandIsVspFlag` e `numMergCand` como as entradas, e as saídas sendo o indicador `availableFlagIvMCShift`, os indicadores de utilização de predição `predFlagL0IvDCShift` e `predFlagL1IvDCShift`, os índices de referência `refIdxL0IvDCShift` e `refIdxL1IvDCShift` e os vetores de movimento `mvL0IvDCShift` e `mvL1IvDCShift`. Quando `availableFlagIvDCShift` é igual a 1, a entrada `mergeCandList[numMergCand]` é fixada como igual a `IvDCShift`, a entrada `mergeCandIsVspFlag[numMergCand]` é fixada como igual a zero, e a variável `numMergCand` é aumentada em 1.

m. Quando `availableFlagCol` é igual a 1 e `numMergCand` é menor que `5 + iv_mv_pred_flag[nuh_layer_id] + DepthFlag`, a entrada `mergeCandList[numMergCand]` é fixada como igual a `Col`, a entrada `mergeCandIsVspFlag[`

numMergCand] é fixada como igual a 0 e a variável numMergCand é aumentada em 1.

9. A variável OrigMergCand é fixada como igual a numMergCand.

10. Quando slice_type é igual a B e numMergCand é menor que 5, o processo de derivação para candidatos à fusão bi-preditiva combinada especificado no sub-parágrafo 8.5.3.2.3 [[H 8.5.3.2.3]] é chamado como mergeCandList, [[mergCandIsVspFlag]] os índices de referência refIdxLoN e refIdxLlN, os indicadores de utilização de lista de predição predFlagLoN e predFlagLlN, os vetores de movimento mvLoN e mvLlN de cada candidato N na mergeCandList, numCurrMergeCand e numOrigMergeCand como entradas, e a saída sendo atribuída a mergeCandList, numCurrMergeCand, aos índices de referência refIdxL0comCand_k e refIdxLlcomCand_k, aos indicadores de utilização de lista de predição predFlagL0comCand_k e predFlagLlcomCand_k e os vetores de movimento mvL0comCand_k e mvLlcomCand_k de cada novo candidato combCand_k sendo adicionados à mergeCandList. O número de candidatos sendo adicionado, numCombMergeCand é fixado como igual a,

$(\text{numCurrMergeCand} - \text{numOrigMergeCand})$. Quando numCombMergeCand é maior que 0, k varia de 0 a numCombMergeCand - 1, inclusive [[,e mergCandIsVspFlag[numOrigMergeCand + k] é fixado como igual a 0]].

11. O processo de derivação para zero candidatos à fusão de móvel especificado no sub-parágrafo 8.5.3.2.4 é chamado com a mergeCandList, os índices de referência refIdxLoN e refIdxLlN, os indicadores de utilização de lista de predição predFlagLoN e predFlagLlN, os vetores de movimento mvLoN e mvLlN de cada candidato N na mergeCandList, e numCurrMergeCand como entradas, e a saída sendo atribuída a mergeCandList, numCurrMergeCand, aos

índices de referência $\text{refIdxL0zeroCand}_m$ e $\text{refIdxL1zeroCand}_m$ aos indicadores de utilização de lista de predição $\text{predFlagL0zeroCand}_m$ e $\text{predFlagL1zeroCand}_m$ e os vetores de movimento mvL0zeroCand_m e mvL1zeroCand_m de cada novo candidato zeroCand_m sendo adicionado à mergeCandList . O número de candidatos que são adicionados, numZeroMergeCand , é fixado como sendo igual a

$(\text{numCurrMergeCand} - \text{numOrigMergeCand} - \text{numCombMergeCand})$. Quando numZeroMergeCand é maior que 0, m varia de 0 a $\text{numZeroMergeCand} - 1$, inclusive e $\text{mergeCandIsVspFlag}[\text{numOrigMergeCand} + \text{numCombMergeCand} + m]$ é igual a 0.

...

[H. 8.5.3.2. Processo de derivação para candidatos à fusão bi-preditiva combinada

As entradas para este processo são:

- uma lista de candidatos à fusão mergeCandList ,
- uma lista $\text{mergeCandIsVspFlag}$,
- os índices de referência refIdxL0N e refIdxL1N de cada candidato N em mergeCandList ,
- os indicadores de utilização de lista de predição predFlagL0N e predFlagL1N de cada candidato N em mergeCandList ,
- os vetores de movimento mvL0N e mvL1N de cada candidato N em mergeCandList ,
- o número de elementos numCurrMergeCand dentro de mergeCandList ,
- o número de elementos numOrigMergeCand dentro da mergeCandList depois do processo de derivação de candidatos à fusão espaciais e temporais.
- as saídas deste processo são:
- a lista de candidatos à fusão mergeCandList ,

- o número de elementos `numCurrMergeCand` dentro de `mergeCandList`,

- os índices de referência `refIdxL0comCandk` e `refIdxL1comCandk` de cada novo candidato `combCandk` adicionado à `mergeCandList` durante a chamada deste processo,

- os indicadores de utilização de lista de predição `predFlagL0combCandk` e `predFlagL1combCandk` de cada novo candidato `combCandk` adicionado a `mergeCandList` durante a chamada deste processo,

- os vetores de movimento `mcL0combCandk` e `mvL1comCandk` de cada novo candidato `combCandk` adicionado à `mergeCandList` durante a chamada deste processo.

Quando `numOrigMergeCand` é maior que 1, e menor que `MaxNumMergeCand`, a variável `numInputMergeCand` é fixada como igual a `numCurrMergeCand`, a variável `combIdx` é fixada como igual a 0, a variável `combStop` é fixada em FALSA e as etapas seguintes são repetidas até que `combStop` seja igual a VERDADEIRA:

1. As variáveis `l0CandIdx` e `l1CandIdx` são derivadas utilizando-se `combIdx` conforme especificado na Tabela 8-6.

2. As atribuições seguintes são feitas, `l0Cand` sendo o candidato na posição `l0CandIdx` e `l1Cand` sendo o candidato na posição `l1CandIdx` na lista de candidatos à fusão `mergeCandList`:

- `l0Cand = mergeCandList[l0CandIdx]`

- `l1Cand = mergeCandList[l1CandIdx]`

3. Quando todas as condições seguintes forem verdadeiras:

- $mergeCandIsVspFlag[l0CandIdx] == 0$,
- $mergeCandIsVspFlag[l1CandIdx] == 0$,
- $predFlagL0l0Cand == 1$
- $predFlagL1l1Cand == 1$
- $(DiffPicOrderCnt(RefPicList0[refIdxL0l0Cand], RefPicList1[refIdxL1l1Cand]) \neq 0) \parallel$
 $(mvL0l0Cand \neq mvL1l1Cand)$

o candidato $combCand_k$ com k igual a $numCurrMergeCand - numInputMergeCand$ é adicionado no final da $mergeCandList$, isto é, $mergeCandList[numCurrMergeCand]$ é fixado como igual a $combCand_k$ e os índices de referência, os indicadores de utilização de lista de predição e os vetores de movimento de $combCand_k$ são derivados da maneira seguinte e $numCurrMergeCand$ é incrementado em

$$refIdxL0combCand_k = refIdxL0l0Cand \quad (H-101)$$

$$refIdxL1combCand_k = refIdxL1l1Cand \quad (H-102)$$

$$predFlagL0combCand_k = 1 \quad (H-103)$$

$$predFlagL1combCand_k = 1 \quad (H-104)$$

$$mvL0combCand_k[0] = mvL0l0Cand[0] \quad (H-105)$$

$$mvL0combCand_k[1] = mvL0l0Cand[1] \quad (H-106)$$

$$mvL1combCand_k[0] = mvL1l1Cand[0] \quad (H-107)$$

$$mvL1combCand_k[1] = mvL1l1Cand[1] \quad (H-108)$$

$$1: numCurrMergeCand = numCurrMergeCand + 1 \quad (H-109)$$

4. A variável com Idx é incrementada em 1.

5. Quando comIdx é igual a (numOrigMergeCand*(numOrigMergeCand - 1)) ou numCurrMergeCand é igual a MaxNumMergeCand, combStop é fixado igual a VERDADEIRO.]]

[0137] Conforme mostrado acima, "mergCandIsVspFlag" é qualquer arranjo de indicadores de BVSP definidos na seção H. 8.5.3.2.1 do Texto de Rascunho 3D-HEVC 1. Cada valor no arranjo "mergCandIsVspFlag" corresponde a um candidato à fusão na lista e indica se o candidato à fusão correspondente é baseado na BVSP. Na etapa 10 da sessão H. 8.5.3.2.1, "mergCandIsVspFlag" é apagado, de modo que "mergCandIsVspFlag" não seja apresentado como uma entrada do processo de derivação para candidatos a fusão bi-preditiva combinada. Além disto, de acordo com uma ou mais técnicas desta revelação, a sessão H. 8.5.3.2.3 é suprimida do Texto de Rascunho 3D-HEVC uma vez que o processo de derivação para candidatos à fusão bi-preditiva combinada é idêntico na 3D-HEVC ao definido na HEVC (isto é, seção 8.5.3.2.3 do Rascunho Operacional HEVC 10). Além disto, de acordo com uma ou mais técnicas desta revelação, conforme mostrado no texto acima, mergCandIsVspFlag[numOrigMergeCand + k] não é fixado como igual a 0 uma vez que já não é necessário fazê-lo.

[0138] A Figura 8 é um diagrama de blocos que mostra um codificador de vídeo 20 exemplar que pode implementar as técnicas desta revelação. A Figura 8 é apresentada para fins de explicação e não dever ser considerada como limitando as técnicas amplamente exemplificadas e descritas nesta revelação. Para fins de explicação, esta revelação descreve o codificador de vídeo 20 no contexto da codificação HEVC. Entretanto, as técnicas

desta revelação podem ser aplicadas a outros padrões ou métodos de codificação.

[0139] No exemplo da Figura 8, o codificador de vídeo 20 inclui uma unidade de processamento de predição 100, uma memória de dados de vídeo 101, uma unidade de geração de resíduos 102, uma unidade de processamento de transformadas 104, uma unidade de quantificação 106, uma unidade de quantificação inversa 108, uma unidade de processamento de transformada inversa 110, uma unidade de reconstrução 112, uma unidade de filtro 114, um buffer de imagens decodificadas 116 e uma unidade de codificação por entropia 118. A unidade de processamento de predição 100 inclui uma unidade de processamento de inter-predição 120 e uma unidade de processamento de intra-predição 126. A unidade de processamento de inter-predição 120 inclui uma unidade de estimação de movimento 122 e uma unidade de compensação de movimento 124. Em outros exemplos, o codificador de vídeo 20 pode incluir mais, menos ou diferentes componentes funcionais.

[0140] O codificador de vídeo 20 pode receber dados de vídeo. A memória de dados de vídeo 101 pode armazenar dados de vídeo a serem codificados pelos componentes do codificador de vídeo 20. Os dados de vídeo armazenados na memória de dados de vídeo 101 pode ser obtidos, por exemplo, da fonte de vídeo 18. O armazenador de imagens decodificadas 116 pode ser uma memória de índices de referência que armazena dados de vídeo de referência para utilização na condição de dados de vídeo pelo codificador de vídeo 20, como, por exemplo, nos modos de intra- ou inter-codificação. A memória de dados de vídeo 101 e/ou o armazenador de imagens decodificadas 116 podem ser formados por qualquer um de diversos aparelhos de memória, tais como memória acesso aleatório dinâmica

(DRAM), que inclui DRAM síncrona (SDRAM), RAM magneto-resistiva (MRAM), RAM resistiva (RRAM), ou outros tipos de aparelhos de memória. A memória de dados de vídeo 101 e o armazenador de imagens decodificadas 116 podem ser fornecidos pelo mesmo aparelho de memória ou por aparelhos de memória separados. Em diversos exemplos, a memória de dados de vídeo 101 pode ficar embutida com outros componentes do codificador de vídeo 20 ou não embutida com relação a esses componentes.

[0141] O codificador de vídeo 20 pode codificar cada CTU em uma fatia de uma imagem dos dados de vídeo. Cada uma das CTUs pode estar associada a blocos de árvore de codificação (CTBs) luma de igual tamanho e com CTBs correspondentes da imagem. Como parte da codificação de uma CTU, a unidade de processamento de predição 100 pode efetuar particionamento de transformação quad-tree de modo a dividir os CTBs da CU em blocos progressivamente menores. Os blocos menores podem ser blocos de codificação de CUs. Por exemplo, a unidade de processamento de predição 100 pode particionar um CTB associado a uma CU em quatro sub-blocos de igual tamanho, particionar um ou mais dos sub-blocos em quatro sub-blocos de igual tamanho, e assim por diante.

[0142] O codificador de vídeo 20 pode codificar CUs de uma CTU de modo a gerar representações codificadas das CUs (isto é, CUs codificadas). Como parte de uma codificação de uma CU, a unidade de processamento de predição 100 pode particionar os blocos de codificação associados à CU entre uma ou mais PUs da CU. Assim, em alguns exemplos cada PU pode estar associada a um bloco de predição luma e a blocos de predição croma correspondentes. O codificador de vídeo 20 e o decodificador de vídeo 30 podem suportar PUs com diversos tamanhos. Conforme indicado

acima, o tamanho de uma CU pode referir-se ao tamanho do bloco de codificação luma da CU e o tamanho de uma PU pode referir-se ao tamanho de um bloco de predição luma da PU. Supondo-se que o tamanho de uma CU específica seja $2N \times 2N$, o codificador de vídeo 20 e o decodificador de vídeo 30 podem suportar tamanhos de PU de $2N \times 2N$ ou $N \times N$, para inter-predição e tamanhos de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$ ou $N \times N$ ou semelhante para inter-predição. O codificador de vídeo 20 e o decodificador de vídeo 30 podem suportar também particionamento assimétrico para tamanhos de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ e $nR \times 2N$ para inter-predição.

[0143] A unidade de predição de inter-predição 120 pode gerar dados preditivos para uma PU efetuando inter-predição em cada PU de uma CU. Os dados preditivos para a PU podem incluir blocos preditivos da PU e informações de movimento para a PU. A unidade de processamento de inter-predição 120 pode executar operações diferentes para uma PU de uma CU dependendo de estar a PU em uma fatia I, uma fatia P ou uma fatia B. Em uma fatia I, todas as PUs podem ser intra-preditadas. Consequentemente, se a PU estiver em uma fatia I, a unidade de predição de inter-predição 120 não efetua inter-predição na PU. Assim, para blocos codificados no modo I, o bloco predito é formado utilizando-se predição espacial a partir de blocos vizinhos codificados anteriormente dentro do mesmo quadro.

[0144] Se uma PU estiver em uma fatia P, a unidade de estimação de movimento 122 pode buscar as imagens de referência em uma lista de imagens de referência ("RefPicList0", por exemplo) para uma região de referência para a PU. A região de referência para a PU pode ser uma região, dentro de uma imagem de referência, que contém blocos de amostra que correspondem mais intimamente aos blocos de amostra de PU. A unidade de estimação de móvel

122 pode gerar um índice de referência que indica uma posição na RefPicList0 da imagem de referência que contém a região de referência para a PU. Além disto, a unidade de estimação de movimento 122 pode gerar um vetor de movimento que indica um deslocamento especial entre o bloco de codificação da PU e a localização de referência associada à região de referência. Por exemplo, o vetor de movimento pode ser um vetor bidimensional que aparelha um deslocamento das coordenadas na imagem central com relação à coordenadas em uma imagem de referência. A unidade de estimação de movimento 122 pode transmitir o índice de referência e o vetor de movimento como as informações de movimento da PU. A unidade de processamento 124 pode gerar os blocos preditivos (isto é, blocos preditivos) da PU com base em amostras reais ou interpoladas associadas à localização de referência indicada pelo vetor de movimento da PU.

[0145] Se uma PU estiver em uma fatia B, a unidade de estimação de movimento 122 pode efetuar uni-predição ou bi-predição para a PU. Para efetuar uni-predição para a PU, a unidade de estimação de movimento 122 pode buscar as imagens de referência da RefPicList0 ou de uma segunda lista de imagens de referência ("RefPicList1") para uma região de referência para a PU. A unidade de estimação de movimento 122 pode transmitir, como a informação de movimento da PU, um indicador que indica uma posição na RefPicList0 ou RefPicList1 da imagem de referência que contém a região de referência. Um vetor de movimento que indica um deslocamento especial entre um bloco de amostra da PU e a localização de referência associada à região de referência e um ou mais indicadores de direção de predição que indicam se a imagem de referência está na RefPicList0 ou na RefPicList1. A unidade

de processamento 124 pode gerar os blocos preditivos da PU, com base, pelo menos em parte, em amostras reais ou interpoladas associadas à localização de referência indicada pelo vetor de movimento da PU.

[0146] Para efetuar inter-predição bidirecional para uma PU, a unidade de estimação de movimento 122 pode buscar as imagens de referência na RefPicList0 para uma região de referência para a PU que pode buscar também as imagens de referência na RefPicList1 para outra região de referência para a PU, a unidade de estimação de movimento 122 pode gerar índices de referência que indicam as posições na RefPicList0 e RefPicList1 das imagens de referência que contêm as regiões de referência. Além disto, a unidade de estimação de movimento 122 pode gerar vetores de movimento que indicam deslocamento espaciais entre as localizações de referência associadas às regiões de referência e um bloco de predição (um bloco de amostra, por exemplo) da PU. As informações de movimento da PU podem incluir os índices de referência e os vetores de movimento da PU. A unidade de compensação de movimento 124 pode gerar os blocos preditivos da PU com base, pelo menos em parte, em amostras reais ou interpoladas associadas à região de referência indicada pelos vetores de movimento da PU.

[0147] De acordo com uma ou mais técnicas desta revelação, a unidade de estimação de movimento 122 pode gerar uma lista de candidatos à fusão para codificar um bloco de vídeo de vídeo 3D. Como parte da geração da lista de candidatos à fusão, a unidade de estimação de movimento 122 pode determinar se o número de candidatos à fusão na lista de candidatos à fusão é menor que 5. Em resposta à determinação de que o número de candidatos à fusão na lista de candidatos à fusão é menor que 5, a

unidade de estimação de movimento 122 pode derivar um ou mais candidatos à fusão bi-preditiva combinada. A unidade de estimação de movimento 122 podem incluir candidato ou candidatos à fusão bi-preditiva combinada na lista de candidatos à fusão. Além disto, em alguns exemplos, a unidade de estimação de movimento 122 pode selecionar um candidato à fusão na lista de candidatos à fusão. O codificador de vídeo 20 pode sinalizar a posição na lista de candidatos à fusão do candidato à fusão selecionado. Em alguns exemplos, o número máximo de candidatos à fusão na lista de candidatos à fusão é igual ou maior que 5 (6, por exemplo).

[0148] Referência contínua é agora feita ao exemplo da Figura 8. A unidade de processamento de intra-predição 126 pode gerar dados preditivos para uma PU efetuando intra-predição na PU. Os dados preditivos para a PU podem incluir blocos preditivos para a PU e diversos elementos de sintaxe. A unidade de predição de intra-predição 126 pode efetuar intra-predição em PUs em fatias I, fatias P e fatias B.

[0149] Para efetuar intra-predição em uma PU, a unidade de processamento de intra-predição 126 pode utilizar vários modos de intra-predição para gerar vários conjuntos de dados preditivos para a PU. Para utilizar alguns modos de intra-predição para gerar um conjunto de dados preditivos para a PU, a unidade de processamento de intra-predição 126 pode estender amostras de blocos vizinhos através do bloco preditivo da PU em uma direção associada ao modo de intra-predição. As PUs vizinhas podem estar acima e à direita, acima e à esquerda ou à esquerda da PU, supondo-se uma ordem de codificação da esquerda para a direita, do topo para a base, para as PUs, CUs e CTUs. A unidade de processamento de intra-predição 126 pode

utilizar diversos números de modos de intra-predição, como, por exemplo, trinta e três modos de adicional direcional. Em alguns exemplos, o número de modos de intra-predição pode depender do tamanho da região associada à PU.

[0150] A unidade de processamento de predição 100 pode selecionar os dados preditivos para PUs de uma CU dentre os dados preditivos gerados pela unidade de processamento de inter-predição 120 para a PU ou dentre os dados preditivos gerados pela unidade de processamento de intra-predição 126 para as PUs. Em alguns exemplos, a unidade de processamento de predição 100 seleciona os dados preditivos para as PUs da CU com base em métricas de taxa/distorção dos conjuntos de dados preditivos. Os blocos preditivos dos dados preditivos selecionados podem ser aqui referidos como blocos preditivos selecionados.

[0151] A unidade de geração de resíduos 102 pode gerar, com base nos blocos de codificação (blocos de codificação Cb, Cr, por exemplo) de uma CU e nos blocos preditivos selecionados (luma preditivo, blocos Cb e Cr, por exemplo) das PUs da CU, blocos residuais (luma residual, blocos residuais Cb e Cr, por exemplo) da CU. Em outras palavras, a unidade de geração de resíduos 102 pode gerar um sinal residual para a CU. Por exemplo, a unidade de geração de resíduos 102 pode gerar os blocos residuais da CU de modo que cada amostra dos blocos residuais tenha um valor igual à diferenças entre uma amostra em um bloco de codificação da CU e uma amostra correspondente em um bloco preditivo selecionado correspondente de uma PU da CU.

[0152] A unidade de processamento de transformadas pode efetuar particionamento de transformação quad-tree para particionar os blocos residuais associados a uma CU em blocos de transformadas que correspondem a (isto é, associados a) TUs da CU. Assim, uma TU pode estar

associada a um bloco de transformadas luma e dois blocos de transformadas croma. Os tamanhos e posições dos blocos de transformadas (blocos de transformadas luma e croma, por exemplo) de TUs de uma CU podem ou podem não ser baseados nos tamanhos e posições dos blocos de predição das PUs da CU. Uma enquanto conhecida como "transformação quad-tree residual" (RQT) pode incluir nós associados a cada uma das TUs. As TUs de uma CU podem corresponder a nós-folha da RQT.

[0153] A unidade de processamento de transformadas 104, pode gerar blocos de coeficientes de transformada para cada TU de uma CU pela aplicação de uma ou mais transformadas aos blocos de transformadas da TU. A unidade de processamento de transformadas 104 pode aplicar diversas transformadas a um bloco de transformadas associado a uma TU. Por exemplo, a unidade de processamento de transformadas 104 pode aplicar uma transformada de cosseno discreta (DCT), uma transformada direcional ou uma transformada conceitualmente semelhante a um bloco de transformadas. Em alguns exemplos, a unidade de processamento de transformadas 104 não aplica transformadas a um bloco de transformadas. Em tais exemplos, o bloco de transformadas pode ser tratado como um bloco de coeficientes de transformada.

[0154] A unidade de quantificação 106 pode quantificar os coeficientes de transformada em um bloco de coeficientes de transformada. O processo de quantificação pode reduzir a profundidade de bit associada a alguns ou todos os coeficientes de transformada de um bloco de coeficientes de transformada. Por exemplo, um coeficiente de transformada n -bit pode ser arredondado para um coeficiente de transformada de m -bits durante a quantificação, onde n é maior do que m . A unidade de

quantificação 106 pode quantificar. Um bloco de coeficientes de transformada associado a uma TU de uma CU com base em um valor de parâmetro de quantificação (QP) associado à CU. O codificador de vídeo 20 pode ajustar o grau de quantificação aplicado aos blocos de coeficientes de transformadas associados a uma CU pelo ajuste do valor de QP associado à CU. A quantificação pode introduzir perda de informação, portanto os coeficientes de transformadas quantificados podem ter precisão mais baixa que os originais.

[0155] A unidade de quantificação inversa 108 e a unidade de processamento de transformadas inversa 110 podem aplicar, quantificação inversa e transformadas inversas a um bloco de coeficientes de transformada, respectivamente, para reconstruir um bloco residual (isto é, um bloco de transformadas) a partir do bloco de coeficientes de transformadas. A unidade de reconstrução 112 pode reconstruir um bloco de codificação de uma CU, de modo que cada amostra do bloco de codificação seja igual à soma de uma amostra de um bloco preditivo de uma PU da CU e uma amostra correspondente de um bloco de transformadas de uma TU da CU. Por exemplo, a unidade de reconstrução 112 pode adicionar blocos residuais reconstruídos de TUs de uma CU a amostras correspondentes de um ou mais blocos preditivos de PUs da CU geradas pela unidade de processamento de predição 100, de modo a se produzir um bloco de codificação reconstruído da CU. Assim, pela reconstrução de blocos de transformadas para cada TU de uma CU desta maneira, o codificador de vídeo 20 pode reconstruir os blocos de codificação da CU.

[0156] A unidade de filtro 114 pode executar uma ou mais operações de desbloqueio para reduzir os artefatos de blocagem nos blocos de codificação associados

a uma CU. O armazenador de imagens decodificadas 116 pode armazenar os blocos de codificação reconstruídos depois que a unidade de filtro 114 executa a operação ou operações de desbloqueio nos blocos de codificação reconstruídos. Assim, o armazenador de imagens decodificadas 116 pode ser uma memória configurada para armazenar dados de vídeo. A unidade de processamento de inter-predição 120 pode utilizar uma imagem de referência que contém os blocos de codificação reconstruídos para efetuar inter-predição em PUs de outras imagens. Além disto, a unidade de predição de intra-predição 126 pode utilizar blocos de codificação reconstruídos no armazenador de imagens decodificadas 116 para efetuar intra-predição em outras PUs na mesma imagem da CU.

[0157] A unidade de codificação por entropia 118 pode receber dados de outros componentes funcionais do codificador de vídeo 20. Por exemplo, a unidade de codificação por entropia 118 pode receber blocos de coeficientes da unidade de quantificação 106 e pode receber elementos de sintaxe da unidade de processamento de predição 100. A unidade de codificação por entropia 118 pode executar um ou mais operações de codificação por entropia nos dados de modo a gerar dados codificados por entropia. Por exemplo, a unidade de codificação por entropia 118, pode executar uma operação de CABAC, uma operação de codificação de comprimento variável adaptativa ao contexto (CAVLC), uma operação de codificação de comprimento de variável para variável (V2V) operação de codificação aritmética binária adaptativa ao contexto baseada em sintaxe (SBAC), uma operação de codificação por Entropia com Particionamento de Intervalos de Probabilidade (PIPE), uma operação de codificação Exponencial-Golomb, ou outro tipo de operação de codificação por entropia nos

dados. O codificador de vídeo 20 pode transmitir um fluxo de bits que inclui dados codificados por entropia gerados pela unidade de codificação por entropia 118. Por exemplo, um fluxo de bits pode incluir dados que representam uma RQT para uma CU. O fluxo de bits pode incluir dados que não são codificados por entropia.

[0158] A Figura 9 é um diagrama de blocos que mostra um decodificador de vídeo 30 exemplar que é configurado para implementar as técnicas desta revelação. A Figura 9 é apresentada para fins de explicação e não limita as técnicas amplamente exemplificadas e descritas nesta revelação. Para fins de explicação, esta revelação descreve o decodificador de vídeo 30 no contexto da codificação HEVC. Entretanto, as técnicas desta revelação podem ser aplicadas a outros padrões ou métodos de codificação.

[0159] No exemplo da Figura 9, o decodificador de vídeo 30 inclui uma unidade de decodificação por entropia 150, uma memória de dados de vídeo 151, uma unidade de processamento de predição 152, uma unidade de quantificação inversa 154, uma unidade de processamento de transformada inversa 156, uma unidade de reconstrução 158, uma unidade de filtro 160 e um armazenador de imagens decodificadas 162. A unidade de processamento de predição 152 inclui uma unidade de processamento 164 e uma unidade de processamento de intra-predição 166. Em outros exemplos, o decodificador de vídeo 30 pode incluir mais, menos ou diferentes componentes funcionais.

[0160] O decodificador de vídeo 30 pode receber um fluxo de bits. A memória de dados de vídeo 151 pode armazenar dados de vídeo, tais como um fluxo de bits de vídeo codificado a serem codificados pelos componentes do decodificador de vídeo 30. Os dados de vídeo armazenados na memória de dados de vídeo 151, podem ser obtidos, por

exemplo, do canal 16, como, por exemplo, de uma fonte de vídeo local, tal como uma câmera, por meio da comunicação em rede cabeada ou sem fio de dados de vídeo, ou acessando-se meios de armazenamento de dados físicos. A memória de dados de vídeo 151 pode formar um armazenador de imagens codificadas CPB que armazena dados de vídeo codificados de um fluxo de bits de vídeo codificado. O armazenador de imagens decodificadas 162 pode ser uma memória de imagens de referência que armazena dados de vídeo de referência para utilização na decodificação de dados de vídeo pelo decodificador de vídeo 30 nos modos de intra- ou inter-condição, por exemplo. A memória de dados de vídeo 151 e o armazenador de imagens decodificadas 162 podem ser formados por qualquer um de diversos aparelhos de memória, tais como memória acesso aleatório dinâmica (DRAM), que inclui DRAM síncrona (SDRAM), RAM magneto-resistiva (MRAM), RAM resistiva (RRAM), ou outros tipos de aparelhos de memória. A memória de dados de vídeo 151 e o armazenador de imagens decodificadas 162 podem ser fornecidos pelo mesmo aparelho de memória ou por aparelhos de memória separados. Em diversos exemplos, a memória de dados de vídeo 101 pode ficar embutida com outros componentes do codificador de vídeo 20 ou não embutida com relação a esses componentes.

[0161] A unidade de decodificação por entropia 150 pode efetuar parse no fluxo de bits de modo a decodificar elementos de sintaxe do fluxo de bits. A unidade de decodificação por entropia 150 pode decodificar por entropia os elementos de sintaxe codificados por entropia no fluxo de bits. A unidade de processamento de predição 152, a unidade de quantificação inversa 154, a unidade de processamento de transformada inversa 156, a unidade de reconstrução 158 e a unidade de filtro 160 podem gerar dados de vídeo decodificados com base nos elementos

de sintaxe obtidos (extraídos, por exemplo) do fluxo de bits.

[0162] O fluxo de bits pode compreender uma série de unidades NAL. As unidades NAL do fluxo de bits podem incluir unidades NAL de fatia codificadas. Como parte da decodificação do fluxo de bits, a unidade de decodificação por entropia 150 pode obter (extrair, por exemplo) e decodificar por entropia elementos de sintaxe das unidades NAL de fatia codificadas. Cada uma das fatias codificadas pode incluir um cabeçalho de fatia e dados de fatia. O cabeçalho de fatia pode conter elementos de sintaxe referentes a uma fatia. Os elementos de sintaxe no cabeçalho de fatia podem incluir um elemento de sintaxe que identifica um PPS associado a uma imagem que contém a fatia.

[0163] Além de obter (decodificar, por exemplo) elementos de sintaxe do fluxo de bits, o decodificador de vídeo 30 pode executar uma operação de reconstrução em CUs. Para executar a operação de reconstrução em uma CU (uma CU não particionada, por exemplo), o decodificador de vídeo 30 pode executar uma operação de reconstrução em cada TU da CU. Executando a operação de reconstrução para cada TU da CU, o decodificador de vídeo 30 pode reconstruir blocos residuais (isto é, blocos de transformadas) das TU da CU.

[0164] Como parte da execução de uma operação de reconstrução em uma TU de uma CU, a unidade de quantificação inversa 154 pode quantificar por inversão, isto é, desquantificar, os blocos de coeficientes da (isto é, associados à) TU. A unidade de quantificação inversa 154 pode utilizar um valor de QP associado à CU da TU de modo a determinar o grau de quantificação e, da mesma maneira, o grau de quantificação inversa para unidade de quantificação

inversa 154 aplicar. Ou seja, a razão de compactação, isto é, a razão do número de bits utilizados para representar a sequência original e o número compactado, pode ser controlado pelo ajuste do valor do que P utilizado quando se quantificam coeficientes de transformada. A razão de compactação pode depender também do método de codificação por entropia utilizado.

[0165] Depois que a unidade de quantificação inversa 154 quantifica por inversão um bloco de coeficientes, a unidade de processamento de transformada inversa 156 pode aplicar uma ou mais transformadas inversas ao bloco de coeficientes de modo a gerar um bloco rádio-sinal associado à TU. Por exemplo, a unidade de processamento de transformada inversa pode aplicar uma DCT inversa, uma transformada de número inteiro inversa, uma transformada de Karhunen-Loeve (KLT) inversa, uma transformada rotacional inversa, uma transformada direcional inversa ou outra transformada inversa ao bloco de coeficientes.

[0166] Se uma PU for codificada utilizando-se intra-predição, a unidade de processamento de intra-predição 166 pode efetuar intra-predição de modo a gerar blocos preditivos para a PU. Por exemplo, a unidade de processamento de intra-predição 166 pode utilizar o modo de intra-predição para gerar o luma preditivo, blocos Cb e Cr para a PU com base nos blocos de predição de PUs espacialmente vizinhos. A unidade de intra-predição 166 pode determinar o modo de intra-predição para a PU com base em um ou mais elementos de sintaxe decodificados a partir do fluxo de bits.

[0167] A unidade de processamento de predição 162 pode construir uma primeira lista de imagens de referência (RefPicList0) e uma segunda lista de imagens de

referência (RefPicList1), com base em elementos de sintaxe obtidos do fluxo de bits. Além disto, se uma PU for codificada utilizando-se intra-predição, a unidade de decodificação por entropia 150 pode determinar (extrair, por exemplo) informações de movimento para a PU. A unidade de compensação de movimento 164 pode determinar, com base nas informações de movimento da PU, um ou mais blocos de referência para a PU. A unidade de compensação de movimento 164 pode gerar, com base em blocos de amostras no bloco ou blocos de referência para a PU, blocos preditivos (uma preditivo, blocos Cb e Cr, por exemplo) para a PU.

[0168] Conforme indicado acima, o codificador de vídeo 20 pode sinalizar as informações de movimento de uma PU utilizando o modo de fusão, o modo por saltos ou o modo AMVP. Quando o codificador de vídeo 20 sinaliza as informações de movimento da PU atual utilizando o modo AMVP, a unidade de decodificação por entropia 150 pode decodificar, a partir do fluxo de bits, um índice de referência, um MVD para a PU atual e o índice de candidato. Além disto, a unidade de compensação de movimento 164 pode gerar uma lista de candidatos a AMVP para a PU atual. A lista de candidatos AMVP inclui um ou mais candidatos a preditor de vetor de movimento. Cada um dos candidatos a vetor de movimento especifica um vetor de movimento de uma PU que é vizinha especial ou temporalmente da PU atual. A unidade de compensação de movimento 164 pode determinar, com base, pelo menos em parte, no índice de candidato, um candidato a preditor de vetor de movimento selecionado na lista de candidatos a AMVP. A unidade de compensação de movimento 164 pode determinar então o vetor de movimento da PU atual adicionando o MVD ao vetor de movimento especificado pelo candidato a preditor de vetor de movimento selecionado. Em outras palavras, para a AMVP, o

vetor de movimento é calculado como vetor de movimento (MV) = MVP + MVD, onde o índice do preditor de vetor de movimento (MVP) é sinalizado e o MVP é um dos candidatos a vetor de movimento (espaciais ou temporais) da lista AMVP, e o MVD é sinalizado para o lado do decodificador.

[0169] Se a PU atual for bi-predita e as informações de movimento da PU forem sinalizadas no modo AMVP, a unidade de decodificação por entropia 150 pode decodificar um índice de referência adicional, MVD, e o índice de candidato a partir do fluxo de bits. A unidade de compensação de movimento 162 pode repetir o processo descrito acima utilizando-se o índice de referência adicional, MVD, e o índice de candidato de modo a derivar um segundo vetor de movimento para a PU atual. Desta maneira, a unidade de compensação de movimento 162 pode derivar o vetor de movimento para RefPicList0 (isto é, um vetor de movimento RefPicList0) e um vetor de movimento para RefPicList1 (isto é, um vetor de movimento RefPicList1).

[0170] De acordo com uma ou mais técnicas desta revelação, a unidade de compensação de movimento 164 pode gerar uma lista de candidatos à fusão para codificar um bloco de vídeo 3D. Como parte da geração da lista de candidatos à fusão, a unidade de compensação de movimento 164 pode determinar se o número de candidatos a fusão na lista de candidatos à fusão é menor que 5. Em resposta à determinação de que o número de candidatos a fusão na lista de candidatos à fusão é menor que 5, a unidade de compensação de movimento 164 pode derivar um ou mais candidatos à fusão bi-preditiva combinada. A unidade de compensação de movimento 164 pode incluir o candidato ou candidatos à fusão bi-preditiva combinada na lista de candidatos à fusão. Além disto, em alguns exemplos, o

decodificador de vídeo 30 pode obter, de um fluxo de bits, um elemento de sintaxe que indica um candidato à fusão selecionado na lista de candidatos à fusão. A unidade de compensação de movimento 164 pode utilizar informações de movimento do candidato selecionado para gerar amostras preditivas da PU atual. Em alguns exemplos o número máximo de candidatos à fusão na lista de candidatos à fusão é igual ou maior que 5 (6, por exemplo).

[0171] Continuando, é feita agora referência à Figura 9. A unidade de reconstrução 158 pode utilizar os blocos de transformadas (luma, blocos de transformadas Cb e Cr, por exemplo) de TUs de uma CU e os blocos preditivos (luma preditivo, blocos Cb e Cr, por exemplo) das PUs da CU, isto é, ou dados de intra-predição ou dados de inter-predição, conforme aplicável, para reconstruir os blocos de condição (luma, blocos de codificação Cb e Cr, por exemplo) da CU. Por exemplo, a unidade de reconstrução 158 pode adicionar amostras dos blocos de transformadas (luma, blocos de transformadas Cb e Cr, por exemplo) a amostras correspondentes dos blocos preditivos (luma preditivo, blocos Cb e Cr, por exemplo) para reconstruir os blocos de codificação (luma, blocos de codificação Cb e Cr, por exemplo) da CU.

[0172] A unidade de filtro 160 pode executar uma operação de desbloqueio para reduzir os artefatos de blocagem associados ao blocos de codificação (luma, blocos de codificação Cb e Cr, por exemplo) da CU. O decodificador de vídeo 30 pode armazenar os blocos de codificação (luma, blocos de codificação Cb e Cr, por exemplo) da CU no armazenador de imagens decodificadas 162. O armazenador de imagens decodificadas 162 pode fornecer imagens de referência para compensação de movimento, intra-predição e apresentação subsequentes em um aparelho de exibição, tal

como o aparelho de exibição 32 da Figura 1. Por exemplo, o decodificador de vídeo 30 pode executar, com base nos blocos (luma, blocos de codificação Cb e Cr, por exemplo) no armazenador de imagens decodificadas 162, operações de intra-predição ou inter-predição em PUs de outras CUs. Desta maneira, o decodificador de vídeo 30 pode obter, do fluxo de bits, níveis de coeficientes de transformada do bloco de coeficientes luma significativo, quantificar por inversão os níveis de coeficientes de transformada, aplicar uma transformada aos níveis de coeficiente de transformada de modo a gerar um bloco de transformada, gerar, cabeçalho pelo menos em parte no bloco de transformada, um bloco de codificação, e transmitir o bloco de codificação para exibição.

[0173] A Figura 10A é um fluxograma que mostra uma operação exemplar do codificador de vídeo 20 para codificar dados associados a vídeos 3d, de acordo com uma ou mais técnicas desta revelação. A operação da Figura 10A, juntamente com operações mostradas em outros fluxogramas desta revelação, é um exemplo. Outras operações exemplares de acordo com as técnicas desta revelação podem incluir mais, menos ou diferentes ações.

[0174] No exemplo da Figura 10A, o codificador de vídeo 20 pode gerar uma lista de candidatos à fusão (200). Em outras palavras, o codificador de vídeo 20 pode gerar uma lista de candidatos à fusão. As Figuras 11 e 12, descritas em outro lugar nesta revelação mostra uma operação exemplar para gerar a lista de candidatos à fusão. Em alguns exemplos, o codificador de vídeo 20 pode gerar a lista de candidatos à fusão da mesma maneira que o decodificador de vídeo 30. De acordo com uma ou mais técnicas desta revelação, quando o codificador de vídeo 20 gera a lista de candidatos à fusão, o codificador de vídeo

20 pode determinar se o número de candidatos à fusão na lista de candidatos à fusão é menor que 5. Em resposta à determinação de que o número de candidatos à fusão na lista de candidatos à fusão é menor que 5, o codificador de vídeo 20 pode derivar um ou mais candidatos à fusão bi-preditiva. O codificador de vídeo 20 pode incluir o candidato ou candidatos à fusão bi-preditiva na lista de candidatos à fusão. Em alguns exemplos, o número máximo de candidatos à fusão na lista de candidatos à fusão é igual a 6.

[0175] Além disto, no exemplo da Figura 10A, o codificador de vídeo 20 pode selecionar um candidato na lista de candidatos à fusão (202). Em algumas exemplos, o codificador de vídeo 20 pode sinalizar o candidato selecionado em um fluxo de bits. Por exemplo, o codificador de vídeo 20 pode incluir um elemento de sintaxe de índice de fusão no fluxo de bits. O codificador de vídeo 20 pode codificar um bloco de vídeo com base no candidato selecionado (204). Por exemplo, o bloco de vídeo pode ser uma CU. Neste exemplo, o codificador de vídeo 20 pode utilizar as informações de movimento (como, por exemplo, vetores de movimento, índices de referência, etc.) do candidato selecionado de modo a determinar um bloco preditivo para uma PU da CU. Além disto, neste exemplo, o codificador de vídeo 20 pode determinar valores de pelo menos algumas amostras de um bloco de transformadas (um bloco residual, por exemplo) com base em amostras do bloco preditivo e amostras correspondentes de um bloco de codificação da CU. Por exemplo, o codificador de vídeo 20 pode determinar valores de pelo menos algumas das amostras do bloco de transformada de modo que as amostras sejam iguais às diferenças entre as amostras do bloco preditivo e as amostras correspondentes de um bloco de codificação da CU.

[0176] A Figura 10B é um fluxograma que mostra uma operação exemplar do decodificador de vídeo 30 para decodificar dados associados a vídeo 3D, de acordo com uma ou mais técnicas desta revelação. No exemplo da Figura 10B, o decodificador de vídeo 30 pode gerar uma lista de candidatos à fusão (220), em outras palavras, o decodificador de vídeo 30 pode gerar uma lista de candidatos à fusão. As Figuras 11 e 12, descritas em outro lugar nesta revelação, mostram uma operação exemplar para gerar a lista de candidatos à fusão. Em alguns exemplos, o decodificador de vídeo 30 pode gerar a lista de candidatos à fusão da mesma maneira que o codificador de vídeo 20. De acordo com uma ou mais técnicas desta revelação, quando o decodificador de vídeo 30 gera a lista de candidatos à fusão, o decodificador de vídeo 30 pode determinar se o número de candidatos à fusão na lista de candidatos à fusão é menor que 5 e, em resposta, à determinação de que o número de candidatos à fusão na lista de candidatos à fusão é menor que 5, o decodificador de vídeo 30 pode derivar um ou mais candidatos à fusão bi-preditiva. O decodificador de vídeo 30 pode incluir o candidato ou candidatos à fusão bi-preditiva na lista de candidatos à fusão. Em alguns exemplos, o número máximo de candidatos à fusão na lista de candidatos à fusão é igual a 6

[0177] Além disto, no exemplo da Figura 10B, o decodificador de vídeo 30 pode determinar um candidato selecionado na lista de candidatos à fusão (222). Em alguns exemplos, o decodificador de vídeo 30 pode determinar o candidato selecionado com base no valor indicado por um elemento de sintaxe sinalizado em um fluxo de bits. O decodificador de vídeo 30 pode decodificar um bloco de vídeo com base no candidato selecionado (224). Por exemplo, o bloco de vídeo pode ser uma CU. Neste exemplo, o

decodificador de vídeo 30 pode utilizar as informações de movimento (como, por exemplo, vetores de movimento, índices de referência, etc.) do candidato selecionado de modo a determinar um bloco preditivo para uma PU da CU. Além disto, neste exemplo, o decodificador de vídeo 30 pode determinar valores de pelo menos algumas das amostras de um bloco de codificação da CU com base no bloco preditivo. Por exemplo, o decodificador de vídeo 30 pode determinar valores de pelo menos algumas das amostras do bloco de codificação de modo que as amostras sejam iguais à soma das amostras do bloco preditivo e das amostras correspondentes de um bloco de transformadas de uma TU da CU.

[0178] A Figura 11 é um fluxograma que mostra uma primeira parte de uma operação 300 exemplar para construir uma lista de candidatos à fusão p-ara o bloco atual, de acordo com uma ou mais técnicas desta revelação. Na operação exemplar da Figura 11, uma ou mais ações podem ser re-dispostas ou omitidas. Em outros exemplos, operações semelhantes podem incluir ações adicionais.

[0179] No exemplo da Figura 11, um codificador de vídeo (o codificador de vídeo 20 ou o decodificador de vídeo 30, por exemplo) pode determinar um IPMVC (302). Em alguns exemplos, o codificador de vídeo pode determinar o IPMVC utilizando um vetor de disparidade para o bloco atual de modo a identificar um bloco correspondente em uma imagem de referência inter-vista. Em tais exemplos, se o bloco correspondente não for intra-predito e não predito inter-vista e tiver um vetor de movimento temporal (isto é, um vetor de movimento que indica a localização em uma imagem de referência associada a uma ocorrência de tempo da do bloco correspondente), o IPMVC pode especificar os vetores de movimento do bloco correspondente, indicadores de direção de predição do bloco correspondente em índices de

referência convertidos do bloco correspondente. Em seguida, o codificador de vídeo pode determinar se há IPMVC está disponível (304). Em alguns exemplos, o IPMVC está indisponível se o bloco correspondente na imagem de referência inter-vista for intra-predito ou estiver fora das fronteiras da imagem de referência inter-vista. Em resposta à determinação que o IPMVC está disponível ("SIM" de 304), o codificador de vídeo pode inserir o IPMVC na lista de candidatos à fusão (3060

[0180] Depois de inserir o IPMVC na lista de candidatos à fusão ou em resposta à determinação que o IPMVC não está disponível ("NÃO"), o codificador de vídeo pode verificar os PUs vizinhos espaciais de modo a determinar se as PUs vizinhas espaciais têm vetores de movimento disponíveis (308). Em alguns exemplos, as PUs vizinhas espaciais cobrem as localizações indicadas como A_0 , A_1 , B_0 , B_1 e B_2 na Figura 2. Para facilitar a explicação, esta revelação pode referir-se as informações de movimento de PUs que cobrem as localizações A_0 , A_1 , B_0 , B_1 e B_2 como A_0 , A_1 , B_0 , B_1 e B_2 , respectivamente.

[0181] No exemplo da Figura 11, o codificador de vídeo pode determinar se A_1 corresponde ao IPMVC (310). Em resposta à determinação de que A_1 não corresponde ao IPMVC ("NÃO" de 310), o codificador de vídeo pode inserir A_1 na lista de candidatos à fusão (312). Caso contrário, em resposta à determinação de que A_1 corresponde ao IPMVC ("SIM" de 310) ou depois de inserir A_1 na lista de candidatos à fusão, o codificador de vídeo pode determinar se B_1 corresponde a A_1 ou ao IPMVC (314). Em resposta à determinação de que B_1 não corresponde a A_1 ou ao IPMVC ("NÃO" de 314), o codificador de vídeo pode inserir B_1 na lista de candidatos à fusão (316). Por outro lado, em resposta à determinação de que B_1 corresponde a A_1 ou ao

IPMVC ("SIM" de 314) ou depois de inserir B_1 na lista de candidatos à fusão, o codificador de vídeo pode determinar se B_0 está disponível (318). Em resposta à determinação de que B_0 está disponível ("SIM" de 318), o codificador de vídeo pode inserir B_0 na lista de candidatos à fusão (320). Se B_2 não estiver disponível ("NÃO" de 318) ou depois de inserir B_0 na lista de candidatos à fusão, o codificador de vídeo pode determinar se o IDMVC está disponível e não corresponde a A_1 ou B_1 (332). O IDMVC pode especificar o vetor de disparidade para a PU atual. O IDMVC pode estar indisponível se o IDMVC indicar uma localização que está fora das fronteiras de uma imagem de referência inter-vista. Em resposta à determinação de que o IDMVC está disponível e não corresponde a A_1 ou B_1 ("SIM" de 332) o codificador de vídeo pode inserir o IDMVC na lista de candidatos à fusão (334). Se o IDMVC não estiver disponível ou se IDMVC corresponder a A_1 ou B_1 ("NÃO" de 332) ou depois de inserir o IDMVC na lista de candidatos à fusão, o codificador de vídeo pode executar a protocolo da operação 300 mostrada na Figura 12 (denotada por "A").

[0182] A Figura 12 é um fluxograma que mostra uma segunda parte da operação 300 exemplar da Figura 11 para construir uma lista de candidatos à fusão para o bloco atual, de acordo com uma ou mais técnicas desta revelação. Conforme indicado acima, o codificador de vídeo pode executar a parte da operação 300 mostrada na Figura 12 se o IDMVC não estiver disponível ou se IDMVC responder a um ou de um ("NÃO" lista de candidatos à fusão. Consequentemente, se o IDMVC não estiver disponível ou se o IDMVC corresponder a A_1 ou B_1 ("NÃO" de 332) ou depois de inserir o IDMVC na lista de candidatos à fusão, o codificador de vídeo pode determinar se a BVSP está habilitada (336). Se a BVSP estiver habilitada ("SIM" de 336), o codificador de

vídeo pode inserir um candidato a DVSP na lista de candidatos à fusão (338). Se a BVSP não estiver habilitada ("NÃO" de 336) ou depois de inserir o candidato a BVSP na lista de candidatos à fusão, o codificador de vídeo pode determinar se A_0 está disponível 340. Se A_0 estiver disponível ("SIM" de 340) o codificador de vídeo pode inserir A_0 na lista de candidatos à fusão (342). Caso contrário, se A_0 não estiver disponível ("NÃO" de 340) ou depois de inserir A_0 na lista de candidatos à fusão, o codificador de vídeo pode determinar se B_2 está disponível (344). Se B_2 estiver disponível ("SIM" de 344) o codificador de vídeo pode inserir B_2 na lista de candidatos à fusão (346).

[0183] Se B_2 não estiver disponível ("NÃO" de 344) ou depois de inserir B_2 na lista de candidatos à fusão, o codificador de vídeo pode determinar se a predição de móvel inter-vista é aplicada (348). Em outras palavras, o codificador de vídeo pode determinar se o bloco atual pode ser codificado utilizando-se predição de movimento inter-vista. Em resposta à determinação de que a predição de movimento inter-vista é aplicada ("SIM" de 348) o codificador de vídeo pode determinar um candidato deslocado (350). Em outras palavras, o codificador de vídeo pode determinar um candidato a DSMV, conforme descrito em outro lugar nesta revelação. Depois de determinar o candidato deslocado, o decodificador de vídeo 30 pode determinar se o candidato deslocado está disponível (352). Se um candidato deslocado estiver disponível ("SIM" de 352), o codificador de vídeo pode incluir um candidato deslocado na lista de candidatos à fusão (354). Se a predição de movimento inter-vista não for aplicado ("NÃO" de 348), o candidato deslocado não está disponível ("NÃO" de 352) ou depois de incluir o candidato deslocado na lista de candidatos à

fusão, o codificador de vídeo pode incluir um candidato à fusão temporal na lista de candidatos à fusão (356).

[0184] Além disto, o codificador de vídeo pode executar um processo de derivação para candidatos à fusão bi-preditiva combinada (358). Um processo de derivação exemplar para candidatos à fusão bi-preditiva combinada de acordo com uma ou mais técnicas desta revelação é descrito em seguida com relação à Figura 13. Além disto, o codificador de vídeo pode executar um processo de derivação para zero candidatos a vetor de movimento (360). Um processo de derivação exemplar para zero candidatos a vetor de movimento é descrito na seção 8.5.3.2.4 do WD HEVC 10.

[0185] A Figura 13 é um fluxograma que mostra um processo de derivação exemplar para candidatos à fusão bi-preditiva combinada de acordo com uma ou mais técnicas desta revelação. O processo de derivação da Figura 13 pode ser executado sem a verificação de quaisquer indicadores de BVSP. Por exemplo, o processo de derivação da Figura 13 pode ser executado sem apresentar `mergCandIsVspFlag` como entrada para o processo de derivação para candidatos à fusão bi-preditiva combinada, conforme é feito na seção H. 8.5.3.2.1 do Texto de Rascunho 3D-HEVC 1. Além disto, o processo de derivação da Figura 13 pode ser executado sem a utilização de `mergCandIsVspFlag` no processo de derivação para candidatos à fusão bi-preditiva combinada, conforme é feito na seção H. 8.5.3.2.3 do Texto de Rascunho 3D-HEVC 1.

[0186] No exemplo da Figura 13, um codificador de vídeo (o codificador de vídeo 20 ou o decodificador de vídeo 30, por exemplo) pode determinar se a fatia atual (isto é, a fatia que o codificador de vídeo está codificando atualmente) é uma fatia B (400). Se a fatia atual não for uma fatia B ("NÃO" de 400), o codificador de vídeo pode encerrar o processo de derivação para candidatos

à fusão bi-preditiva combinada. Entretanto, em resposta à determinação de que a fatia atual é uma fatia B ("SIM" de 400), o codificador de vídeo pode determinar se o número de candidatos à fusão na lista de candidatos à fusão (isto é, a lista de candidatos à fusão) é menor que 5 (402). Se o número de candidatos à fusão na lista de candidatos à fusão não for menor que 5, o codificador de vídeo pode encerrar o processo de derivação para candidatos à fusão bi-preditiva combinada.

[0187] Por outro lado, em resposta à determinação de que o número de candidatos à fusão na lista de candidatos à fusão é menor que 5 ("SIM" de 402), o codificador de vídeo pode fixar o valor de um índice de combinação (comIdx, por exemplo) em zero (404). O codificador de vídeo pode determinar então se os vetores de movimento que correspondem ao valor atual do índice de combinação estão disponíveis (406).

[0188] Em resposta à determinação de que os vetores de movimento que correspondem ao valor atual do índice de combinação estão disponíveis ("SIM" de 406) o codificador de vídeo pode incluir um candidato à fusão bi-preditiva combinada associado ao valor atual no índice de combinação na lista de candidatos à fusão (408). O candidato à fusão bi-preditiva combinada associado ao valor atual do índice de combinação pode especificar as informações de movimento RefPicList0 e as informações de movimento RefPicList1 de acordo com a Tabela 1.

[0189] Além disto, o codificador de vídeo pode determinar se o valor atual do índice de combinação é igual a $(\text{numOrigMergeCand} * (\text{numOrigMergeCand} - 1))$, onde numOrigMergeCand denota o número de candidatos na lista de candidatos à fusão antes da chamada do processo de derivação da Figura 13 (410). Se o valor atual do índice de

combinação for igual $(\text{numOrigMergeCand} * (\text{numOrigMergeCand} - 1))$ ("SIM" de 410), o codificador de vídeo pode encerrar o processo de derivação para candidatos à fusão bi-preditiva combinada. Por outro lado, se o valor atual do índice de combinação não for igual a $(\text{numOrigMergeCand} * (\text{numOrigMergeCand} - 1))$ ("NÃO" de 410) o codificador de vídeo pode determinar se o número total de candidatos à fusão na lista de candidatos à fusão é igual a MaxNumMergeCand (412). Conforme indicado em outro lugar nesta revelação, MaxNumMergeCand indica o número máximo de candidatos à fusão na lista de candidatos à fusão. Se o número total de candidatos à fusão na lista de candidatos à fusão for igual a MaxNumMergeCand ("SIM" de 412), o codificador de vídeo pode encerrar o processo de derivação para candidatos à fusão bi-preditiva combinada.

[0190] Entretanto, em resposta à determinação de que o número total de candidatos à fusão na lista de candidatos à fusão não é igual a MaxNumMergeCand ("NÃO" de 412) ou em resposta à determinação de que os vetores de movimento que correspondem ao valor atual do índice de combinação não estão disponíveis ("NÃO" de 406), o codificador de vídeo pode incrementar o valor atual do índice de combinação (414). O codificador de vídeo pode executar ações (406)-(414), com relação ao valor incrementado do índice de combinação. Desta maneira, o codificador de vídeo pode continuar a derivar candidatos à fusão bi-preditiva combinada até que o valor atual do índice de combinação seja igual a $(\text{numOrigMergeCand} * (\text{numOrigMergeCand} - 1))$ ou o número de candidatos totais, inclusive candidatos à fusão bi-preditiva combinada (recém gerados) na lista de fusão seja igual MaxNumMergeCand .

[0191] A Figura 14A é um fluxograma que mostra uma operação exemplar do codificador de vídeo 20 para codificar um bloco de vídeo, de acordo com uma ou mais técnicas desta revelação. No exemplo da Figura 14A, o codificador de vídeo 20 pode gerar uma lista de candidatos à fusão (450). Em outras palavras, o codificador de vídeo 20 pode gerar uma lista de candidatos à fusão. No exemplo da Figura 14A, o codificador de vídeo 20 pode determinar se o número de candidatos à fusão na lista é menor que 5 (452). Em alguns exemplos, o codificador de vídeo 20 pode, nesta etapa, determinar se o número de candidatos à fusão na lista é menor que 5 e o número máximo de candidatos à fusão na lista é maior que 5 (igual a 6, por exemplo). Em resposta à determinação de o número de candidatos à fusão na lista é menor que 5 ("SIM" de 452), o codificador de vídeo 20 pode derivar um ou mais candidatos à fusão bi-preditiva combinada (454) e incluir o candidato ou candidatos à fusão bi-preditiva combinada na lista de candidatos à fusão 456. Cada respectivo candidato à fusão bi-preditiva combinada do candidato ou candidatos à fusão bi-preditiva combinada pode corresponder a um respectivo par de candidatos à fusão já na lista. O respectivo candidato à fusão bi-preditiva combinada pode ser uma combinação de vetor de movimento de um primeiro candidato à fusão do respectivo par e vetor de movimento de um segundo candidato à fusão do respectivo par. O vetor de movimento do primeiro candidato à fusão e o vetor de movimento do segundo candidato à fusão, referem-se a lista de imagens de referência diferentes (lista 0 e lista 1). Por outro lado, em alguns exemplos, se o número de candidatos na lista não for menor que 5 ("NÃO" de 452), o codificador de vídeo 20 não inclui nenhum candidato à fusão bi-preditiva combinada na lista (458).

[0192] Em alguns exemplos, o codificador de vídeo 20 pode derivar o candidato ou candidatos à fusão bi-preditiva combinada depois de inserir um IPMVC, se disponível, na lista de candidatos à fusão, depois de executar um processo de derivação para candidatos à fusão espaciais e depois de executar um processo de derivação para um candidato à fusão temporal. Um processo de derivação para candidatos à fusão espaciais pode derivar e inserir até quatro candidatos a vetor de movimento espaciais na lista de candidatos à fusão. O processo de derivação para o candidato à fusão temporal pode adicionar um candidato a preditor de vetor de movimento temporal (TMVP), se disponível, à lista de candidatos à fusão.

[0193] Além disto, no exemplo da Figura 14A, o codificador de vídeo 20 pode selecionar um candidato na lista de candidatos à fusão (460). Em alguns exemplos, o codificador de vídeo 20 pode determinar o candidato selecionado com base no valor indicado por um elemento de sintaxe sinalizado em um fluxo de bits. Além disto, o codificador de vídeo 20 pode sinalizar a posição na lista de candidatos à fusão do candidato à fusão selecionado (452). O codificador de vídeo 20 pode codificar um bloco de vídeo com base no candidato selecionado (564). O codificador de vídeo 20 pode codificar o bloco de vídeo de acordo com um ou mais dos exemplos apresentados em outro lugar nesta revelação.

[0194] A Figura 14B é um fluxograma que mostra uma operação exemplar do decodificador de vídeo 30 para decodificar um bloco de vídeo, de acordo com uma ou mais técnicas revelação. No exemplo da Figura 14B, o decodificador de vídeo 30 pode gerar uma lista de candidatos à fusão (480). No exemplo da Figura 14B, o decodificador de vídeo 30 pode determinar se o número de

candidatos à fusão na lista é menor que 5 (482). Em alguns exemplos, o decodificador de vídeo 30 pode, nesta etapa, determinar se o número de candidatos à fusão na lista é menor que 5 e o número máximo de candidatos à fusão na lista é igual a 6, por exemplo. Em resposta à determinação de que o número de candidatos à fusão na lista é menor que 5 ("SIM" de 452), o decodificador de vídeo 30 pode derivar um ou mais candidatos à fusão bi-preditiva combinada (484). Cada respectivo candidato à fusão bi-preditiva combinada do candidato ou candidatos à fusão bi-preditiva combinada pode corresponder a um par de candidatos à fusão já na lista. O respectivo candidato à fusão bi-preditiva combinada pode ser uma combinação de vetor de movimento de um primeiro candidato à fusão do respectivo par e vetor de movimento de um segundo candidato à fusão do respectivo par. O vetor de movimento do primeiro candidato à fusão e o vetor de movimento do segundo candidato à fusão referem-se a imagens em lista de imagens de referência diferentes (lista 0 e lista 1, por exemplo). O decodificador de vídeo 30 pode incluir o candidato ou candidatos à fusão bi-preditiva combinada na lista (586). Por outro lado, em alguns exemplos, se o número de candidatos à fusão na lista não for menor que 5 ("NÃO" de 482), o decodificador de vídeo 30 não inclui qualquer candidato à fusão bi-preditiva combinada na lista (488).

[0195] Em alguns exemplos, o decodificador de vídeo 30 pode derivar o candidato ou candidatos à fusão bi-preditiva combinada depois de inserir um IPMVC, se disponível, na lista de candidatos à fusão, depois de executar um processo de derivação para candidatos à fusão espaciais e depois de executar um processo de derivação para um candidato à fusão temporal. O processo de derivação para candidatos à fusão espaciais pode derivar e inserir

até quatro candidatos a vetor de movimento espaciais na lista de candidatos à fusão. O processo de derivação para o candidato à fusão temporal pode adicionar um candidato a preditor de vetor de movimento temporal (TMVP), se disponível, à lista de candidatos à fusão.

[0196] Além disso, no exemplo da Figura 14B, o decodificador de vídeo 30 pode determinar um candidato selecionado na lista de candidatos à fusão (490). Em alguns exemplos, o decodificador de vídeo 30 pode determinar o candidato selecionado com base no valor indicado por um elemento de sintaxe sinalizado em um fluxo de bits. Por exemplo, o decodificador de vídeo 30 pode obter, de um fluxo de bits, um elemento de sintaxe que indica um candidato à fusão selecionado na lista de candidatos à fusão. O decodificador de vídeo 30 pode decodificar um bloco de vídeo com base no candidato selecionado (492). Por exemplo, o decodificador de vídeo 30 pode utilizar informações de movimento do candidato selecionado para gerar amostras preditivas da PU atual. O decodificador de vídeo 30 pode decodificar o bloco de vídeo (como, por exemplo, uma CU, PU, etc.) de acordo com um ou mais dos exemplos apresentados em outro lugar nesta revelação.

[0197] Os parágrafos seguintes apresentam exemplos adicionais desta revelação.

[0198] Exemplo 1. Um método para codificar dados de vídeo, o método compreendendo: gerar uma primeira lista de candidatos à fusão de acordo com um primeiro processo para codificar um bloco de vídeo que não está associado a dados de vídeo tridimensionais, no qual a primeira lista inclui um ou mais candidatos à fusão bi-preditiva; e gerar uma segunda lista de candidatos à fusão de acordo com um segundo processo para codificar um bloco de vídeo que está associado a dados de vídeo

tridimensionais, no qual a segunda lista inclui um ou mais candidatos à fusão bi-preditiva, no qual o primeiro processo e o segundo processo são os mesmos.

[0199] Exemplo 2. O método do exemplo 1, no qual a geração da primeira lista e a geração da segunda lista ocorrem apenas quando a condição seguinte é satisfeita: o número de candidatos à fusão disponíveis é menor que 5.

[0200] Exemplo 3. O método de qualquer um dos exemplos 1 ou 2, que compreende também definir o número máximo de candidatos à fusão AMVP de fusão antes de se chamar um processo de derivação para gerar qualquer lista de fusão.

[0201] Exemplo 4. O método do exemplo 4, no qual o número máximo de candidatos a MVP de fusão é definido substancialmente da maneira seguinte: $\text{MaxNumMergeCand} = 5 - \text{five_minus_max_num_merge_cand}$ e, em seguida, depois que esse processo é chamado MaxNumMergeCand é reconfigurado em: $\text{MaxNumMergeCand} = 5 - \text{five_minus_max_num_merge_cand} + \text{iv_mv_pred_flag}[\text{nuh_layer_id}]$.

[0202] Um método para codificar dados associados a vídeo tridimensional 3D, o método compreendendo: gerar uma lista de candidatos à fusão para codificar um bloco de vídeo associado a vídeo 3D, no qual a lista inclui um ou mais candidatos à fusão bi-preditivo combinado e no qual, quando o número máximo de candidatos à fusão é igual a 6 e há 5 candidatos antes que um processo de derivação de candidatos à fusão bi-preditiva combinada seja chamado, um candidato zero é gerado e incluído na lista, em que um candidato zero define um índice de referência em componentes de vetor de movimento como 0.

[0203] Exemplo 6. Um método para codificar dados associados a vídeo tridimensional (3D) o método compreendendo: gerar uma lista de candidatos à fusão para codificar um bloco de vídeo associado à vídeo 3D, no qual a lista inclui um ou mais candidatos à fusão bi-preditiva e no qual, antes de se gerar a lista, o número máximo de candidatos à fusão é fixado em 5, 4 dos candidatos são introduzidos em um processo de derivação de lista de fusão e um candidato é recém-gerado durante o processo de derivação de lista de fusão.

[0204] Exemplo 7. O método do exemplo 6, no qual, o candidato recém gerado é ordenado como o quinto candidato na lista.

[0205] Exemplo 8. O método do exemplo 6, no qual, se o processo de derivação de lista de fusão for incapaz de gerar um candidato recém gerado não zero, o processo de derivação de lista de fusão gera um candidato de valor zero como o candidato recém gerado.

[0206] Em um ou mais exemplos, as funções aqui descritas podem ser implementadas em hardware, software, firmware ou qualquer combinação deles. Se implementadas em software, as funções podem ser armazenadas ou transmitidas como uma ou mais instruções ou código, em um meio passível de leitura por computador e executadas por uma unidade de processamento baseada em hardware. Os meios passíveis de leitura por computador podem incluir meios de armazenamento passíveis de leitura por computador que correspondem a um meio tangível tal como meios de armazenamento de dados ou meios de comunicação que incluam qualquer meio que facilite a transferência de um programa de computador de um lugar para outro, como, por exemplo, de acordo com um protocolo de comunicação. Desta maneira, os meios passíveis de leitura por computador podem corresponder geralmente a (1)

meios de armazenamento passíveis de leitura por computador tangíveis que são não transitórios ou (2) a um meio de comunicação tal como um sinal ou onda portadora. Os meios de armazenamento de dados podem ser quaisquer meios disponíveis que possam ser acessados por um ou mais computadores ou um ou mais processadores para recuperar instruções, códigos e/ou estruturas de dados para implementação das técnicas descritas nesta revelação. Um produto de programa de computador pode incluir um meio passível de leitura por computador.

[0207] A título de exemplo, e não de limitação, tais meios de armazenamento passíveis de leitura por computador podem compreender RAM, ROM, EEPROM, CD-ROM ou outro armazenamento em disco óptico, armazenamento em disco magnético ou outros aparelhos de armazenamento magnético, memória flash ou qualquer outro meio que possa ser utilizado para armazenar código de programa desejado sob a forma de instruções ou estruturas de dados e que possa ser acessado por um computador. Além disto, qualquer conexão é denominada apropriadamente de meio passível de leitura por computador. Por exemplo, se as instruções forem transmitidas de um site da Web, servidor ou outra fonte remota utilizando-se um cabo coaxial, um cabo de fibra ótica, um par trançado, uma linha de assinante digital (DSL) ou tecnologias sem fio tais como infravermelho, rádio e microonda, então o cabo coaxial, o cabo de fibra ótica, o par trançado, a DSL ou tecnologias sem fio tais como infravermelho, rádio e microonda são incluídos na definição de meio. Deve ficar entendido, contudo, que os meios de armazenamento passíveis de leitura por computador e os meios de armazenamento de dados não incluem conexões, ondas portadoras, sinais ou outros meios transitórios, mas são em vez disso direcionados a meios de armazenamento não

transitórios, tangíveis. Disco (*disk* e *disc*) conforme aqui utilizado, inclui disco compacto (CD), disco de laser, disco óptico, disco versátil digital (DVD), disco flexível e disco Blu-ray, onde discos (*disks*) reproduzem usualmente dados magneticamente, enquanto discos (*discs*) reproduzem dados opticamente com lasers. Combinações dos componentes acima devem ser também incluídas dentro do alcance dos meios passíveis de leitura por computador.

[0208] As instruções podem ser executadas por um ou mais processadores, tais como um ou mais processadores de sinais digitais (DSPs), microprocessadores de propósito geral, circuitos integrados específicos de aplicativo (ASICs), arranjos lógicos programáveis no campo (FPGAs) ou outros circuitos lógicos integrados ou discretos equivalentes. Por conseguinte, o termo processador conforme aqui utilizado, pode referir-se a qualquer uma das estruturas precedentes ou a qualquer outra estrutura adequada para implementação das técnicas aqui descritas. Além disto, sob alguns aspectos, a funcionalidade aqui descrita pode ser proporcionada dentro de hardware dedicado e/ou módulos de software configurados para codificação e decodificação ou incorporados a um codec combinado. Além disto, as técnicas podem ser completamente implementadas em um ou mais circuitos ou elementos lógicos.

[0209] As técnicas desta revelação podem ser implementadas em uma ampla variedade de aparelhos ou equipamentos, que incluem um aparelho telefônico sem fio, um circuito integrado (IC) ou um conjunto de ICs (um conjunto de chips, por exemplo). Diversos componentes, módulos ou unidades são descritos nesta revelação para enfatizar aspectos funcionais de aparelhos configurados para executar as técnicas descritas, mas não exigem necessariamente execução por unidades de hardware

diferentes. Em vez disso, conforme descrito acima, diversas unidades podem ser combinadas em uma unidade de hardware de codec ou apresentadas por uma coleção de unidades de hardware interoperantes, que incluem um ou mais processadores descritos acima, em conjunto com software e/ou firmware adequado.

[0210] Foram descritos diversos exemplos. Estes e outros exemplos estão dentro do alcance das reivindicações seguintes.

REIVINDICAÇÕES

1. Método para codificar dados de vídeo tridimensionais (3D):

gerar uma lista de candidatos à fusão para codificar um bloco de vídeo do dados de vídeo 3D, no qual o número máximo de candidatos à fusão na lista de candidatos à fusão é igual a 6 e gerar a lista de candidatos à fusão compreende:

determinar se o número de candidatos à fusão na lista de candidatos à fusão é menor que 5; e

em resposta à determinação de que o número de candidatos à fusão na lista de candidatos à fusão é menor que 5:

derivar um ou mais candidatos à fusão bi-preditiva combinada no qual cada respectivo candidato à fusão bi-preditiva combinada do candidato ou candidatos à fusão bi-preditiva combinada corresponde a um respectivo par de candidatos à fusão já na lista de candidatos à fusão, no qual o respectivo candidato à fusão bi-preditiva combinada é uma combinação de vetor de movimento de um primeiro candidato à fusão do respectivo par e vetor de movimento de um segundo candidato à fusão do respectivo par, no qual o vetor de movimento do primeiro candidato à fusão e o vetor de movimento do segundo candidato à fusão referem-se a imagens em listas de imagens de referência diferentes; e

incluir o candidato ou candidatos à fusão bi-preditiva combinada na lista de candidatos à fusão.

2. Método, de acordo com a reivindicação 1, no qual gerar a lista de candidatos à fusão compreende também:

em resposta à determinação de que há 5 candidatos à fusão na lista de candidatos à fusão antes da adição de qualquer um do candidato ou candidatos à fusão bi-preditiva

à lista de candidatos à fusão, incluir um candidato zero na lista de candidatos à fusão, no qual os componentes de vetor de movimento do candidato zero são iguais a zero, e o índice de referência do candidato zero é igual a zero, o índice de referência indicando a localização de uma imagem de referência em uma lista de imagens de referência.

3. Método, de acordo com a reivindicação 2, no qual a geração da lista de candidatos à fusão ocorre sem verificação de quaisquer indicadores de síntese de vista de deformação retroativa (BVSP).

4. Método, de acordo com a reivindicação 1, no qual o método para codificar os dados compreende um método para decodificar os dados de vídeo 3D, e o bloco de vídeo é uma unidade de predição (PU), o método compreendendo também:

obter, de um fluxo de bits, um elemento de sintaxe que indica um candidato à fusão selecionado na lista de candidatos à fusão; e

utilizar as informações de movimento do candidato selecionado para gerar amostras preditivas da PU.

5. Método, de acordo com a reivindicação 1, no qual o método para codificar os dados compreende um método para codificar os dados de vídeo 3D, o método compreendendo:

selecionar um candidato à fusão na lista de candidatos à fusão; e

sinalizar a posição na lista de candidatos à fusão do candidato à fusão selecionado.

6. Método, de acordo com a reivindicação 1, no qual:

gerar a lista de candidatos à fusão compreende derivar o candidato ou candidatos à fusão bi-preditiva combinada depois de inserir um candidato a vetor de

movimento de predição inter-vista (IPMVC), se disponível na lista de candidatos à fusão, depois de executar um processador de derivação para candidatos à fusão espaciais e depois de executar um processo de derivação para um candidato à fusão temporal,

o processo de derivação para candidatos à fusão espaciais deriva e insere até quatro candidatos a vetor de movimento espaciais na lista de candidatos à fusão, e

o processo de derivação para o candidato à fusão temporal adiciona um candidato a preditor de movimento temporal (TMVP), se disponível, na lista de candidatos à fusão.

7. Aparelho de codificação de vídeo que compreende:

um meio de armazenamento de dados configurado para armazenar dados de vídeo tridimensionais (3D); e

um ou mais processadores configurados para:

gerar uma lista de candidatos à fusão para codificar um bloco de vídeo dos dados de vídeo 3D, no qual o número máximo de candidatos à fusão na lista de candidatos à fusão é igual a 6 e como parte da geração da lista de candidatos à fusão, o processo ou processadores:

determinam se o número de candidatos à fusão na lista de candidatos à fusão é menor que 5; e

em resposta à determinação de que o número de candidatos à fusão na lista de candidatos à fusão é menor que 5:

derivam um ou mais candidatos à fusão bi-preditiva combinada no qual cada respectivo candidato à fusão bi-preditiva combinada do candidato ou candidatos à fusão bi-preditiva combinada corresponde a um respectivo par de candidatos à fusão já na lista de candidatos à fusão, no qual o respectivo candidato à fusão bi-preditiva

combinada é uma combinação de vetor de movimento de um primeiro candidato à fusão do respectivo par e vetor de movimento de um segundo candidato à fusão do respectivo par, no qual o vetor de movimento do primeiro candidato à fusão e o vetor de movimento do segundo candidato à fusão referem-se a imagens em listas de imagens de referência diferentes; e

incluem o candidato ou candidatos à fusão bi-preditiva combinada na lista de candidatos à fusão.

8. Aparelho de codificação de vídeo, de acordo com a reivindicação 7, no qual, como parte da geração da lista de candidatos à fusão, o processador ou processadores:

incluem, em resposta à determinação de que há 5 candidatos à fusão na lista de candidatos à fusão antes da adição de qualquer um do candidato ou candidatos à fusão bi-preditiva à lista de candidatos à fusão, um candidato zero na lista de candidatos à fusão, no qual os componentes de vetor de movimento do candidato zero são iguais a zero, e o índice de referência do candidato zero é igual a zero, o índice de referência indicando a localização de uma imagem de referência em uma lista de imagens de referência.

9. Aparelho de codificação de vídeo, de acordo com a reivindicação 7, no qual o processador ou processadores geram a lista de candidatos à fusão sem verificar quaisquer indicadores de síntese de vista de deformação retroativa (BVSP).

10. Aparelho de codificação de vídeo, de acordo com a reivindicação 7, no qual o processador ou processadores são configurados para decodificar os dados de vídeo 3D, e o bloco de vídeo é uma unidade de predição (PU), o processador ou processadores sendo configurados para

obter, de um fluxo de bits, um elemento de sintaxe que indica um candidato à fusão selecionado na lista de candidatos à fusão; e

utilizar as informações de movimento do candidato selecionado para gerar amostras preditivas da PU.

11. Aparelho de codificação de vídeo, de acordo com a reivindicação 7, no qual o processador ou processadores são configurados para codificar os dados de vídeo 3D, o processador ou processadores sendo configurados para:

selecionar um candidato à fusão na lista de candidatos à fusão; e

sinalizar a posição na lista de candidatos à fusão do candidato à fusão selecionado.

12. Aparelho de codificação de vídeo, de acordo com a reivindicação 7, no qual:

o processador ou processadores são configurados para derivar o candidato ou candidatos à fusão bi-preditiva combinada depois de inserir um candidato a vetor de movimento de predição inter-vista (IPMVC), se disponível, na lista de candidatos à fusão, depois de executar um processo de derivação para candidatos à fusão espaciais e depois de executar um processo de derivação para um candidato à fusão temporal,

o processo de derivação para candidatos à fusão espaciais deriva e insere até quatro candidatos a vetor de movimento espaciais na lista de candidatos à fusão, e

o processo de derivação para o candidato à fusão temporal adiciona um candidato a preditor de movimento temporal (TMVP), se disponível, na lista de candidatos à fusão.

13. Aparelho de codificação de vídeo que compreende:

um dispositivo para gerar uma lista de candidatos à fusão para codificar um bloco de vídeo de dados de vídeo tridimensionais (3D), no qual o número máximo de candidatos à fusão na lista de candidatos à fusão é igual a 6 e o dispositivo para grupo a lista de candidatos à fusão compreende:

um dispositivo para determinar se o número de candidatos à fusão na lista de candidatos à fusão é menor que 5;

um dispositivo para derivar, em resposta à determinação de que o número de candidatos à fusão na lista de candidatos à fusão é menor que 5, um ou mais candidatos à fusão bi-preditiva combinada, no qual cada revelado candidato à fusão bi-preditiva do candidato ou candidatos à fusão bi-preditiva combinada corresponde a um respectivo par de candidatos à fusão já na lista de candidatos à fusão, no qual o respectivo candidato à fusão bi-preditiva combinada é uma combinação de vetor de movimento de um primeiro candidato à fusão do respectivo par e vetor de movimento de um segundo candidato à fusão do respectivo par, no qual o vetor de movimento do primeiro candidato à fusão e o vetor de movimento do segundo candidato à fusão referem-se a imagens em lista de imagens de referência diferentes; e

um dispositivo para incluir o candidato ou candidatos à fusão bi-preditiva combinada na lista de candidatos à fusão.

14. Aparelho de codificação de vídeo, de acordo com a reivindicação 13, no qual o dispositivo para gerar a lista de candidatos à fusão compreende também:

um dispositivo para incluir, em resposta à determinação de que há 5 candidatos à fusão na lista de candidatos à fusão antes da adição de qualquer um do

candidato ou candidatos à fusão bi-preditiva à lista de candidatos à fusão, um candidato zero na lista de candidatos à fusão, no qual os componentes de vetor de movimento do candidato zero são iguais a zero e o índice de referência do candidato zero é igual a zero, o índice de referência indicando a localizações de uma imagem de referência em uma lista de imagens de referência.

15. Aparelho de codificação de vídeo, de acordo com a reivindicação 13, no qual a geração da lista de candidatos à fusão ocorre sem verificação de quaisquer indicadores de síntese de vista retroativa de deformação (BVSP).

16. Aparelho de codificação de vídeo, de acordo com a reivindicação 13, no qual o aparelho de codificação de vídeo decodifica os dados de vídeo 3D e o bloco de vídeo é uma unidade de predição (PU), o aparelho de codificação de vídeo compreendendo também:

um dispositivo para obter, de um fluxo de bits, um elemento de sintaxe que indica um candidato à fusão selecionado na lista de candidatos à fusão; e

um dispositivo para utilizar as informações de movimento do candidato selecionado para gerar amostras preditivas da PU.

17. Aparelho de codificação de vídeo, de acordo com a reivindicação 13, no qual o aparelho de codificação de vídeo codifica os dados de vídeo 3D e o aparelho de codificação de vídeo compreende:

um dispositivo para selecionar um candidato à fusão na lista de candidatos à fusão; e

um dispositivo para sinalizar a posição na lista de candidatos à fusão do candidato à fusão selecionado.

18. Aparelho de codificação de vídeo, de acordo com a reivindicação 13, no qual:

gerar a lista de candidatos à fusão compreende derivar o candidato ou candidatos à fusão bi-preditiva combinada depois de inserir um a vetor de movimento de predição inter-vista (IPMVC), se disponível, na lista de candidatos à fusão, depois de executar um processo de derivação para candidatos à fusão espaciais e depois de executar um processo de derivação para um candidato à fusão temporal,

o processo de derivação para candidatos à fusão espaciais deriva e insere até quatro candidatos a vetor de movimento espaciais na lista de candidatos à fusão, e

o processo de derivação para o candidato à fusão temporal adiciona um candidato a preditor de movimento temporal (TMVP), se disponível à lista de candidatos à fusão.

19. Meio de armazenamento de dados, passível de leitura por computador que tem instruções armazenadas nele que, quando executadas, fazem com que um aparelho de codificação de vídeo codifique dados de vídeo tridimensionais (3D), as instruções fazendo com que o aparelho de codificação de vídeo:

gere uma lista de candidatos à fusão para codificar um bloco de vídeo dos dados de vídeo 3D, no qual o número máximo de candidatos à fusão na lista de candidatos à fusão é igual a 6 e gerar a lista de candidatos à fusão compreende:

determinar se o número de candidatos à fusão na lista de candidatos à fusão é menor que 5; e

em resposta à determinação de que o número de candidatos à fusão na lista de candidatos à fusão é menor que 5:

derivar um ou mais candidatos à fusão bi-preditiva combinada no qual cada respectivo candidato à

fusão bi-preditiva combinada do candidato ou candidatos à fusão bi-preditiva combinada corresponde a um respectivo par de candidatos à fusão já na lista de candidatos à fusão, no qual o respectivo candidato à fusão bi-preditiva combinada é uma combinação de vetor de movimento de um primeiro candidato à fusão do respectivo par e vetor de movimento de um segundo candidato à fusão do respectivo par, no qual o vetor de movimento do primeiro candidato à fusão e o vetor de movimento do segundo candidato à fusão referem-se a imagens em listas de imagens de referência diferentes; e

incluir o candidato ou candidatos à fusão bi-preditiva combinada na lista de candidatos à fusão.

20. Meio de armazenamento de dados, passível de leitura por computador, de acordo com a reivindicação 19, no qual gerar a lista de candidatos à fusão compreende também:

em resposta à determinação de que há 5 candidatos à fusão na lista de candidatos à fusão antes da adição de qualquer um do candidato ou candidatos à fusão bi-preditiva à lista de candidatos à fusão, incluir um candidato zero na lista de candidatos à fusão, no qual os componentes de vetor de movimento do candidato zero são iguais a zero, e o índice de referência do candidato zero é igual a zero, o índice de referência indicando a localização de uma imagem de referência em uma lista de imagens de referência.

21. Meio de armazenamento de dados, passível de leitura por computador, de acordo com a reivindicação 19, no qual a geração da lista de candidatos à fusão ocorre sem verificação de quaisquer indicadores de síntese de vista de deformação retroativa (BVSP).

22. Meio de armazenamento de dados, passível de leitura por computador, de acordo com a reivindicação 19,

no qual o bloco de vídeo é uma unidade de predição (PU), as instruções fazendo também com que, o aparelho de codificação de vídeo:

obtenha, de um fluxo de bits, um elemento de sintaxe que indica um candidato à fusão selecionado na lista de candidatos à fusão; e

utilize as informações de movimento do candidato selecionado para gerar amostras preditivas da PU.

23. Meio de armazenamento de dados, passível de leitura por computador, de acordo com a reivindicação 29, no qual as instruções fazem também com que, o aparelho de codificação de vídeo:

selecione um candidato à fusão na lista de candidatos à fusão; e

sinalize a posição na lista de candidatos à fusão do candidato à fusão selecionado.

24. Meio de armazenamento de dados, passível de leitura por computador, de acordo com a reivindicação 19, no qual as instruções fazem com que o aparelho de codificação de vídeo derive o candidato ou candidatos à fusão bi-preditiva combinada depois de inserir um candidato a vetor de movimento de predição inter-vista (IPMVC), se disponível, na lista de candidatos à fusão, depois de executar um processo de derivação para candidatos à fusão espaciais e depois de executar um processo de derivação para um candidato à fusão temporal,

no qual o processo de derivação para candidatos à fusão espaciais deriva e insere até quatro a vetor de movimento espaciais na lista de candidatos à fusão, e

no qual, o processo de derivação para o candidato à fusão temporal adiciona um candidato a preditor de vetor de movimento temporal (TMVP) se disponível à lista de candidatos à fusão.

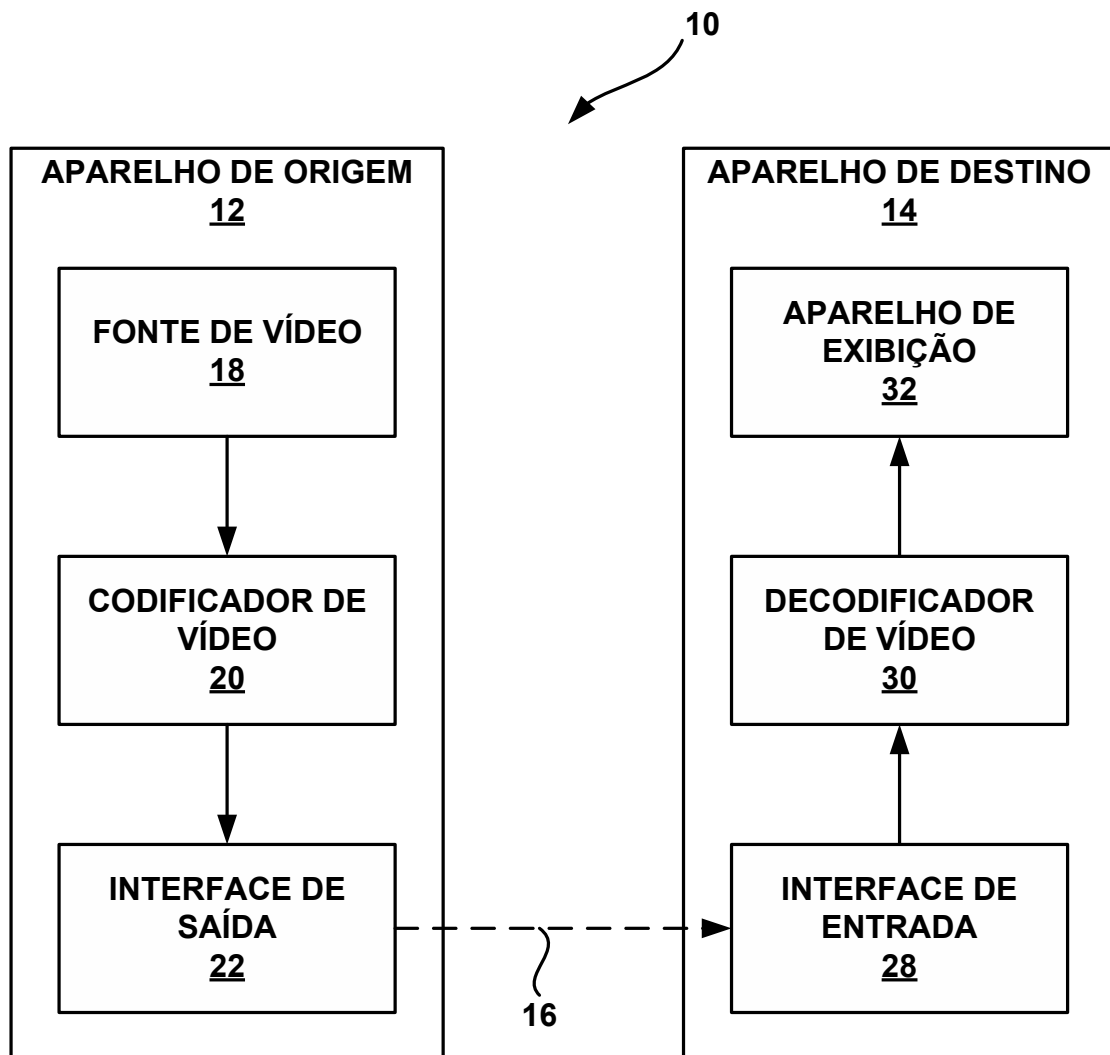
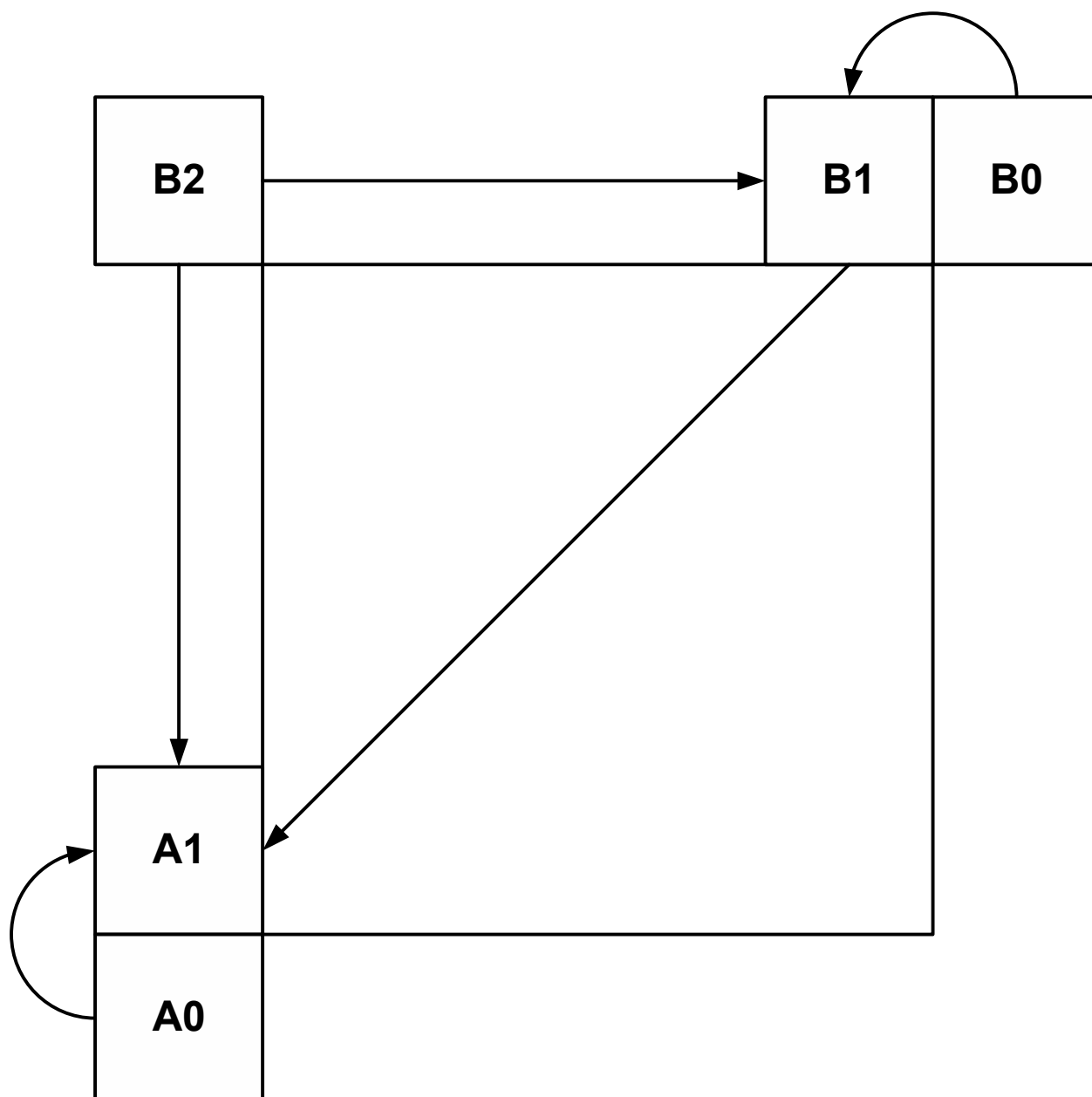


FIG. 1

**FIG. 2**

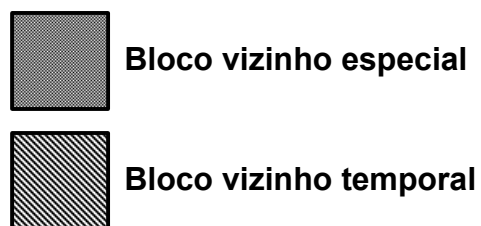
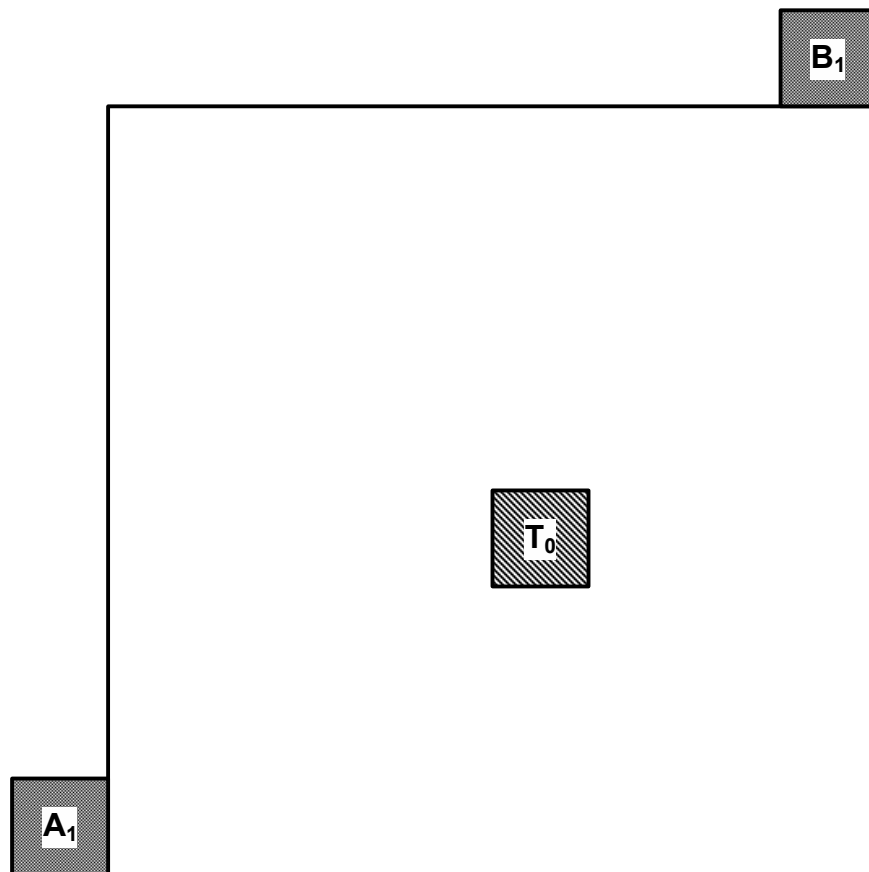


FIG. 3

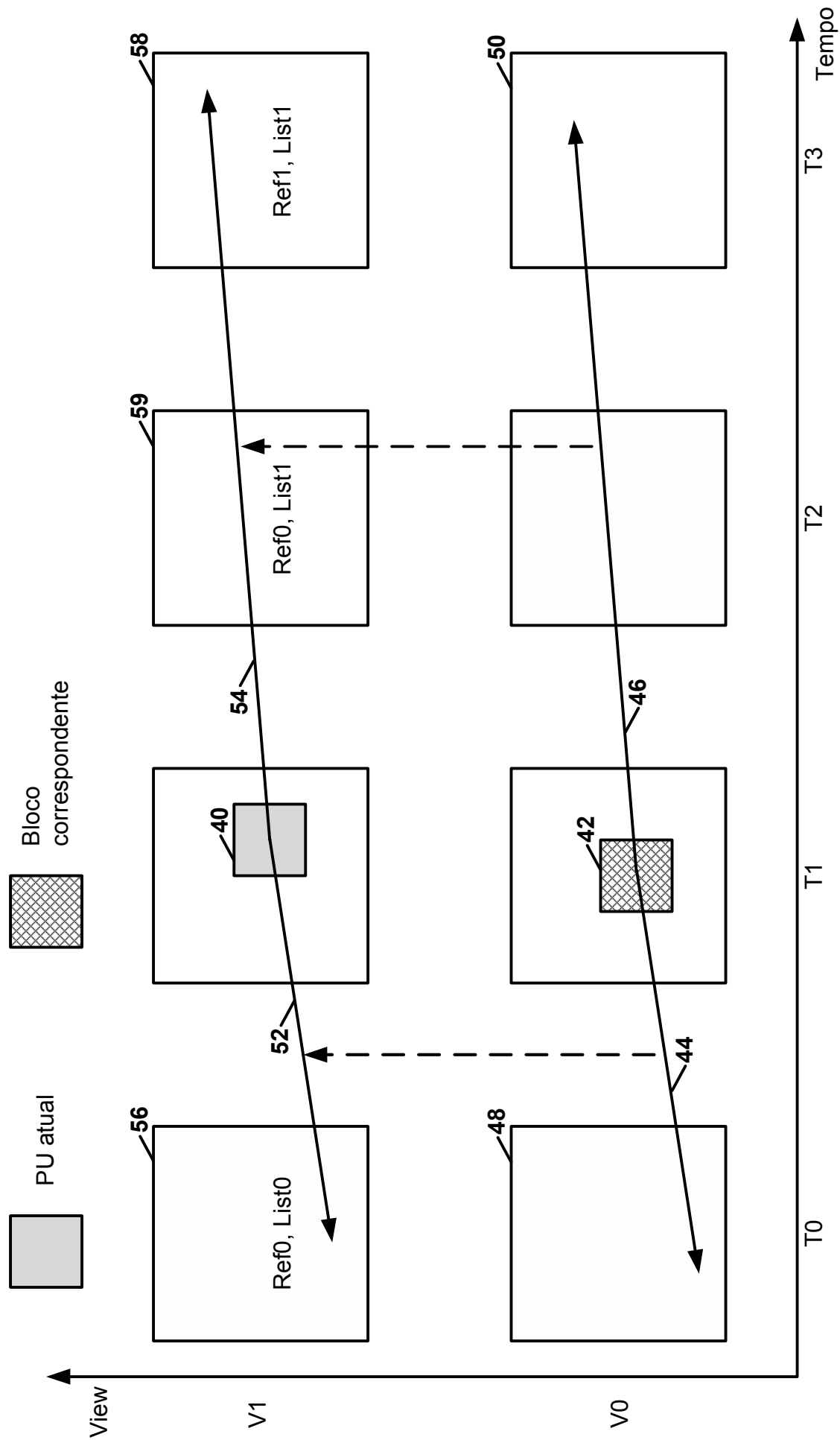


FIG. 4

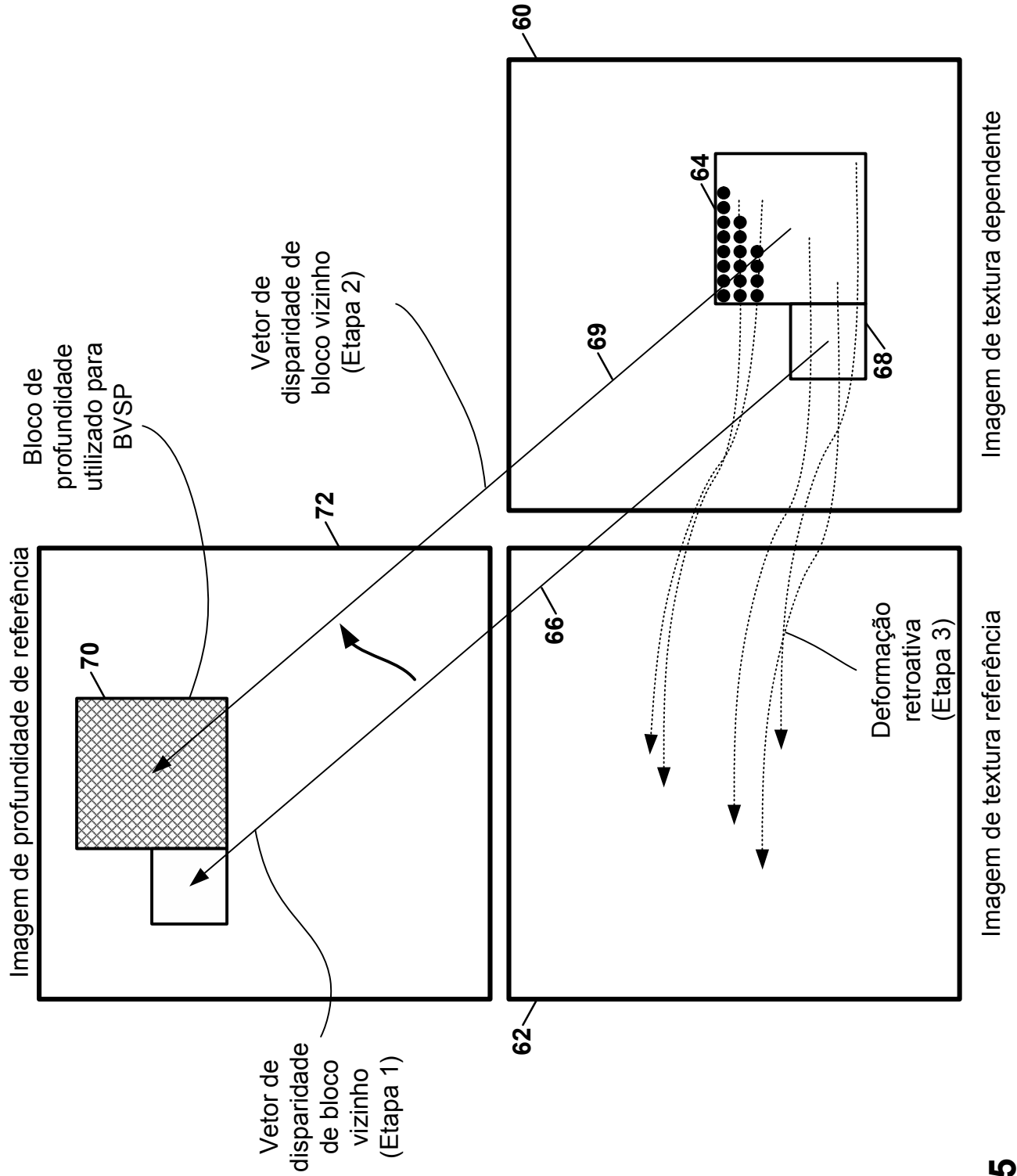


FIG. 5

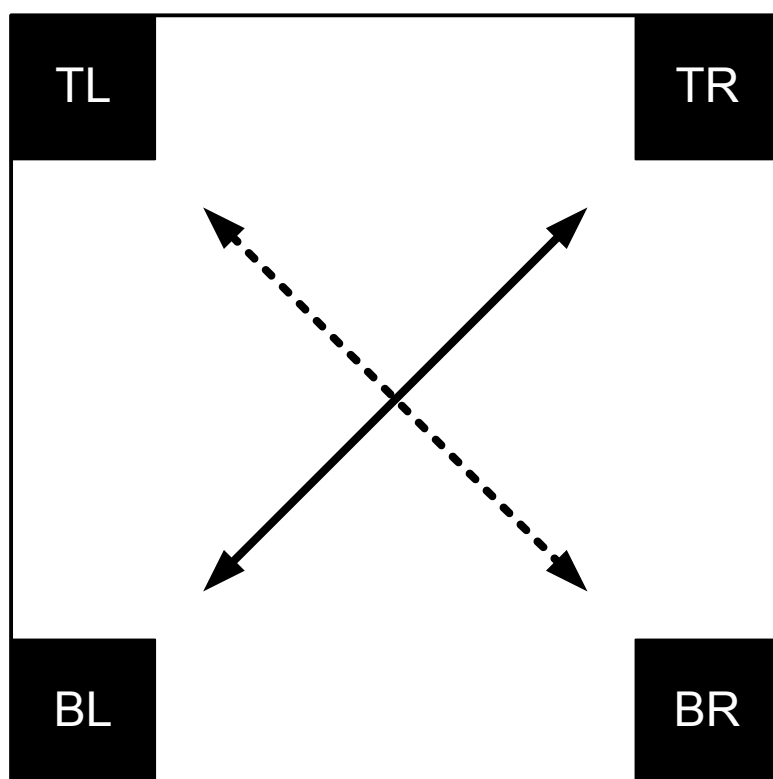


FIG. 6

| comblidx | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|-----------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|
| I0CandIdx | 0 | 1 | 0 | 2 | 1 | 2 | 0 | 3 | 1 | 3 | 2 | 3 | 0 | 4 | 1 | 4 | 2 | 4 | 3 | 4 |
| I1CandIdx | 1 | 0 | 2 | 0 | 2 | 1 | 3 | 0 | 3 | 1 | 3 | 2 | 4 | 0 | 4 | 1 | 4 | 2 | 4 | 3 |

FIG. 7

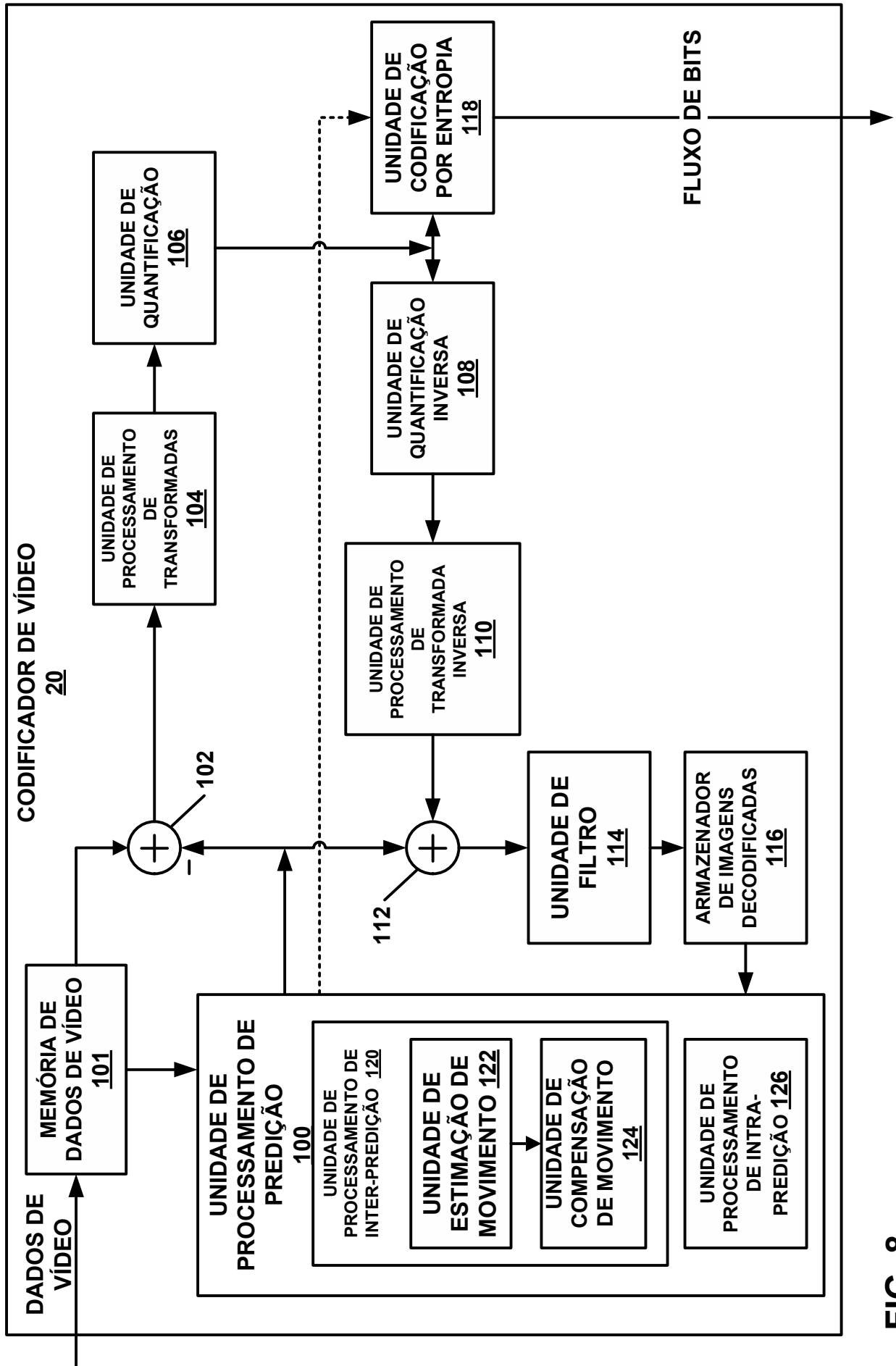


FIG. 8

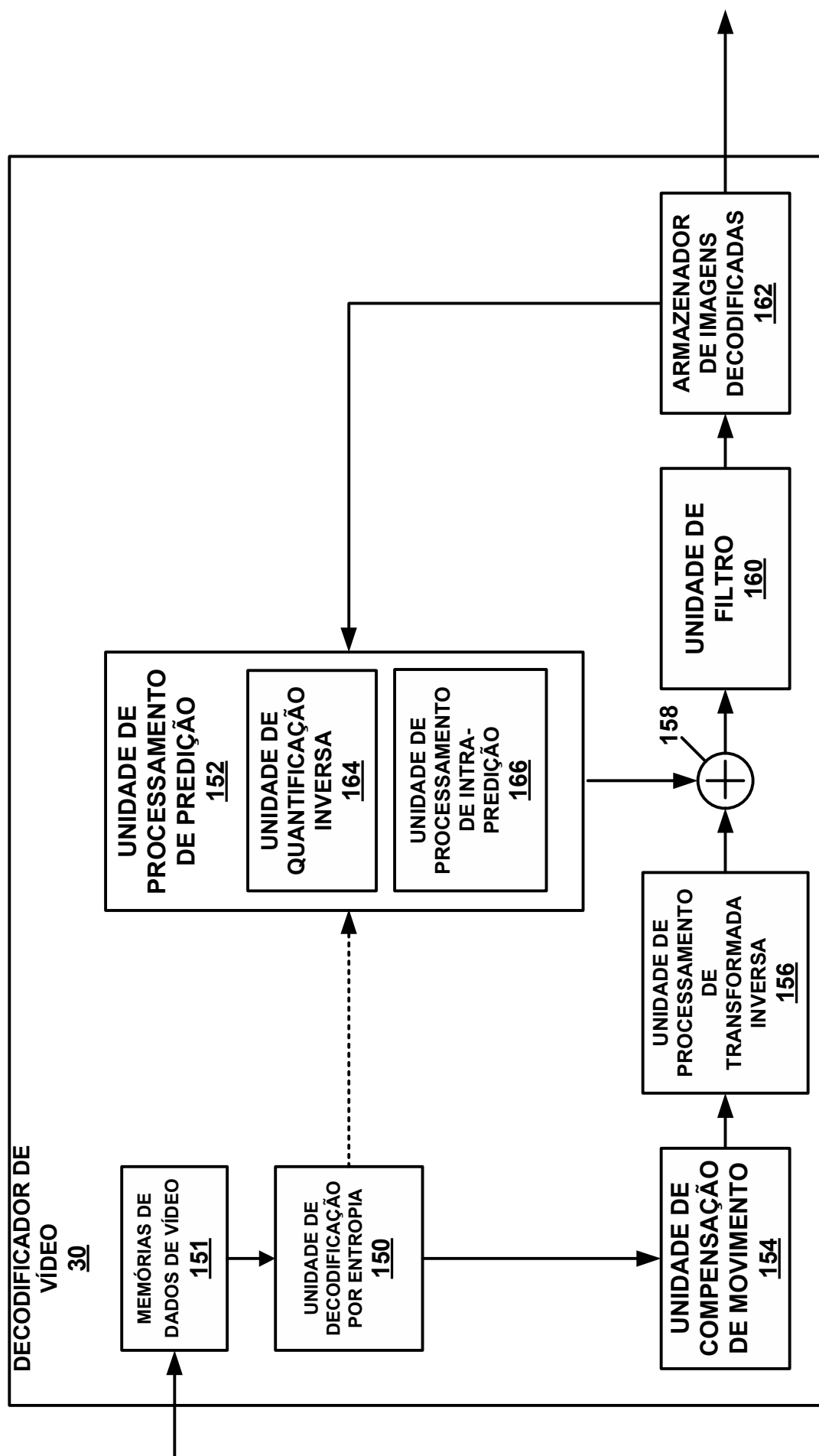


FIG. 9

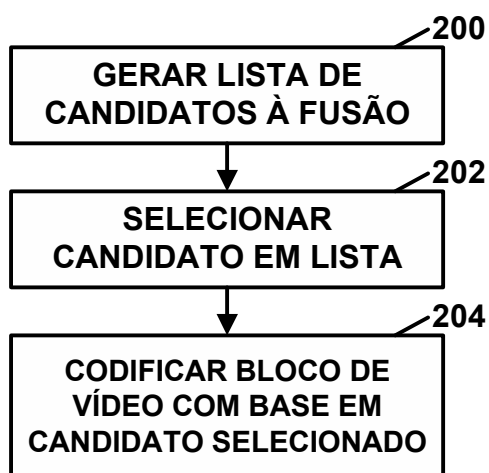


FIG. 10A

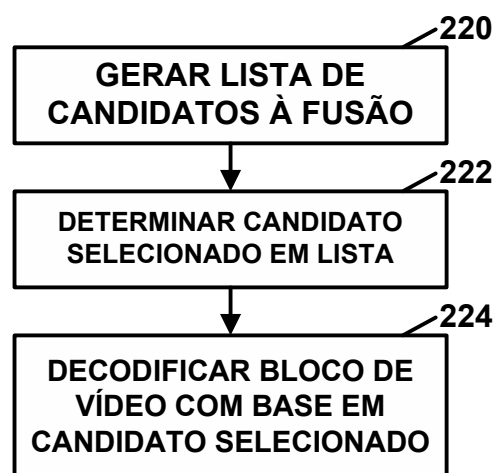


FIG. 10B

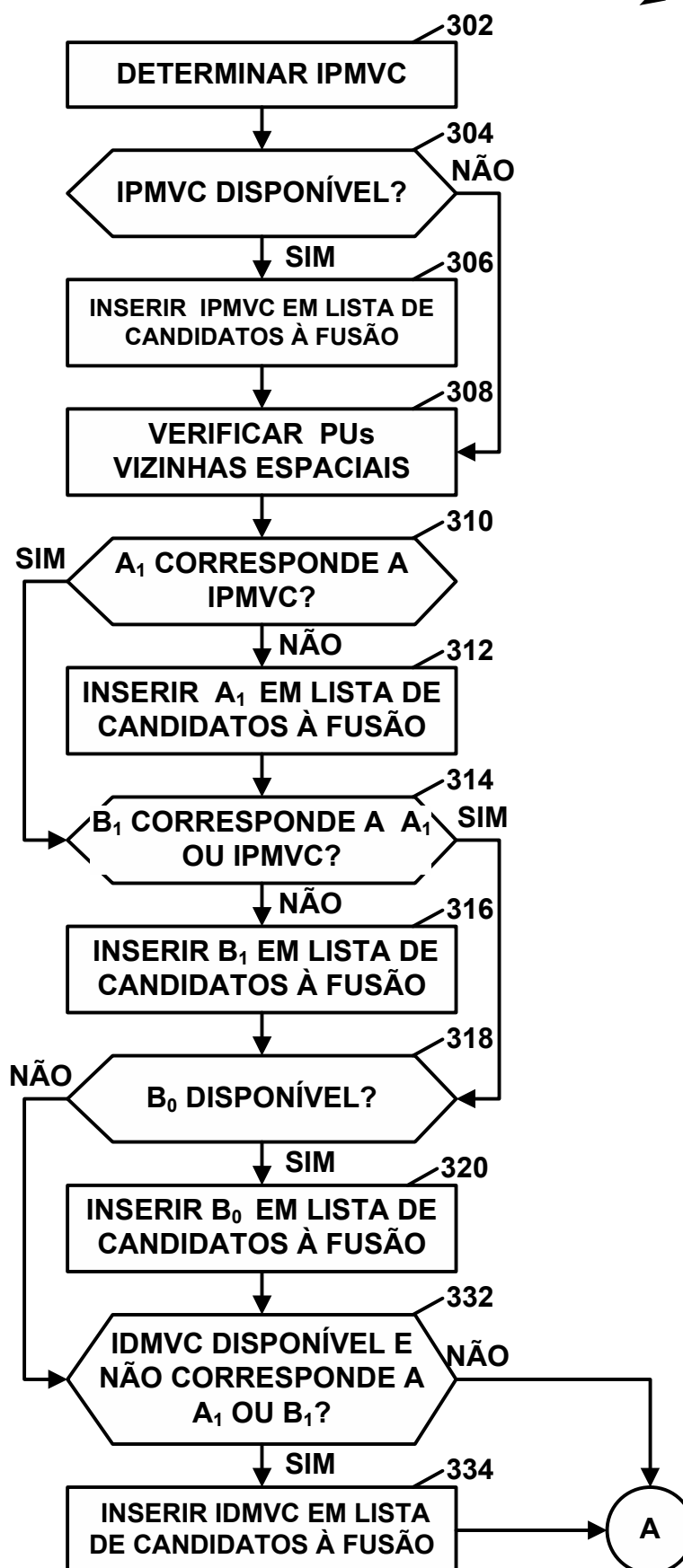


FIG. 11

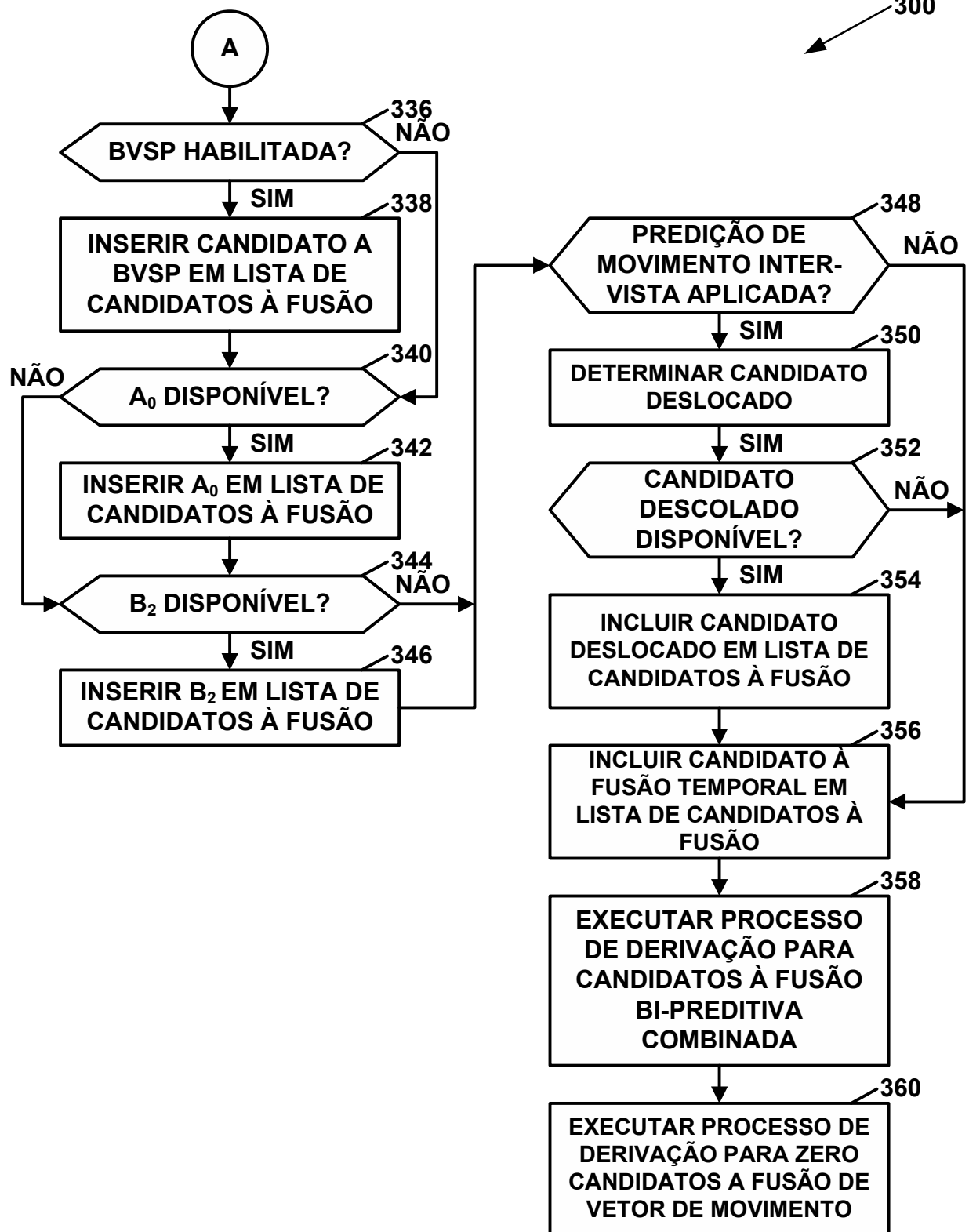


FIG. 12

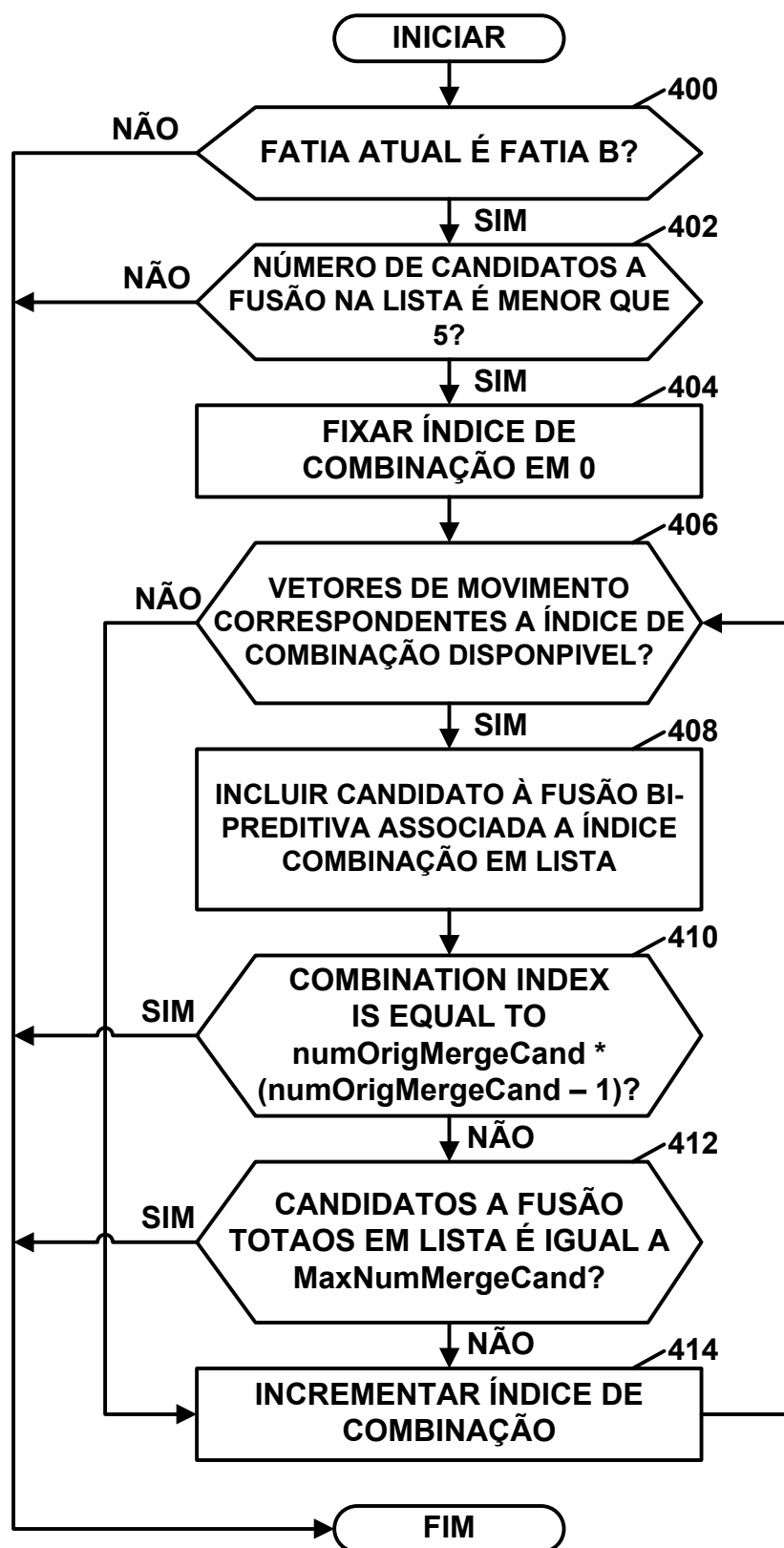


FIG. 13

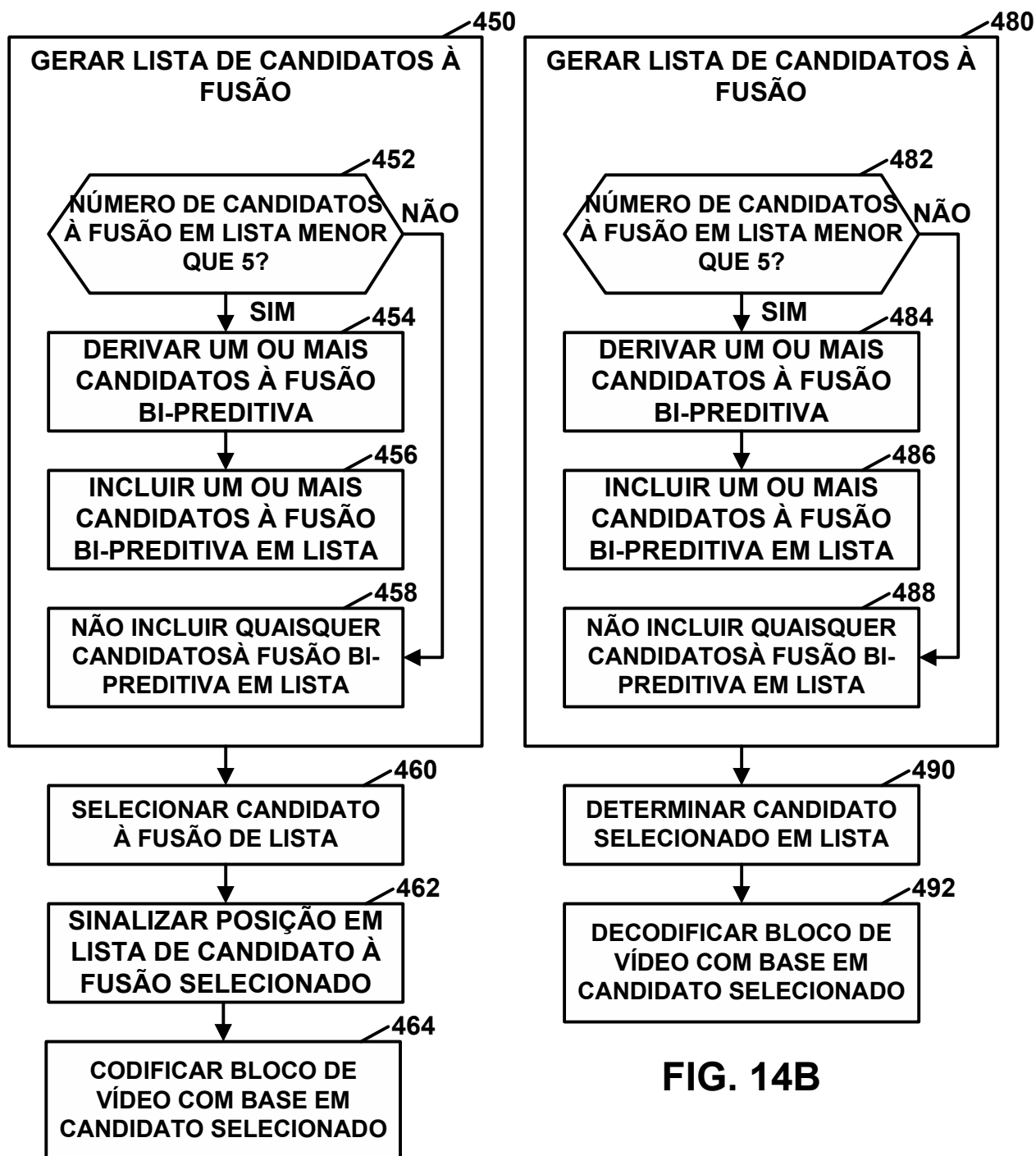


FIG. 14A

FIG. 14B

RESUMOCANDIDATOS À FUSÃO BI-PREDITIVA COMBINADA PARA
CODIFICAÇÃO DE VÍDEO 3D

Um codificador de vídeo gera uma lista de candidatos à fusão para codificar um bloco de vídeo de dados de vídeo tridimensionais (3D). O número máximo de candidatos à fusão na lista de candidatos à fusão pode ser igual a 6. Como parte da geração da lista de candidatos à fusão, o codificador de vídeo determina se o número de candidatos à fusão na lista de candidatos à fusão é menor que 5. Se este for o caso, o codificador de vídeo deriva um ou mais candidatos à fusão bi-preditiva combinada. O codificador de vídeo inclui o candidato ou candidatos à fusão bi-preditiva combinada na lista de candidatos à fusão.