



República Federativa do Brasil

Ministério do Desenvolvimento, Indústria,
Comércio e Serviços

Instituto Nacional da Propriedade Industrial

(11) BR 112016015998-5 B1

(22) Data do Depósito: 09/01/2015

(45) Data de Concessão: 14/11/2023

(54) Título: SINALIZAÇÃO DE RESOLUÇÃO DE VETOR DE MOVIMENTO ADAPTATIVA PARA CODIFICAÇÃO DE VÍDEO

(51) Int.Cl.: H04N 19/56; H04N 19/51; H04N 19/44; H04N 19/53; H04N 19/533; (...).

(30) Prioridade Unionista: 08/01/2015 US 14/592,767; 09/01/2014 US 61/925,633; 16/10/2014 US 62/064,761; 17/03/2014 US 61/954,457.

(73) Titular(es): QUALCOMM INCORPORATED.

(72) Inventor(es): XIANG LI; JOEL SOLE ROJALS; MARTA KARCZEWCZ; HONGBIN LIU; LI ZHANG; JIANLE CHEN.

(86) Pedido PCT: PCT US2015010852 de 09/01/2015

(87) Publicação PCT: WO 2015/106126 de 16/07/2015

(85) Data do Início da Fase Nacional: 08/07/2016

(57) Resumo: SINALIZAÇÃO DE RESOLUÇÃO DE VETOR DE MOVIMENTO ADAPTATIVA PARA CODIFICAÇÃO DE VÍDEO. Um dispositivo para decodificação de dados de vídeo inclui uma memória configurada para armazenar dados de vídeo e um decodificador de vídeo que compreende um ou mais processadores configurados para selecionar adaptativamente precisão de vetor de movimento para os vetores de movimento utilizados para codificar blocos de dados de vídeo.

"SINALIZAÇÃO DE RESOLUÇÃO DE VETOR DE MOVIMENTO ADAPTATIVA
PARA CODIFICAÇÃO DE VÍDEO"

[0001] Este pedido reivindica o benefício do Pedido Provisório U.S. No. 61/925.633 depositado em 9 de Janeiro de 2014, Pedido Provisório U.S. No. 61/954.457 depositado em 17 de março de 2014, e Pedido Provisório U.S. No. 62/064.761 depositado em 16 de outubro de 2014, que são aqui incorporados por referência na sua totalidade.

CAMPO TÉCNICO

[0002] Esta divulgação refere-se à codificação de vídeo e, mais particularmente, à codificação de vídeo interpretação.

FUNDAMENTOS

[0003] Capacidades de vídeo digitais podem ser incorporadas em uma ampla gama de dispositivos, incluindo televisores digitais, sistemas de broadcast diretos digitais, sistemas de broadcast sem fio, assistentes digitais pessoais (PDAs), computadores desktop ou computadores laptop, câmeras digitais, dispositivos de gravação digital, reprodutores de mídia digital, dispositivos de jogos de vídeo, consoles de jogos de vídeo, telefones de rádio celular ou via satélite, dispositivos de vídeo teleconferência, e semelhantes. Dispositivos de vídeo digital implementam as técnicas de compressão de vídeo, tais como as descritas nos padrões definidos por MPEG-2, MPEG-4, a ITU-T H.263 ou UIT-T H.264/MPEG-4, Parte 10, Codificação de Vídeo Avançada (AVC) e extensões de tais normas, para transmitir e receber informações de vídeo digital de forma mais eficiente.

[0004] As técnicas de compressão de vídeo executam predição espacial e/ou predição temporal para reduzir ou eliminar a redundância inerente em sequências de vídeo. Para codificação de vídeo baseada em bloco, um quadro de vídeo ou uma fatia pode ser dividido em macroblocos. Cada macrobloco pode ser adicionalmente particionado. Os macroblocos em um quadro intracodificado (i) ou fatia são codificados usando a predição espacial em relação ao macroblocos vizinhos. Macroblocos em um quadro intercodificado (P ou B) ou fatia podem usar predição espacial em relação aos macroblocos vizinhos no mesmo quadro ou fatia ou predição temporal em relação a outros quadros de referência.

SUMÁRIO

[0005] De um modo geral, esta divulgação descreve técnicas para selecionar adaptativamente precisão de vetor de movimento para vetores de movimento utilizados para codificar blocos de dados de vídeo e para determinar por um decodificador de vídeo mesma precisão de vetor de movimento que o codificador de vídeo selecionado para cada um dos blocos.

[0006] Em um exemplo, um método de decodificação de dados de vídeo codificado inclui determinar que um modo de codificação para um primeiro bloco é o modo de mesclagem; determinar que uma precisão de vetor de movimento para o primeiro bloco é a precisão de pixel inteiro; construir uma lista de candidato de mesclagem para o primeiro bloco, em que a lista de candidato de mesclagem compreende um candidato de vetor de movimento de precisão de fracionário; selecionar o candidato de vetor de movimento de precisão de fracionário para decodificar o primeiro bloco; arredondar o candidato de

vetor de movimento de precisão de fracionário para determinar um vetor de movimento de precisão de pixel inteiro; e localizar um bloco de referência para o primeiro bloco usando o vetor de movimento de precisão de pixel inteiro.

[0007] Em um outro exemplo, um método de codificação de dados de vídeo inclui determinar que uma precisão de vetor de movimento para um primeiro bloco é precisão de pixel inteiro; construir uma lista de candidato de mesclagem para o primeiro bloco, em que a lista de candidato de mesclagem compreende um candidato de vetor de movimento de precisão de fracionário; selecionar o candidato de vetor de movimento de precisão de fracionário para codificar o primeiro bloco; codificar o primeiro bloco usando um modo de mesclagem, arredondar o candidato de vetor de movimento de precisão de fracionário para determinar um vetor de movimento de precisão de pixel inteiro; e localizar um bloco de referência para o primeiro bloco usando o vetor de movimento de precisão de pixel inteiro.

[0008] Em um outro exemplo, um dispositivo para a decodificação de vídeo inclui uma memória configurada para armazenar dados de vídeo e um decodificador de vídeo que compreende um ou mais processadores configurados para: determinar um modo de codificação para um primeiro bloco é o modo de mesclagem; determinar uma precisão de vetor de movimento para o primeiro bloco é a precisão de pixel inteiro; construir uma lista de candidato de mesclagem para o primeiro bloco, em que a lista de candidato de mesclagem compreende um candidato de vetor de movimento de precisão de fracionário; selecionar o candidato de vetor de movimento de precisão de fracionário para decodificar o primeiro bloco;

arredondar o candidato de vetor de movimento de precisão de fracionário para determinar um vetor de movimento de precisão de pixel inteiro; e localizar um bloco de referência para o primeiro bloco usando o vetor de movimento de precisão de pixel inteiro.

[0009] Em um outro exemplo, um meio de armazenamento legível por computador que armazena instruções que, quando executadas por um ou mais processadores fazem com que um ou mais processadores: determinem um modo de codificação para um primeiro bloco é o modo de mesclagem; determinem uma precisão de vetor de movimento para o primeiro bloco é a precisão de pixel inteiro; construir uma lista de candidato de mesclagem para o primeiro bloco, em que a lista de candidato de mesclagem compreende um candidato de vetor de movimento de precisão de fracionário; selecionem o candidato de vetor de movimento de precisão de fracionário para decodificar o primeiro bloco; arredondem o candidato de vetor de movimento de precisão de fracionário para determinar um vetor de movimento de precisão de pixel inteiro; e localizem um bloco de referência para o primeiro bloco usando o vetor de movimento de precisão de pixel inteiro.

[0010] Em um outro exemplo, um aparelho para a decodificação de dados de vídeo codificado inclui meios para determinar que um modo de codificação para um primeiro bloco é o modo de mesclagem; meios para determinar que uma precisão de vetor de movimento para o primeiro bloco é a precisão de pixel inteiro; meios para construir uma lista de candidato de mesclagem para o primeiro bloco, em que a lista de mesclagem compreende um candidato de vetor de movimento de precisão de fracionário; meios para selecionar o candidato

de precisão de vetor de movimento de fracionário para decodificar o primeiro bloco; meios para arredondar o candidato de vetor de movimento de precisão de fracionário para determinar um vetor de movimento de precisão de pixel inteiro; e meios para localizar um bloco de referência para o primeiro bloco usando o vetor de movimento de precisão de pixel inteiro.

[0011] Os detalhes de uma ou mais exemplos são apresentados nos desenhos acompanhantes e na descrição abaixo. Outras características, objetos e vantagens serão evidentes a partir da descrição e desenhos, e a partir das reivindicações.

BREVE DESCRIÇÃO DOS DESENHOS

[0012] A figura 1 é um diagrama de blocos que ilustra um exemplo de codificação de vídeo e o sistema de decodificação que podem utilizar as técnicas desta divulgação para suportar adaptativa resolução de vetor de movimento.

[0013] A figura 2 é um diagrama de blocos que ilustra um exemplo de um codificador de vídeo que pode aplicar técnicas de resolução para suportar resolução de vetor de movimento adaptativa.

[0014] A figura 3 é um diagrama de blocos que ilustra um exemplo de um decodificador de vídeo, que decodifica uma sequência de vídeo codificado.

[0015] A figura 4 é um diagrama conceitual ilustrando posições de pixel fracionário para uma posição de pixel completo.

[0016] As figuras 5A-5C são diagramas conceituais que ilustram as posições de pixel de crominância e de luminância correspondentes.

[0017] A figura 6 é uma ilustração de um exemplo de combinação de modelo em forma de L para derivação de vetor de movimento do lado do decodificador (DMVD).

[0018] A figura 7 é um diagrama conceitual que ilustra uma derivação de MV bidirecional baseada em espelho exemplar.

[0019] A figura 8 é um fluxograma que ilustra um método exemplar para a codificação de dados de vídeo, de acordo com as técnicas descritas nesta divulgação.

[0020] A figura 9 é um fluxograma que ilustra um método exemplar para a decodificação de dados de vídeo, de acordo com as técnicas descritas nesta divulgação.

[0021] A figura 10 é um fluxograma que ilustra um método exemplar para a codificação de dados de vídeo.

[0022] A figura 11 é um fluxograma que ilustra um método exemplar para a codificação de dados de vídeo.

[0023] A figura 12 é um fluxograma que ilustra um método exemplar para a codificação de dados de vídeo.

[0024] A figura 13 é um fluxograma que ilustra um método exemplar para a codificação de dados de vídeo.

[0025] A figura 14 é um fluxograma que ilustra um método exemplar para a codificação de dados de vídeo.

[0026] A figura 15 é um fluxograma que ilustra um método exemplar para a codificação de dados de vídeo.

[0027] A figura 16 é um fluxograma que ilustra um método exemplar para a codificação de dados de vídeo.

[0028] A figura 17 é um fluxograma que ilustra um método exemplar para a codificação de dados de vídeo.

DESCRIÇÃO DETALHADA

[0029] De um modo geral, esta divulgação descreve técnicas para selecionar adaptativamente precisão de vetor de movimento para os vetores de movimento utilizados para codificar blocos de dados de vídeo e para determinação por um decodificador de vídeo da mesma precisão de vetor de movimento que a selecionada pelo codificador de vídeo. De acordo com algumas técnicas, o decodificador de vídeo pode derivar, sem sinalização explícita no fluxo de bits de vídeo codificado, a precisão de vetor de movimento selecionado pelo codificador de vídeo. De acordo com outras técnicas, o codificador vídeo pode sinalizar, no fluxo de bits de vídeo codificado, a precisão de vetor de movimento selecionado. As técnicas desta divulgação podem incluir, por exemplo, selecionar adaptativamente entre precisão de pixel inteiro e diferentes níveis de precisão de pixel subinteiro, por vezes referida como precisão de pixel fracionário. Por exemplo, as técnicas podem incluir selecionar adaptativamente entre precisão de pixel inteiro e precisão de um quarto de pixel ou precisão de um oitavo de pixel para vetores de movimento utilizados para codificar blocos de dados de vídeo. O termo precisão de "oitavo pixel" nesta divulgação destina-se a referir-se a precisão de um oitavo ($1/8^{\circ}$) de um pixel, por exemplo, um de: a posição de pixel completa (0/8), um oitavo de um pixel (1/8), e dois oitavos de um pixel (2/8, também um quarto de um pixel), três oitavos de um pixel (3/8), quatro oitavos de um pixel (4/8, também é uma metade de um pixel e dois quartos de um pixel), cinco

oitavos de um pixel ($5/8$), seis oitavos de um pixel ($6/8$, também três quartos de um pixel), ou sete oitavos de um pixel ($7/8$).

[0030] Codificadores e decodificadores H.264 e H.265 convencionais suportam vetores de movimento com precisão de um quarto de pixel. Tal precisão de pixel, no entanto, que não é sinalizada nem derivada, mas em vez disso, é fixa. Em alguns casos, a precisão de um oitavo de pixel pode prover certas vantagens sobre a precisão de um quarto de pixel ou a precisão de pixel inteiro. No entanto, codificar cada vetor de movimento para precisão de um oitavo de pixel pode exigir muitos bits de codificação que podem superar os benefícios de outra forma providos pelos vetores de precisão de movimento de um oitavo de um pixel. Para alguns tipos de conteúdo de vídeo, pode ser preferível codificar os vetores de movimento sem interpolação de todo, por outras palavras, utilizando apenas precisão de pixel inteiro.

[0031] O conteúdo da tela, tal como o conteúdo gerado por um computador, envolve tipicamente a série de pixels que todos têm exatamente os mesmos valores de pixel, seguido por uma alteração nítida em valores de pixel. Por exemplo, no conteúdo da tela que inclui texto azul em um fundo branco, os pixels que formam uma letra azul podem todos ter os mesmos valores de pixels, enquanto o fundo branco todo também tem os mesmos valores de pixel, mas os valores de pixels brancos podem ser significativamente diferentes dos valores de pixel azul. Conteúdo adquirido por uma câmera, por outro lado, geralmente inclui mudanças lentas em valores de pixel, devido ao movimento, sombras, mudanças de

iluminação, e outros fenômenos naturais. Como o conteúdo da tela e conteúdo adquirido de câmera normalmente têm características diferentes, codificar ferramentas eficazes para um pode não ser necessariamente eficaz para o outro. Como um exemplo, a interpolação de subpixel para codificação de interpredição pode melhorar a codificação do conteúdo da câmera, mas a complexidade associada e overhead de sinalização pode realmente reduzir codificação de qualidade e/ou eficiência de largura de banda para o conteúdo da tela.

[0032] As técnicas da presente invenção incluem de forma adaptativa a determinação de precisão de vetor de movimento com base em, por exemplo, o conteúdo do vídeo a ser codificado. Em alguns exemplos, as técnicas desta divulgação incluem derivar, por um codificador, uma precisão de vetor de movimento adequada para o conteúdo de vídeo a ser codificado. Usando as mesmas técnicas de derivação, um decodificador de vídeo também pode determinar, sem receber um elemento de sintaxe indicando a precisão de vetor de movimento, a precisão de vetor de movimento foi utilizada para codificar os dados de vídeo. Em outros exemplos, um codificador de vídeo pode ser um sinal, no fluxo de bits de vídeo codificado, a precisão de vetor de movimento selecionada pelo codificador de vídeo.

[0033] Adaptativamente selecionar precisão de vetor de movimento pode melhorar vídeo em geral de codificação de qualidade, permitindo que maiores vetores de movimento de precisão (por exemplo, $1/4^\circ$ ou $1/8^\circ$ vetores de precisão de movimento) sejam usados para conteúdo de vídeo onde o uso de tal maior precisão de vetor de movimento melhora a qualidade de codificação de vídeo, por exemplo,

através da produção de uma melhor compensação de taxa de distorção. Adaptativamente selecionar precisão de vetor de movimento também pode melhorar a qualidade geral de codificação de vídeo, possibilitando o uso de vetores de movimento de menor precisão (por exemplo, precisão de inteiro) para conteúdo de vídeo onde o uso de vetores de precisão de movimento mais elevados não melhoram, ou mesmo pioram, codificação de vídeo de qualidade.

[0034] Várias técnicas nesta descrição podem ser descritas com referência a um codificador de vídeo, que se destina a ser um termo genérico que pode referir-se a um codificador de vídeo ou a um decodificador de vídeo. A menos que expressamente indicado de outra forma, não deve presumir que as técnicas descritas em relação a um codificador de vídeo ou a um decodificador de vídeo não podem ser realizadas por outro de um codificador de vídeo ou de um decodificador de vídeo. Por exemplo, em muitos casos, um decodificador de vídeo realiza a mesma, ou às vezes uma técnica de decodificação recíproca, que um codificador de vídeo a fim de decodificar os dados de vídeo codificados. Em muitos casos, um codificador de vídeo também inclui um circuito de decodificação de vídeo, e, assim, o codificador de vídeo realiza a decodificação de vídeo como parte de codificação de dados de vídeo. Assim, a menos que indicado de outra forma, as técnicas descritas nesta divulgação no que diz respeito a um decodificador de vídeo podem também ser realizadas por um codificador de vídeo, e vice-versa.

[0035] Esta divulgação também pode usar termos como camada atual, bloco atual, imagem atual, fatia atual, etc. No contexto desta divulgação, o termo atual se destina

a identificar uma camada, bloco, imagem, fatia, etc. que está atualmente sendo codificada, ao contrário, por exemplo, camadas previamente codificadas, blocos, imagens e fatias ou ainda serem blocos codificados, imagens e fatias.

[0036] A figura 1 é um diagrama de blocos que ilustra um sistema de codificação e decodificação de vídeo exemplar 10 que pode utilizar as técnicas desta divulgação para suportar resolução adaptativa de vetor de movimento. Como mostrado na figura 1, o sistema 10 inclui um dispositivo de origem 12 que transmite codificação de vídeo para um dispositivo de destino 14 através de um canal de comunicação 16. Os dispositivo de origem 12 dispositivo de destino 14 podem compreender qualquer um de uma vasta gama de dispositivos. Em alguns casos, os dispositivo de origem 12 e o dispositivo de destino 14 podem compreender dispositivos de comunicação sem fio, tais como telefones celulares, os chamados radiotelefones celulares ou por satélite, ou quaisquer dispositivos sem fio que podem comunicar informação de vídeo através de um canal de comunicação 16, cujo caso canal de comunicação 16 é sem fio. As técnicas desta divulgação, no entanto, que dizem geralmente respeito a técnicas para suportar a precisão de subpixel adaptativa para vetores de movimento, não são necessariamente limitadas a aplicações ou configurações sem fio. Por exemplo, estas técnicas podem ser aplicadas para transmissões de televisão através do ar, transmissões de televisão por cabo, transmissões de televisão por satélite, transmissões de vídeo via Internet, vídeo digital codificado que é codificado em um meio de armazenamento, ou outros cenários. Por conseguinte, o canal de comunicação 16 pode compreender

qualquer combinação de meios de comunicação com ou sem fio apropriado para a transmissão de dados de vídeo codificados.

[0037] No exemplo da figura 1, o dispositivo de origem 12 inclui uma fonte de vídeo 18, o codificador de vídeo 20, um modulador / demodulador (modem) 22 e um dispositivo de destino 24. O dispositivo de destino 14 inclui um receptor 26, um modem 28, um decodificador de vídeo 30, e um dispositivo de exibição 32. Em conformidade com esta divulgação, o codificador de vídeo 20 do dispositivo de origem 12 pode ser configurado para aplicar as técnicas que suportam precisão de subpixel adaptativa para vetores de movimento. Em outros exemplos, um dispositivo de origem e um dispositivo de destino podem incluir outros componentes ou arranjos. Por exemplo, dispositivo de origem 12 pode receber dados de vídeo de uma fonte de vídeo externa 18, tal como uma câmara externa. Da mesma forma, o dispositivo de destino 14 pode interagir com um dispositivo externo, em vez de incluir um dispositivo de visualização integrado.

[0038] O sistema ilustrado 10 da figura 1 é meramente um exemplo. As técnicas para suportar precisão de subpixel adaptativa para vetores de movimento podem ser realizadas por qualquer de codificação de vídeo digital e/ou um dispositivo de decodificação. Embora geralmente as técnicas da presente invenção sejam realizadas por um dispositivo de codificação de vídeo, as técnicas também podem ser realizadas por um codificador de vídeo / decodificador, tipicamente referido como um "codec". Além disso, as técnicas da presente divulgação podem também ser realizadas por um pré-processador de vídeo. Dispositivo de origem 12 e o dispositivo de destino 14 são meramente exemplos de tais

dispositivos de codificação no qual o dispositivo de origem 12 gera dados de vídeo codificados para a transmissão para o dispositivo de destino 14. Em alguns exemplos, os dispositivos 12, 14 podem operar de um modo substancialmente simétrico de tal modo que cada um dos dispositivos 12, 14 incluem codificação de vídeo e componentes de decodificação. Assim, o sistema 10 pode suportar transmissão de vídeo unidirecional ou bidirecional entre dispositivos de vídeo 12, 14, por exemplo, para streaming de vídeo, reprodução de vídeo, transmissão de vídeo, ou de telefonia de vídeo.

[0039] A fonte de vídeo 18 de dispositivo de origem 12 pode incluir um dispositivo de captura de vídeo, como uma câmera de vídeo, um arquivo de vídeo contendo vídeo capturado anteriormente, e/ou uma transmissão de vídeo a partir de um provedor de conteúdo de vídeo. Como uma outra alternativa, a fonte de vídeo 18 pode gerar dados baseados em computação gráfica como o vídeo de origem, ou uma combinação de vídeo ao vivo, vídeo arquivados e vídeo gerado por computador. Em alguns casos, se a fonte de vídeo 18 é uma câmera de vídeo, dispositivo de origem 12 e dispositivo de destino 14 podem formar os chamados telefones com câmera ou telefones de vídeo. Como mencionado acima, no entanto, as técnicas descritas nesta divulgação podem ser aplicáveis à codificação de vídeo, em geral, e podem ser aplicadas a aplicações sem fio e/ou com fio. Em cada caso, o vídeo capturado, pré-capturado, ou gerado pelo computador pode ser codificado por um codificador de vídeo 20. A informação de vídeo codificado pode, então, ser modulada através de um modem 22 de acordo com um padrão de comunicação, e transmitida para o dispositivo de destino 14, através do

transmissor 24. Modem 22 pode incluir vários misturadores, filtros, amplificadores ou outros componentes concebidos para a modulação do sinal. Transmissor 24 pode incluir circuitos concebidos para a transmissão de dados, incluindo os amplificadores, filtros, e uma ou mais antenas.

[0040] O receptor 26 do dispositivo de destino 14 recebe informações através o canal 16, e modem 28 demodula a informação. Mais uma vez, o processo de codificação de vídeo pode implementar uma ou mais das técnicas aqui descritas para suportar precisão de subpixel adaptativa para vetores de movimento. As informações comunicadas ao longo do canal 16 podem incluir informações de sintaxe definidas pelo codificador de vídeo 20, que também é usado pelo decodificador de vídeo 30, que inclui elementos de sintaxe que descrevem características e/ou processamento de macroblocos e outras unidades codificadas, por exemplo, grupos de imagens (GOP). O dispositivo de exibição 32 mostra os dados de vídeo decodificados a um usuário, e pode compreender qualquer um de uma variedade de dispositivos de exibição, tais como um Tubo de raios catódicos (CRT), um display de cristal líquido (LCD), um display de plasma, um display diodo emissor de luz orgânica (OLED) ou outro tipo de dispositivo de exibição.

[0041] No exemplo da figura 1, canal de comunicação 16 pode compreender qualquer meio de comunicação com ou sem fio, como um espectro de frequência de rádio (RF) ou uma ou mais linhas de transmissão físicas, ou qualquer combinação dos meios de comunicação com e sem fio. O canal de comunicação 16 pode formar parte de uma rede baseada em pacotes, tal como uma rede de área local, uma rede de área

ampla, ou uma rede global tal como a Internet. O canal de comunicação 16 geralmente representa qualquer meio adequado de comunicação, ou coleção de diferentes mídias de comunicação, para transmitir dados de vídeo a partir do dispositivo de origem 12 para o dispositivo de destino 14, incluindo qualquer combinação adequada dos meios de comunicação com ou sem fio. O canal de comunicação 16 pode incluir roteadores, comutadores, estações base, ou qualquer outro equipamento que possa ser útil para facilitar a comunicação de dispositivo de origem 12 para o dispositivo de destino 14.

[0042] Codificador de vídeo 20 e decodificador de vídeo 30 podem operar de acordo com um padrão de compressão de vídeo, como o padrão H.264 ITU-T, alternativamente referido como MPEG-4, Parte 10, Codificação de Vídeo Avançada (AVC). As técnicas desta divulgação, no entanto, não estão limitadas a qualquer padrão de codificação particular. Outros exemplos incluem o MPEG-2 e ITU-T H.263. Embora não mostrado na figura 1, em alguns aspectos, codificador de vídeo 20 e decodificador de vídeo 30 podem cada um ser integrado com um codificador e decodificador de áudio, e pode incluir unidades apropriadas MUX-DEMUX, ou outro hardware e software, para lidar com a codificação de áudio e vídeo em um fluxo comum de dados ou fluxos de dados separados. Se as unidades aplicáveis, MUX-DEMUX podem estar de acordo com o protocolo multiplexador ITU H.223, ou outros protocolos, tais como o protocolo de Datagrama de Usuário (UDP).

[0043] O padrão ITU-T H.264 / MPEG-4 (AVC) foi formulado pelo ITU-T Video Coding Experts Group (VCEG) em

conjunto com a norma ISO/IEC Moving Picture Experts Group (MPEG) como o produto de uma parceria coletiva conhecida como o Joint Video Team (JVT). Em alguns aspectos, as técnicas descritas nesta divulgação podem ser aplicadas a dispositivos que geralmente se conformam com a norma H.264. A norma H.264 é descrita em ITU-T Recommendation H.264, Codificação de Vídeo Avançada para serviços audiovisuais gerais, pelo Grupo de Estudo ITU-T, e datada de Março de 2005, que pode ser aqui referida como a norma H.264 ou especificação H.264, ou o padrão H.264/AVC ou especificação. O Joint Video Team (JVT) continua a trabalhar em extensões para H.264/AVC, e desenvolver novos padrões, por exemplo, para HEVC.

[0044] O codificador de vídeo 20 e decodificador de vídeo 30 cada um pode ser implementado como qualquer um de uma variedade de circuitos de codificador adequado, tal como um ou mais microprocessadores, processadores de sinais digitais (DSPs), circuito integrado de aplicação específica (ASIC), arranjo de porta programável em campo (FPGAs), lógica discreta, software, hardware, firmware ou quaisquer suas combinações. Cada um codificador de vídeo 20 e decodificador de vídeo 30 pode ser incluído em um ou mais codificadores ou decodificadores, cada um dos quais pode ser integrado como parte de um codificador / decodificador combinado (codec) de uma respectiva câmara, computador, aparelho móvel, dispositivo de assinante, dispositivo de broadcast, set-top box, servidor ou similares.

[0045] Uma sequência de vídeo inclui tipicamente uma série de quadros de vídeo. Um grupo de imagens (GOP) compreende geralmente uma série de um ou mais quadros de

vídeo. Um GOP pode incluir dados de sintaxe de um cabeçalho de GOP, um cabeçalho de um ou mais quadros de GOP, ou qualquer outro local, que descreve uma série de quadros incluídos no GOP. Cada quadro pode incluir dados de sintaxe de quadro que descreve um modo de codificação para o respectivo quadro. O codificador de vídeo 20 opera tipicamente em blocos de vídeo dentro de quadros de vídeo individuais de modo a codificar os dados de vídeo. Um bloco de vídeo pode corresponder a um macrobloco ou um macrobloco de uma partição. Os blocos de vídeo podem ter tamanhos fixos ou variados, e podem ser diferentes em tamanho de acordo com um padrão de codificação especificado. Cada quadro de vídeo pode incluir uma pluralidade de fatias. Cada fatia pode incluir uma pluralidade de macroblocos, que podem ser arranjados em partições, também referida como sub-blocos.

[0046] Como exemplo, a norma ITU-T H.264 suporta intrapredição em vários tamanhos de bloco, tais como 16 por 16, 8 por 8, ou 4 por 4, para componentes de luminância, e 8x8 para componentes de croma, bem como interpredição em vários tamanhos de bloco, como 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 e 4x4 para componentes de luminância e tamanhos escalados para componentes de croma correspondentes. Nesta divulgação, "NxN" e "N por N" podem ser utilizados indiferentemente para se referirem às dimensões de pixel do bloco em termos de dimensões vertical e horizontal, por exemplo, 16x16 pixels ou 16 por 16 pixels. Em geral, um bloco de 16x16 terá 16 pixels na direção vertical ($Y = 16$) e 16 elementos na direção horizontal ($X = 16$). Do mesmo modo, um bloco de NxN geralmente tem N pixels na direção vertical e N pixels na direção horizontal, em que N representa um valor

inteiro positivo. Os pixels de um bloco podem ser dispostos em filas e colunas. Além disso, os blocos não têm necessariamente de ter o mesmo número de pixels na direção horizontal que na direção vertical. Por exemplo, os blocos podem compreender, $N \times M$ pixels em que M não é necessariamente igual a N .

[0047] Tamanhos de blocos que são menores de 16 por 16 podem ser referidos como partições de 16 por 16 macrobloco. Blocos de vídeo podem compreender blocos de dados de pixel no domínio do pixel, ou blocos de coeficientes de transformada no domínio da transformada, por exemplo, após a aplicação de uma transformada, tais como uma transformada de cosseno discreta (DCT), transformada de número inteiro, uma transformada de wavelet, ou uma transformada conceitualmente similar para os dados do bloco de vídeo residuais representando diferenças de pixel entre os blocos de vídeo codificados e blocos de vídeo preditivos. Em alguns casos, um bloco de vídeo pode compreender blocos de coeficientes de transformada quantizados, no domínio da transformada.

[0048] Blocos de vídeo menores podem prover uma melhor resolução, e podem ser utilizados para localização de um quadro de vídeo que inclui altos níveis de detalhe. Em geral, os macroblocos e as várias partições, por vezes referidas como sub-blocos, podem ser considerados blocos de vídeo. Além disso, uma fatia pode ser considerada uma pluralidade de blocos de vídeo, tais como macroblocos e/ou sub-blocos. Cada fatia pode ser uma unidade independentemente decodificável de um quadro de vídeo. Alternativamente, quadros por si só podem ser unidades

decodificáveis, ou outras partes de um quadro podem ser definidas como unidades decodificáveis. O termo "unidade codificada" pode referir-se a qualquer unidade independentemente decodificável de um quadro de vídeo, tal como um quadro inteiro, uma fatia de um quadro, um grupo de imagens (GOP), também referida como uma sequência, ou outra unidade de forma independente decodificável definida de acordo com as técnicas de codificação aplicáveis.

[0049] Uma nova codificação de vídeo padrão, referida como Codificação de Vídeo de Alta Eficiência (HEVC), foi recentemente finalizada. Esforços estão em andamento para desenvolver várias extensões para HEVC, incluindo uma extensão referida como a extensão de Codificação de Conteúdo de Tela. Os esforços de normalização HEVC basearam-se em um modelo de um dispositivo de codificação de vídeo referido como o modelo de teste HEVC (HM). O HM pressupõe várias capacidades de dispositivos de codificação de vídeo mais dispositivos de acordo com, por exemplo, ITU-T H.264/AVC. Por exemplo, enquanto H.264 provê nove modos de codificação intrapredição, HM provê até trinta e três modos de codificação de intrapredição.

[0050] HM refere-se a um bloco de dados de vídeo como uma unidade de codificação (CU). Sintaxe de dados dentro de um fluxo de bits pode definir uma maior unidade de codificação (LCU), que é uma unidade de codificação de maior em termos de número de pixels. Em geral, uma CU tem um efeito similar a um macrobloco de H.264, exceto que uma CU não tem uma distinção de tamanho. Assim, uma CU pode ser dividida em sub-CUs. Em geral, as referências da presente divulgação a uma CU pode referir-se a uma maior unidade de codificação de

uma imagem ou de uma sub-CU de uma LCU. Uma LCU pode ser dividida em sub-CUs, e cada sub-CU pode ser dividida em sub-CUs. Dados de sintaxe de um fluxo de bits podem definir um número máximo de vezes que uma LCU pode ser dividida, referida como a profundidade da CU. Por conseguinte, um fluxo de bits pode também definir uma unidade de codificação menor (SCU). Esta divulgação também usa o termo "bloco" para se referir a qualquer uma de uma CU, PU, ou TU. Além disso, quando esta divulgação refere-se a exemplos envolvendo uma unidade de codificação ou CU, deve ser entendido que outros exemplos podem ser providos em relação a macroblocos substituídos para as unidades de codificação.

[0051] Uma LCU pode ser associada com uma estrutura de dados quadtree. Em geral, uma estrutura de dados inclui um nó quadtree por CU, onde um nó raiz corresponde ao LCU. Se uma CU é dividida em quatro sub-CU, o nó correspondente a CU inclui quatro nós de folhas, cada um dos quais corresponde a uma das sub-CUs. Cada nó da estrutura de dados quadtree pode prover dados de sintaxe para a CU correspondente. Por exemplo, um nó no quadtree pode incluir um flag de divisão, que indica se a CU correspondente ao nó é dividida em sub-CUs. Elementos de sintaxe para uma CU pode ser definida recursivamente, e podem depender do fato de a CU ser dividida em sub-CUs.

[0052] Uma CU que não é dividida (por exemplo, correspondente a um nó de folha na estrutura de dados quadtree) pode incluir uma ou mais unidades de predição (PUs). Em geral, uma PU representa toda ou uma porção da CU correspondente, e inclui dados para obter uma amostra de referência para a PU. Por exemplo, quando a PU é codificada

em modo intra, a PU pode incluir dados que descrevem um modo de intrapredição para a PU. Como outro exemplo, quando a PU é codificada em modo inter, a PU pode incluir dados que definem um vetor de movimento para a PU. Os dados que definem o vetor de movimento podem descrever, por exemplo, um componente horizontal do vetor de movimento, um componente vertical do vetor de movimento, uma resolução para o vetor de movimento (por exemplo, precisão de pixel inteiro, precisão de um quarto de pixel, precisão de um oitavo de pixel), um quadro de referência para o qual o vetor de movimento aponta, e/ou uma lista de referência (por exemplo, da lista 0 ou lista 1) para o vetor de movimento. Os dados para a CU definindo a PU (s) também podem descrever, por exemplo, particionamento da CU em uma ou mais PUs. Modos de particionamento podem diferir entre se a CU é não codificada, codificada em modo de intrapredição, ou codificada em modo de interpredição.

[0053] Uma CU tendo uma ou mais PUs também pode incluir uma ou mais unidades de transformada (TU). Após predição usando uma PU, um codificador de vídeo pode calcular um valor residual para a porção da CU correspondente à PU. O valor residual pode ser transformado, quantizado e digitalizado. Uma TU não é necessariamente limitada ao tamanho de uma PU. Assim, UTs podem ser maiores ou menores do que PUs correspondentes à mesma CU. Em alguns exemplos, o tamanho máximo de uma TU pode corresponder ao tamanho da CU, que inclui a TU.

[0054] De acordo com as técnicas desta divulgação, o codificador de vídeo 20 pode codificar em modo inter uma CU utilizando uma ou mais PUs tendo vetores de movimento de

precisão de pixel subinteiro variada. Por exemplo, o codificador de vídeo 20 pode selecionar entre utilizar um vetor de movimento tendo precisão de pixel inteiro ou precisão de pixel fracionário (por exemplo, um quarto ou um oitavo) para uma PU com base no conteúdo dos dados de vídeo sendo codificados. De acordo com algumas técnicas desta divulgação, um codificador de vídeo 20 pode não necessitar gerar, por inclusão no fluxo de bits dos dados de vídeo codificados, uma indicação da precisão de subpixel para um vetor de movimento de uma PU. Em vez disso, decodificador de vídeo 30 pode derivar a precisão de vetor de movimento utilizando as mesmas técnicas de derivação usadas pelo codificador de vídeo 20. De acordo com outras técnicas desta divulgação, o codificador de vídeo 20 pode incluir, no fluxo de bits de dados de vídeo codificados, um ou mais elementos de sintaxe que o decodificador de vídeo 30 pode usar para determinar a precisão de vetor de movimento selecionado.

[0055] Para calcular valores para posições de pixel subinteiros, codificador de vídeo 20 pode incluir uma variedade de filtros de interpolação. Por exemplo, interpolação bilinear pode ser utilizada para calcular os valores para posições de pixel subinteiros. O codificador de vídeo 20 pode ser configurado para executar uma busca de movimento em relação a dados de luminância de uma PU para calcular um vetor de movimento utilizando os dados de luminância de PU. O codificador de vídeo 20 pode então reutilizar o vetor de movimento para codificar os dados de crominância da PU. Normalmente, os dados de crominância tem uma resolução mais baixa do que os dados de luminância correspondentes, por exemplo, um quarto da resolução dos

dados de luminância. Portanto, o vetor de movimento para dados de crominância pode ter uma precisão mais elevada do que para os dados de luminância. Por exemplo, vetores de movimento de precisão de um quarto de pixel para dados de luminância podem possuir uma precisão de um oitavo de pixel para os dados de crominância. Da mesma forma, vetores de movimento de precisão de um oitavo de pixel para dados de luminância podem ter precisão de um dezesseis avos de pixel para dados de crominância.

[0056] Após a codificação intrapreditiva ou interpredictiva para produzir dados preditivos e dados residuais, e após qualquer transformada (tal como a transformada de número inteiro 4x4 ou 8x8 utilizada em H.264/AVC ou uma transformada de cosseno discreta DCT) para produzir coeficientes de transformada, quantização de coeficientes de transformada pode ser realizada. Quantização geralmente refere-se a um processo em que os coeficientes de transformada são quantizados para possivelmente reduzir a quantidade de dados utilizados para representar os coeficientes. O processo de quantização pode reduzir a profundidade do bit associado com alguns ou todos os coeficientes. Por exemplo, um valor de n-bit pode ser arredondado para baixo para um valor de m-bit durante quantização, em que n é maior do que m.

[0057] Após quantização, codificação de entropia dos dados quantizados pode ser realizada, por exemplo, de acordo com a codificação de comprimento variável adaptativa de conteúdo (CAVLC), codificação aritmética binária adaptativa de contexto (CABAC), ou outra metodologia de codificação de entropia. Uma unidade de processamento

configurada para codificação de entropia, ou outra unidade de processamento, pode exercer outras funções de processamento, tal como codificação de comprimento de zero execução de coeficientes quantizados e/ou geração de informação de sintaxe, tal como valores padrão de bloco codificado (CBP), tipo de macrobloco, modo de codificação, tamanho LCU, ou semelhantes.

[0058] O decodificador de vídeo 30 do dispositivo de destino 14 pode ser configurado para executar a técnicas semelhantes, e geralmente simétricas, a qualquer uma ou todas as técnicas de codificador de vídeo 20 desta divulgação. Por exemplo, um decodificador de vídeo 30 pode ser configurado para receber informações definindo um contexto em que uma indicação de uma precisão de subpixel para um vetor de movimento de uma PU de uma CU foi codificada. O codificador de vídeo 20 pode prover, e decodificador de vídeo 30 pode receber as informações de contexto em um quadtree para uma LCU incluindo a CU e a PU. A informação de contexto pode corresponder à informação para o tamanho de CU e/ou da PU, por exemplo, uma profundidade da CU, um tamanho de PU, e/ou um tipo para a PU. Decodificador de vídeo 30 pode usar as informações de contexto para decodificar a indicação da precisão de subpixel do vetor de movimento, por exemplo, para determinar se o vetor de movimento tem precisão de um quarto de pixel ou precisão de um oitavo de pixel. Por exemplo, decodificador de vídeo 30 pode realizar um processo de codificação de entropia inversa utilizando a informação de contexto para decodificar por entropia a indicação da precisão de subpixel para o vetor de movimento.

[0059] O codificador de vídeo 20 e o decodificador de vídeo 30 cada um pode ser implementado como qualquer um de uma variedade de conjunto de circuitos de codificador ou decodificador adequado, conforme o caso, tal como um ou mais microprocessadores, processadores de sinais digitais (DSPs), circuitos integrados de aplicação específica (ASIC), arranjos de porta programáveis em campo (FPGA), conjunto de circuitos de lógica discreta, software, hardware, firmware ou quaisquer suas combinações. Cada um do codificador de vídeo 20 e decodificador de vídeo 30 pode ser incluído em um ou mais codificadores ou decodificadores, cada um dos quais pode ser integrado como parte de um codificador de vídeo / decodificador combinado (codec). Um aparelho, incluindo um codificador de vídeo 20 e/ou decodificador de vídeo 30 pode compreender um circuito integrado, um microprocessador, e/ou um dispositivo de comunicação sem fio, tal como um telefone celular.

[0060] A figura 2 é um diagrama de blocos que ilustra um exemplo de codificador de vídeo 20 que pode implementar técnicas para suportar resolução de vetor de movimento adaptativa. O codificador de vídeo 20 pode realizar intra e interpredição de blocos dentro de quadros de vídeo, incluindo TUs, CUs, e PUs e calcular valores residuais que podem ser codificados como TUs. Intracodificação se baseia na predição espacial para reduzir ou eliminar a redundância espacial em vídeo dentro de um determinado quadro de vídeo. Intercodificação se baseia em predição temporal para reduzir ou eliminar a redundância temporal em vídeo dentro de quadros adjacentes de uma sequência de vídeo. Modo intra (I-modo) pode se referir a qualquer um dos vários modos de compressão

especiais e modos inter, tais como predição unidirecional (modo P) ou predição bidirecional (modo B) pode se referir a qualquer um dos vários modos de compressão temporais. Unidade de estimação de movimento 42 e unidade de compensação de movimento 44 pode executar a codificação de interpredição, enquanto a unidade de intrapredição 46 pode executar a codificação de intrapredição.

[0061] Como mostrado na figura 2, codificador de vídeo 20 recebe um bloco de vídeo atual dentro de um quadro de vídeo a ser codificado. No exemplo da figura 2, codificador de vídeo 20 inclui memória de vídeo de dados 38, a unidade de compensação de movimento 44, a unidade de estimação de movimento 42, a unidade de intrapredição 46, armazenador de imagem decodificada 64, somador 50, unidade de transformada 52, unidade de quantização 54 e unidade de codificação de entropia 56. Para reconstrução do bloco de vídeo, um codificador de vídeo 20 também inclui a unidade de quantização inversa 58, unidade de transformada inversa 60, e o somador 62. Um filtro de desbloqueio (não mostrado na figura 2) pode também ser incluído para filtrar limites de bloco para remover artefatos de bloqueio de vídeo reconstruído. Se desejado, o filtro de desbloqueio que tipicamente filtra a saída do somador 62.

[0062] Memória de dados de vídeo 38 pode armazenar os dados de vídeo a serem codificados pelos componentes do codificador de vídeo 20. Os dados de vídeo armazenados na memória de dados de vídeo 38 pode ser obtidos, por exemplo, a partir de fonte de vídeo 18. O armazenador de imagem decodificada 64 pode ser uma memória de imagem de referência, que armazena dados de referência de vídeo para utilização na

codificação de dados de vídeo por um codificador de vídeo 20, por exemplo, em modos intra ou intercodificação. A memória de dados de vídeo 38 e armazenador de imagem decodificada 64 pode ser formada por qualquer um de uma variedade de dispositivos de memória, tais como a memória dinâmica de acesso aleatório (DRAM), incluindo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM), ou outros tipos de dispositivos de memória. A memória de dados de vídeo 38 e armazenador de imagem decodificada 64 podem ser providos pelo mesmo dispositivo de memória ou dispositivos de memória separados. Em vários exemplos, memória de dados de vídeo 38 pode ser em chip com outros componentes de um codificador de vídeo 20, ou fora do chip em relação a esses componentes.

[0063] Durante o processo de codificação, codificador de vídeo 20 recebe um quadro de vídeo ou uma fatia a ser codificada. O quadro ou fatia pode ser dividido em vários blocos de vídeo (por exemplo, LCUs). A unidade de estimação de movimento 42 e a unidade de compensação de movimento 44 executam codificação interpreditiva do bloco de vídeo recebido relativo a um ou mais blocos em um ou mais quadros de referência para prover compressão temporal. A unidade de intrapredição 46 pode realizar a codificação intrapreditiva do bloco de vídeo recebido em relação a um ou mais blocos vizinhos no mesmo quadro ou uma fatia como o bloco de ser codificada para prover compressão espacial.

[0064] A Unidade de seleção de modo 40 pode selecionar um dos modos de codificação, intra ou inter, por exemplo, com base em resultados de erro, e provê o bloco intra ou intercodificado resultante para o somador 50 para

gerar dados de bloco residuais e para o somador 62 para reconstruir o bloco codificado para o uso como um quadro de referência. Quando a unidade de seleção de modo 40 seleciona a codificação de modo inter para um bloco, a unidade de seleção de resolução 48 pode selecionar uma resolução para um vetor de movimento para o bloco. Por exemplo, a unidade de seleção de resolução 48 pode selecionar a precisão de um oitavo de pixel ou a precisão de um quarto de pixel para um vetor de movimento para o bloco.

[0065] Como um exemplo, a unidade de seleção de resolução 48 pode ser configurada para comparar uma diferença de erro entre o uso de vetor de movimento de precisão de um quarto de pixel para codificar um bloco e utilizar um vetor de movimento de precisão de um oitavo pixel para codificar o bloco. A unidade de estimação de movimento 42 pode ser configurada para codificar um bloco que utilize um ou mais vetores de movimento de precisão de quarto de pixel em uma primeira passagem de codificação e um ou mais vetores de movimento de precisão de um oitavo de pixel em uma segunda passagem de codificação. A unidade de estimação de movimento 42 pode usar ainda uma variedade de combinações de um ou mais vetores de movimento de precisão de um quarto de pixel e um ou mais vetores de movimento de precisão de um oitavo de pixel para o bloco em uma terceira passagem de codificação. A unidade de seleção de resolução 48 pode calcular os valores de taxa de distorção para cada passagem de codificação do bloco e calcular as diferenças entre os valores de taxa de distorção.

[0066] Quando a diferença excede um limite, a unidade de seleção de resolução 48 pode selecionar a vetor

de movimento de precisão de um oitavo de pixel para codificar o bloco. Unidade de seleção de resolução 48 pode também avaliar a informação de taxa de distorção, analisar um orçamento de bit, e/ou analisar outros fatores para determinar se deve usar a precisão de um oitavo de pixel ou precisão de um quarto de pixel para um vetor de movimento quando se codifica um bloco durante um processo de predição de modo inter. Depois de selecionar a precisão de um oitavo de pixel ou a precisão de um quarto de pixel para um bloco a ser codificado em modo inter, a unidade de seleção de modo 40 ou estimação de movimento pode enviar uma mensagem (por exemplo, um sinal) à unidade de estimação de movimento 42 indicativa da precisão selecionada para um vetor de movimento.

[0067] A unidade de estimação de movimento 42 e a unidade de compensação de movimento 44 podem ser altamente integradas, mas são ilustradas separadamente para fins conceituais. A estimação de movimento é o processo de geração de vetores de movimento, que estimam o movimento para blocos de vídeo. Um vetor de movimento, por exemplo, pode indicar o deslocamento de um bloco preditivo dentro de um quadro de referência preditivo (ou outra unidade codificada) em relação ao bloco atual sendo codificado dentro do quadro atual (ou outra unidade de codificação). Um bloco preditivo é um bloco que se encontra em correspondência ao bloco a ser codificado, em termos de diferença de pixels, que pode ser determinado pela soma da diferença absoluta (SAD), soma de diferença quadrada (SSD), ou outras métricas de diferença. Um vetor de movimento, também pode indicar o deslocamento de uma partição de um macrobloco. A compensação de movimento

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

[0068] A unidade de estimação de movimento 42 calcula um vetor de movimento para o bloco de vídeo de um quadro intercodificado, comparando o bloco de vídeo com blocos de vídeo de um quadro de referência na unidade de compensação de movimento 64. O armazenador de imagem decodificada 44 também pode interpolar pixels subinteiros do quadro de referência, por exemplo, um quadro I ou quadro P. A norma ITU H.264, como um exemplo, descreve duas listas: lista 0, que inclui quadros de referência que têm uma ordem de exibição anterior do que um quadro atual a ser codificado, e lista 1, que inclui quadros de referência que têm uma ordem de exibição posterior do que o quadro atual que sendo codificado. Portanto, os dados armazenados no armazenador de imagem decodificada 64 podem ser organizados de acordo com essas listas.

[0069] De acordo com as técnicas desta divulgação, a unidade de compensação de movimento 44 pode ser configurada para interpolar os valores para as posições de um dezesseis avos de pixel de dados de crominância de uma CU quando um vetor de movimento para dados de luminância da CU têm precisão de um oitavo de pixel. Interpolar valores para as posições de um dezesseis avos de pixel dos dados de crominância, a unidade de compensação de movimento 44 pode utilizar a interpolação bilinear. Portanto, somador 50 pode calcular um residual para os dados de crominância da CU em

relação aos valores bilineares interpolados de posições de um dezesseis avos de pixel de um bloco de referência. Deste modo, o codificador de vídeo 20 pode calcular, usando interpolação bilinear, valores de posições de um dezesseis avos de pixel de dados de crominância de um bloco de referência identificado por um vetor de movimento e codificar dados de crominância de uma unidade de codificação com base nos valores bilineares interpolados do bloco de referência, quando os dados de luminância da unidade de codificação forem codificados utilizando um vetor de movimento tendo precisão de um oitavo de pixel para os dados de luminância.

[0070] A unidade de estimação de movimento 42 compara blocos de um ou mais quadros de referência do armazenador de imagem decodificada 64 com um bloco sendo codificado de um quadro atual, por exemplo, um quadro P ou quadro B. Quando os quadros de referência no armazenador de imagem decodificada 64 incluem valores de subpixels inteiros, um vetor de movimento calculado pela unidade de estimação de movimento 42 pode referir-se a uma localização de pixel subinteiro de um quadro de referência. A unidade de estimação de movimento 42 e/ou unidade de compensação de movimento 44 podem também ser configuradas para calcular os valores para as posições de pixel subinteiros de quadros de referência armazenados no armazenador de imagem decodificada 64. Se nenhum dos valores para as posições de pixel subinteiros são armazenados no armazenador de imagem decodificada 64. A unidade de estimação de movimento 42 envia o vetor de movimento calculado para unidade de codificação de entropia 56 e unidade de compensação de movimento 44. O

bloco de quadro de referência identificado por um vetor de movimento pode ser referido como um bloco preditivo.

[0071] A unidade de estimação de movimento 42, a unidade de compensação de movimento 44, a unidade de seleção de modo 40, ou outra unidade do codificador de vídeo 20, também pode sinalizar o uso de precisão de um quarto de pixel ou precisão de um oitavo de pixel para um vetor de movimento usado para codificar um bloco. Por exemplo, a unidade de estimação de movimento 42 pode enviar uma indicação de um pixel de precisão subinteiro para o vetor de movimento para unidade de codificação de entropia 56. A unidade de estimação de movimento 42 também pode prover informação de contexto relativo à informação de tamanho para uma PU correspondente ao vetor de movimento para unidade de codificação de entropia 56, onde as informações de tamanho podem incluir qualquer um ou todos de uma profundidade de uma CU incluindo a PU, um tamanho de PU, e/ou um tipo de codificação para a PU.

[0072] A unidade de compensação de movimento 44 pode calcular dados de predição com base no bloco de predição. O codificador de vídeo 20 forma um bloco de vídeo residual subtraindo os dados de predição de unidade de compensação de movimento 44 a partir do bloco de vídeo original que está sendo codificado. Somador 50 representa o componente ou componentes que executam esta operação de subtração. Unidade de transformada 52 aplica uma transformada, tal como uma transformada de cosseno discreta (DCT), ou uma transformada conceitualmente semelhante, ao bloco residual, produzindo um bloco de vídeo compreendendo valores de coeficiente de transformada residual.

[0073] A unidade de transformada 52 pode realizar outras transformadas, tais como as definidas pelo padrão H.264, que são conceitualmente semelhantes a DCT. Wavelet Transformada de Wavelet, transformada de inteiro, transformadas de sub-bandas ou outros tipos de transformadas também poderiam ser utilizadas. Em qualquer caso, unidade de transformada 52 aplica a transformada ao bloco residual, produzindo um bloco de coeficientes de transformada residuais. A transformada pode converter a informação residual a partir de um domínio de valores de pixel para um domínio da transformada, tal como um domínio de frequência. A unidade de quantização 54 quantifica os coeficientes de transformada residual para reduzir ainda mais a taxa de bits. O processo de quantização pode reduzir a profundidade do bit associado com alguns ou todos os coeficientes. O grau de quantização pode ser modificado por ajuste de um parâmetro de quantificação.

[0074] Depois de quantização, unidade de codificação de entropia 56 codifica por entropia coeficientes de transformada quantizados. Por exemplo, unidade de codificação de entropia 56 pode executar CAVLC, CABAC de conteúdo, ou outra técnica de codificação de entropia. Após a codificação de entropia pela unidade de codificação de entropia 56, o vídeo codificado pode ser transmitido para um outro dispositivo ou arquivado para posterior transmissão ou recuperação. No caso de codificação aritmética binária adaptativa de contexto, o contexto pode ser baseado em macroblocos vizinhos.

[0075] Em alguns casos, a unidade de codificação de entropia 56 ou outra unidade de um codificador de vídeo

20 pode ser configurada para executar outras funções de codificação, além da codificação de entropia. Por exemplo, a unidade de codificação de entropia 56 pode ser configurada para determinar os valores de CBP para os macroblocos e partições. Além disso, em alguns casos, a unidade de codificação de entropia 56 pode executar codificação de comprimento de execução dos coeficientes em um macrobloco ou partição da mesma. Em particular, a unidade de codificação de entropia 56 pode aplicar uma varredura zig-zag ou outro padrão de varredura para varrer os coeficientes de transformada em um macrobloco ou partição e codificar execução de zeros para uma maior compressão. A unidade de codificação de entropia 56 pode também construir informações de cabeçalho com elementos de sintaxe apropriados para a transmissão no fluxo de bits de vídeo codificado.

[0076] De acordo com as técnicas desta divulgação, nos casos em que a precisão de subpixel é sinalizada ao invés de derivada, a unidade de codificação de entropia 56 pode ser configurada para codificar uma indicação de uma precisão de subpixel para um vetor de movimento, por exemplo, para indicar se o vetor de movimento tem precisão de pixel inteiro ou tem precisão de subpixel, tais como a precisão de um quarto pixel ou precisão de um oitavo de pixel (ou outras precisões de subpixel, em vários exemplos). A unidade de codificação de entropia 56 pode codificar a indicação usando CABAC. Além disso, a unidade de codificação de entropia 56 pode utilizar informação de contexto para a realização CABAC para codificar a indicação que indica informações de tamanho para uma PU correspondente ao vetor de movimento, em que as informações de tamanho podem incluir qualquer um ou todos de

uma profundidade de uma CU incluindo a PU, um tamanho da PU, e/ou um tipo para a PU.

[0077] O codificador de vídeo 20 pode preditivamente sinalizar o vetor de movimento. Dois exemplos de técnicas de sinalização de predição que podem ser implementados pelo codificador de vídeo 20 incluem predição do vetor de movimento avançado (AMVP) e sinalização de modo de mesclagem. Em AMVP, codificador de vídeo 20 e decodificador de vídeo 30 ambos montam listas de candidatos com base em vetores de movimento determinados a partir de blocos já codificados. O codificador de vídeo 20, em seguida, sinaliza um índice para a lista de candidato para identificar um preditor de vetor de movimento (MVP) e sinaliza uma diferença de vetor de movimento (MVD). O decodificador de vídeo 30 interpreta um bloco usando o MVP, modificado pelo MVD, por exemplo, usando um vetor de movimento igual a MVP + MVD.

[0078] No modo de mesclagem, codificador de vídeo 20 e decodificador de vídeo 30 ambos montam uma lista de candidato com base em blocos já codificados, e codificador de vídeo 20 sinaliza um índice para um dos candidatos na lista de candidato. No modo de mesclagem, decodificador de vídeo 30 interpreta o bloco atual utilizando o vetor de movimento e o índice de imagem de referência do candidato sinalizado. Em ambos AMVP e modo de mesclagem, codificador de vídeo 20 e decodificador de vídeo 30 utilizam as mesmas técnicas de construção de lista, de modo que a lista usada pelo codificador de vídeo 20 ao determinar como codificar um bloco corresponde à lista usada pelo decodificador de vídeo 30 ao determinar como decodificar o bloco.

[0079] A unidade de quantização inversa 58 e unidade de transformada inversa 60 aplica quantização inversa e transformada inversa, respectivamente, para reconstruir o bloco residual no domínio do pixel, por exemplo, para utilização posterior como um bloco de referência. A unidade de compensação de movimento 44 pode calcular um bloco de referência, adicionando o bloco residual a um bloco preditivo de um dos quadros de armazenador de imagem decodificada 64. A unidade de compensação de movimento 44 também pode aplicar um ou mais filtros de interpolação ao bloco residual reconstruído para calcular valores de pixel subinteiro para uso na estimação de movimento. Somador 62 adiciona o bloco residual reconstruído para o movimento do bloco de predição compensado produzido pela unidade de compensação de movimento 44 para produzir um bloco de vídeo reconstruído para armazenamento no armazenador de imagem decodificada 64. O bloco de vídeo reconstruído pode ser usado pela unidade de estimação de movimento 42 e a unidade de compensação de movimento 44 como um bloco de referência para intercodificar um bloco em um quadro de vídeo subsequente.

[0080] A figura 3 é um diagrama de blocos que ilustra um exemplo do decodificador de vídeo 30, que decodifica uma sequência de vídeo codificado. No exemplo da figura 3, decodificador de vídeo 30 inclui uma unidade de decodificação de entropia 70, unidade de compensação de movimento 72, unidade de intrapredição 74, unidade de quantização inversa 76, unidade de transformada inversa 78, armazenador de imagem decodificada 82, e somador 80. O decodificador de vídeo 30 pode, em alguns exemplos, executar uma passagem de decodificação recíproca geralmente à

passagem de codificação descrita em relação ao codificador de vídeo 20 (figura 2). A unidade de compensação de movimento 72 pode gerar dados de predição baseado em vetores de movimento recebidos da unidade de decodificação de entropia 70.

[0081] A memória de dados de vídeo 68 pode armazenar os dados de vídeo, tal como um fluxo de bits de vídeo codificado, para ser decodificado pelos componentes do decodificador de vídeo 30. Os dados de vídeo armazenados na memória de dados de vídeo 68 podem ser obtidos, por exemplo, a partir do meio legível por computador 16, por exemplo, a partir de uma fonte de vídeo local, tal como uma câmera, através de comunicação em rede com ou sem fio de dados de vídeo, ou acessando meios de armazenamento de dados físico. Memória de dados de vídeo 68 pode formar um armazenador de imagem codificada (CEC) que armazena os dados codificados de vídeo a partir de um fluxo de bits de vídeo codificado. O armazenador de imagem decodificada 82 pode ser uma memória de imagem de referência que armazena dados de referência de vídeo para utilização na decodificação de dados de vídeo pelo decodificador de vídeo 30, por exemplo, nos modos de intra ou intercodificação. A memória de dados de vídeo 68 e armazenador de imagem decodificada 82 podem ser formados por qualquer um de uma variedade de dispositivos de memória, tais como a memória dinâmica de acesso aleatório (DRAM), incluindo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM), ou outros tipos de dispositivos de memória. A memória de dados de vídeo 68 e o armazenador de imagem decodificada 82 podem ser providos pelo mesmo dispositivo de memória ou dispositivos de memória separados.

Em vários exemplos, memória de dados de vídeo 68 pode ser em chip com outros componentes do decodificador de vídeo 30, ou fora do chip em relação a esses componentes.

[0082] A unidade de decodificação de entropia 70 pode recuperar um fluxo de bits codificado, por exemplo, a partir de memória de dados de vídeo 68. O fluxo de bits codificado pode incluir dados de vídeo codificados por entropia. A unidade de decodificação de entropia 70 pode decodificar os dados de vídeo codificados por entropia, e dos dados de vídeo decodificados por entropia, unidade de compensação de movimento 72 pode determinar informação de movimento, incluindo os vetores de movimento, precisão de vetor de movimento, índices de lista de imagem de referência, e outra informação de movimento. A unidade de compensação de movimento 72 pode, por exemplo, determinar tal informação realizando AMVP e técnicas de modo de mesclagem descrito acima.

[0083] A unidade de compensação de movimento 72 pode utilizar os vetores de movimento no fluxo de bits recebidos para identificar um bloco de predição em quadros de referência no armazenador de imagem decodificada 82. A unidade de intrapredição 74 pode utilizar modos de intrapredição recebidos no fluxo de bits para formar um bloco de predição dos blocos espacialmente adjacentes. A unidade de quantização inversa 76 quantiza inversamente, isto é, dequantifica os coeficientes quantizados de blocos providos no fluxo de bits e decodificados pela unidade de decodificação de entropia 70. O processo de quantização inversa pode incluir um processo convencional, por exemplo, tal como definido pela norma de decodificação H.264. O

processo de quantização inversa também pode incluir a utilização de um parâmetro de quantificação QPY calculado por um codificador de vídeo 20, para cada macrobloco, para determinar um grau de quantização e, do mesmo modo, um grau de quantização inversa, que deve ser aplicado.

[0084] A unidade de transformada inversa 58 aplica uma transformada inversa, por exemplo, uma transformada discreta de cosseno inversa, uma transformada de inteira inversa, ou uma transformada inversa conceitualmente semelhante, para os coeficientes de transformada, a fim de produzir blocos residuais no domínio do pixel. A unidade de compensação de movimento 72 produz blocos de movimento compensado, possivelmente realizando interpolação com base em filtros de interpolação. Identificadores para filtros de interpolação para serem usados para a estimação de movimento com precisão de subpixel podem ser integrados nos elementos de sintaxe. A unidade de compensação de movimento 72 pode usar filtros de interpolação como usado pelo codificador de vídeo 20 durante a codificação do bloco de vídeo para calcular os valores interpolados para pixels subinteiros de um bloco de referência. A unidade de compensação de movimento 72 pode determinar os filtros de interpolação utilizados pelo codificador de vídeo 20 de acordo com informações de sintaxe recebidas e usar os filtros de interpolação para produzir blocos preditivos. De acordo com as técnicas desta divulgação, a unidade de compensação de movimento 72 pode interpolar valores das posições de um dezesseis avos de pixel de dados de crominância de um bloco de referência quando um vetor de movimento tem precisão de um oitavo de pixel para dados de luminância. Por exemplo, a unidade de compensação

de movimento 72 pode utilizar interpolação bilinear para interpolar os valores das posições de um dezesseis avos de pixel do bloco de referência.

[0085] A unidade de compensação de movimento 72 utiliza algumas das informações de sintaxe para determinar tamanhos de LCUs e CUs usados para codificar quadro (s) e/ou fatia (s) da sequência de vídeo codificado, informações de partição que descreve como cada macrobloco de um quadro da sequência de vídeo codificado é particionado, modos indicando como cada partição é codificada, um ou mais quadros de referência (e listas de quadro de referência) para cada CU intercodificada, e outras informações para decodificar a sequência de vídeos codificada.

[0086] Somador 80 soma os blocos residuais com os blocos de predição correspondentes gerados pela unidade de compensação de movimento 72 ou unidade de intrapredição para formar blocos decodificados. Se desejado, um filtro de desbloqueio pode também ser aplicado para filtrar os blocos decodificados, a fim de remover artefatos de bloqueio. Os blocos de vídeo decodificados são então armazenados no armazenador de imagem decodificada 82, que provê os blocos de referência de compensação de movimento subsequente e também produz vídeo decodificado para apresentação em um dispositivo de exibição (tais como dispositivo de exibição 32 da figura 1).

[0087] A figura 4 é um diagrama conceitual ilustrando posições de pixel fracionário para uma posição de pixel completa. Em particular, a figura 4 ilustra posições de pixel fracionário de pixel completo (pel) 100. Pixels completos 100 correspondem a posições de meio pixel 102A-

102C (metade de pels 102), posições de um quarto de pixel 104A-104L (um quarto de pels 104), e as posições de um oitavo de um pixel 106A-106 AV (um oitavo de pels 106).

[0088] A figura 4 ilustra posições de um oitavo de pixel 106 de um bloco usando potinhos delineando para indicar que estas posições podem ser opcionalmente incluídas. Isto é, se um vetor de movimento de precisão tem um oitavo de pixel, o vetor de movimento pode apontar para qualquer posição de pixel completo 100, posições de metade do pixel 102, posições de quartos de pixel 104, ou posições de um oitavo de pixel 106. No entanto, se o vetor de movimento de precisão tem um quarto de pixel, o vetor de movimento pode apontar para qualquer posição de pixel total 100, posições de meio pixel 102, ou posições de um quarto de pixel 104, mas não seria apontar para posições de um oitavo de pixel 106. Além disso, deverá ser entendido que, em outros exemplos, podem ser utilizadas outras precisões, por exemplo, uma precisão de pixel-XVI, uma precisão de um trinta e dois avos de pixel, ou similar.

[0089] Um valor para o pixel na posição de pixel completo 100 pode ser incluído em um quadro de referência correspondente. Isto é, o valor para o pixel na posição de pixel completo 100 geralmente corresponde ao valor real de um pixel no quadro de referência, por exemplo, que finalmente renderizado e exibido quando o quadro de referência é apresentado. Os valores para posições de meio pixel 102, posições de um quarto de pixel 104, e posições de um oitavo de pixel 106 (coletivamente como referidas posições de pixel fracionário) podem ser interpolados utilizando filtros adaptativos de interpolação ou filtros de interpolação

fixos, por exemplo, filtros de vários números de "derivações" (coeficientes) tal como vários filtros de Wiener, filtros bilineares, ou outros filtros. Em geral, o valor de uma posição de pixel fracionário pode ser interpolado a partir de um ou mais pixels vizinhos, que correspondem aos valores das posições de pixel completos vizinhos ou posições de pixel fracionário previamente determinado.

[0090] De acordo com as técnicas desta divulgação, um codificador de vídeo, tal como um codificador de vídeo 20, pode adaptativamente selecionar precisão para um vetor de movimento, por exemplo, entre precisão de pixel inteiro ou precisão de pixel fracionário tal como precisão de um oitavo de pixel e uma precisão de um quarto de pixel. O codificador de vídeo 20 pode fazer esta seleção para cada vetor de movimento, cada CU, cada LCU, cada fatia, cada quadro, cada GOP, ou outras unidades de dados de vídeo codificados. Quando codificador de vídeo 20 seleciona precisão de um quarto de pixel para um vetor de movimento, o vetor de movimento pode se referir a qualquer uma posição total de pixels 100, posições de meio pixel 102, ou posições de um quarto de pixel 104. Quando codificador de vídeo 20 seleciona precisão de um oitavo de pixel para um vetor de movimento, o vetor de movimento pode se referir a qualquer uma posição de pixel total 100, posições de meio pixel 102, posições de um quarto de pixel 104, ou posições de um oitavo de pixel 106.

[0091] Figura 5A-5C são diagramas que ilustram as posições de crominância conceituais e pixel de luminância correspondentes. As figuras 5A-5C ilustram também como vetores de movimento calculados para os dados de luminância

podem ser reutilizados para dados de crominância. Por uma questão preliminar, as figuras 5A-5C ilustram uma fila parcial de posições de pixel. Deve entender-se que, na prática, uma posição de pixel completa pode ter uma grade retangular de posições de pixel fracionário associadas, tais como a ilustrada na figura 4. O exemplo das figuras 5A-5C destina-se a ilustrar os conceitos descritos nesta divulgação, e não pretende ser uma lista exaustiva de correspondências entre as posições de pixel de crominância de fracionários e posições de pixel de luminância de fracionários.

[0092] As figuras 5A-5C ilustram posições de pixel de um bloco de luminância, incluindo a posição de pixel completo de luminância 110, posição de meio pixel de luminância 116, posição de um quarto de pixel 112, e as posições de um oitavo de pixel de luminância 114A, 114B. As figuras 5A-5C ilustram também posições de pixels correspondentes de um bloco de crominância, incluindo a posição de pixel completo de crominância 120, posição de um quarto de pixel de crominância 122, posição de um oitavo de pixel de crominância 124, e posições de dezesseis avos de pixel de crominância 126A, 126B. Neste exemplo, pixel de crominância completo 120 corresponde ao pixel de luminância completo 110. Além disso, neste exemplo, o bloco de crominância será reduzido por um fator de dois horizontalmente e verticalmente em relação ao bloco de luminância. Assim, um quarto de pixel de crominância 122 corresponde à metade do pixel de luminância 116. Da mesma forma, um oitavo de pixel de crominância 124 corresponde a um quarto de pixel de luminância 112, dezesseis avos de pixel

de crominância 126A corresponde a um oitavo de pixel de luminância 114A e dezesseis avos pixel de crominância 126B corresponde a um oitavo de pixel de luminância 114B.

[0093] Em codecs de vídeo avançados, tais como H.264/AVC, HEVC e, potencialmente, os codecs sucessores H.264 e HEVC, o custo de bits de vetores de movimento de sinalização pode aumentar. Para diminuir este custo de bit, derivação de MV do lado do decodificador (DMVD) pode ser utilizada. Em S. Kamp e M. Wien, "Decoder-side motion vector derivation for block-based vídeo coding" *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, pp. 1732 -1745, dezembro de 2012, DMVD foi proposta com base em uma correspondência de modelo em forma de L.

[0094] A figura 6 é uma ilustração de um exemplo de correspondência de modelo em forma de L para DMVD. No exemplo da figura 6, bloco atual 132 de imagem atual 134 é interpretado utilizando correspondência de modelos. Modelo 136 define uma forma que cobre blocos vizinhos já decodificados do bloco atual 132. Um decodificador de vídeo (por exemplo, decodificador de vídeo 30) pode, por exemplo, primeiro comparar os valores de pixels incluídos nos blocos vizinhos já decodificados cobertos pelo modelo 136 com valores de pixel incluídos nos blocos vizinhos já decodificados cobertos pelo modelo colocalizado 138, que cobre blocos localizados em um quadro de referência de imagens de referência 140. O decodificador de vídeo pode então mover o modelo para outros locais na imagem de referência e comparar os valores de pixel cobertos pelo modelo com os valores dos pixels incluídos nos blocos vizinhos já decodificados cobertos por modelo 136.

[0095] Com base nessas comparações múltiplas, o decodificador de vídeo pode determinar a melhor correspondência, tal como melhor correspondência 142 mostrada no exemplo da figura 6. O decodificador de vídeo pode, então, determinar um deslocamento entre a melhor correspondência e o modelo colocalizado. Este deslocamento (por exemplo, deslocamento 144 na figura 6) corresponde ao vetor de movimento utilizado para predizer o bloco atual 132.

[0096] Tal como ilustrado na figura 6, quando um bloco é codificado no modo DMVD, MV para o bloco é buscado pelo decodificador de vídeo 30, em vez de ser diretamente sinalizado para decodificador de vídeo 30. O MV que leva à distorção mínima por correspondência de modelo é selecionado como o MV final para o bloco. Para manter elevada a eficiência da codificação, pode ser necessário para o decodificador 30 selecionar vetor de movimento candidato como o MV para decodificar o bloco atual, o que pode aumentar a complexidade da decodificação.

[0097] Para reduzir complexidade de decodificação em DMVD, um método de derivação MV bidirecional baseado em espelho foi proposto em Y.-J. Chiu, L. Xu, W. Zhang, H. Jiang, "DECODER-SIDE MOTION ESTIMATION AND WIENER FILTER FOR HEVC", oficina VCIP 2013, Malásia, 17-20 de novembro de 2013.

[0098] A figura 7 é um diagrama conceitual que ilustra uma derivação MV bidirecional baseada em espelho exemplo. Tal como ilustrado na figura 7, derivação MV bidirecional baseada em espelho pode ser aplicada por estimação de movimento centro-simétrica em torno de centros de buscas na precisão da amostra fracionada no lado do

decodificador. O tamanho / localização da janela de busca pode ser predefinido e sinalizado no fluxo de bits. Na figura 7, DMV é um deslocamento que é adicionado ao PMV0 e é subtraído de PMV1 para gerar um par de MV, MV0 e MV1. Todos os valores de DMV dentro de uma janela de busca podem ser verificados e a soma das diferenças absolutas (SAD) entre a referência de L0 e os blocos de referência de L1 pode ser utilizado como a medição de estimação de movimento centro-simétrica. Um par de MV com um SAD mínimo pode ser selecionado como os MVs finais para o bloco.

[0099] Para resolução de vetor de movimento adaptativo, compensação de movimento de subpixel pode normalmente ser muito mais eficiente do que a compensação de movimento de pixel inteiro. Por algum conteúdo, no entanto, como a textura com frequência muito alta ou o conteúdo da tela, compensação de movimento de subpixel pode não ter um melhor desempenho ou, em alguns casos ainda pior desempenho. Em tais casos, pode ser melhor apenas para ter MVs com precisão de pixel inteiro.

[0100] Tal como descrito em L. Guo, P. Yin, Y. Zheng, X. Lu, Q. Xu, J. Sole, "Adaptive motion vector resolution with implicit signaling", ICIP 2010: 2057-2060, resolução MV adaptativa foi proposta com base em resíduos reconstruídos. Quando a variância do bloco de resíduo reconstruído está acima de um limite, precisão de um quarto de pixel de vetor de movimento é usada. Caso contrário, a precisão de meio pixel de vetor de movimento se aplica. Conforme descrito em J. An, X. Li, X. Guo, S. Lei, "Progressive MV Resolution", JCTVC-F125, Torino, Itália, julho 2011, resolução MV é adaptável determinada com base na

magnitude de diferença MV sinalizada. Conforme descrito em Y. Zhou, B. Li, J. Xu, GJ Sullivan, B. Lin, "Motion Vector Resolution COntrol for Screen Content Coding", JCTVC-P0277, San Jose, EUA, janeiro de 2014, informações de precisão de vetor de movimento são sinalizadas ao nível da fatia.

[0101] Até agora, métodos de precisão de vetor de derivação do lado do decodificador não têm provado serem muito eficientes, especialmente para a codificação de conteúdo de tela. Além disso, nenhum único método de derivação de precisão de vetor de movimento do lado do decodificador tem provado ser eficaz para todos os tipos de conteúdo, incluindo conteúdo adquirido por câmera, conteúdo de tela, e outros tipos de conteúdo. Além disso, os métodos de precisão de vetor de movimento adaptativos até agora provaram não serem eficientes para codificação de conteúdo da tela.

[0102] Em alguns exemplos, implementando as técnicas descritas nesta divulgação, um método de derivação de precisão de vetor de movimento do lado do decodificador é proposto para o conteúdo de tela. Neste exemplo, precisão de vetor de movimento pode ser dependente do resultado da correspondência de modelo do lado do decodificador. Quando um resultado de correspondência de modelo de uma posição de pixel inteiro e que desta posição de subpixel vizinho são bastante diferentes, a região em causa pode ser considerada como o conteúdo da tela e MV com precisão de inteiro pixel deve ser usado. Caso contrário, precisão de vetor de movimento de subpixel é usada. Para definir "bastante diferente", um ou mais limites fixos ou adaptativos podem ser usados.

[0103] O decodificador de vídeo 30 pode, por exemplo, decodificar dados de vídeo por determinação de uma precisão de vetor de movimento com base na correspondência de modelo. Em tal exemplo, decodificador de vídeo 30 pode, para um bloco atual a ser codificado, identificar uma posição de pixel inteiro de um bloco vizinho já codificado e, com base em uma localização da posição de pixel inteiro, aplicar um modelo para determinar uma pluralidade de posições de pixel. Decodificador de vídeo 30 pode também aplicar o modelo a uma pluralidade de posições de subpixel para determinar uma pluralidade de posições de subpixels. O modelo pode, por exemplo, definir uma forma, e decodificador de vídeo 30 pode aplicar o modelo de dados de vídeo para determinar a pluralidade de posições de pixel inteiro localizando a pluralidade de posições de pixel inteiro com base em uma localização da forma em relação ao bloco atual. Da mesma forma, decodificador de vídeo 30 pode aplicar o modelo aos dados de vídeo para determinar a pluralidade de posições de subpixel, localizando a pluralidade de posições de pixel de subpixels com base em uma localização da forma em relação ao bloco atual.

[0104] O decodificador de vídeo 30 pode comparar um ou mais valores de pixel para a pluralidade de posições de pixel inteiro com um ou mais valores de pixel para a pluralidade de posições de subpixel e, com base na comparação, determinar uma precisão de vetor de movimento de um vetor de movimento. Decodificador de vídeo 30 pode decodificar o bloco atual utilizando o vetor de movimento. O decodificador de vídeo 30 pode, por exemplo, determinar o

vetor de movimento utilizando um modo de mesclagem, um modo de AMVP, ou algum outro tal modo.

[0105] O decodificador de vídeo 30 pode determinar a precisão de vetor de movimento para o vetor de movimento através da comparação de um ou mais valores de pixel para a pluralidade de posições de pixel inteiro com os um ou mais valores de pixel para a pluralidade de posições de subpixel para determinar um valor de diferença que corresponde a uma quantidade de diferença de valores de pixel entre um ou mais valores de pixel para a pluralidade de posições de pixel inteiro e os um ou mais valores de pixel para a pluralidade de posições de subpixel. Em resposta ao valor de diferença sendo maior do que um valor limite, decodificador de vídeo 30 determina a precisão de vetor de movimento é a precisão de pixel inteiro. Em resposta ao valor de diferença sendo inferior a um valor limite, decodificador de vídeo 30 pode determinar a precisão de vetor de movimento para ser precisão de subpixel. O valor limite pode ser um valor fixo, um valor adaptativo, ou algum outro tipo de valor. Para comparar os um ou mais valores de pixel para a pluralidade de posições de pixel inteiro com os um ou mais valores de pixel para a pluralidade de posições de subpixels, decodificador de vídeo 30 pode, por exemplo, determinar uma soma das diferenças absolutas entre os um ou mais valores de pixel para a pluralidade de posições de pixel inteiro e os um ou mais valores de pixel para a pluralidade de posições de subpixels.

[0106] De acordo com outras técnicas desta divulgação, a precisão de vetor de movimento pode ser dependente das propriedades (tais como exatidão, gradiente, ou se transformada é ignorada) de bloco (s) espacialmente

vizinho, bloco (s) temporalmente vizinho, ou ambos. A informação de precisão de vetor de movimento pode ser derivada no lado do decodificador. Alternativamente ou adicionalmente, a precisão de vetor de movimento pode ser dependente da precisão de vetor de movimento de blocos espacialmente vizinhos, blocos temporalmente vizinhos, ou ambos.

[0107] O decodificador de vídeo 30 pode, por exemplo, determinar uma precisão de vetor de movimento com base em propriedades de blocos vizinhos. Os blocos vizinhos podem, por exemplo, incluir pelo menos um bloco espacialmente vizinho e/ou pelo menos um bloco temporalmente vizinho. Para um bloco atual sendo codificado, decodificador de vídeo 30 pode localizar um ou mais blocos vizinhos e determinar uma propriedade de um ou mais blocos vizinhos. A propriedade pode, por exemplo, ser uma ou mais de uma exatidão dos um ou mais blocos vizinhos, uma exatidão de um ou mais blocos vizinhos, se um ou mais blocos vizinhos foram codificados em um modo de pulso, e/ou uma precisão de vetor de movimento dos um ou mais blocos vizinhos. Com base na propriedade de um ou mais blocos vizinhos, decodificador de vídeo 30 pode determinar uma precisão de vetor de movimento de um vetor de movimento e decodificar o bloco atual utilizando o vetor de movimento. Decodificador de vídeo 30 pode, por exemplo, determinar, sem sinalização (por exemplo, com base em um contexto) que a propriedade ou propriedades para determinar, sempre podem determinar uma propriedade ou propriedades fixas, ou podem receber uma indicação de tais propriedade ou propriedades de determinar.

[0108] Em uma outra técnica exemplar desta divulgação, um indicador no qual o método ou métodos de precisão de vetor de movimento do lado do decodificador são utilizados pode ser sinalizado no fluxo de bits. Por exemplo, o indicador pode ser sinalizado diretamente no fluxo de bits ou derivado de outras informações codificadas no fluxo de bits, tal como o tipo de fatia e nível temporal.

[0109] O decodificador de vídeo 30 pode, por exemplo, receber em um fluxo de bits de vídeo codificado, uma indicação de um tipo de sinalização de precisão de vetor de movimento e, com base no tipo de sinalização de precisão de vetor de movimento, determinar uma precisão de vetor de movimento de um bloco de dados de vídeo. O decodificador de vídeo 30 pode usar um vetor de movimento da precisão de vetor de movimento determinada para localizar um bloco de referência para o bloco de dados de vídeo. O tipo de sinalização de precisão de vetor de movimento pode, por exemplo, ser um de (1) um tipo de correspondência de modelo, tal como descrito acima, (2) um tipo baseado em propriedade de bloco vizinho, tal como descrito acima, ou (3) um tipo de sinalização direta como será descrito em mais detalhe abaixo.

[0110] O decodificador de vídeo 30 pode, por exemplo, receber a indicação de um cabeçalho de fatia, um SPS, um PPS, ou em algum outro nível. A indicação pode, por exemplo, ser um tipo de fatia. Em outras palavras, decodificador de vídeo 30 pode determinar um tipo de fatia para uma fatia particular e, com base nesse tipo de fatia, pode determinar uma precisão de vetor de movimento a usar para blocos de decodificação dessa fatia. A indicação pode, por exemplo, ser um nível temporal de uma fatia. Em outras

palavras, decodificador de vídeo 30 pode determinar um nível temporal, para uma fatia e, com base no nível temporal da fatia determinar uma precisão de vetor de movimento para ser usado para os blocos de decodificação da fatia.

[0111] Em um outro exemplo, a informação de precisão de vetor de movimento pode ser sinalizada no fluxo de bits, tal como no nível de LCU de unidade de codificação maior, nível de CU ou o nível de PU. Em outras palavras, o codificador de vídeo 20 pode gerar um ou mais elementos de sintaxe para inclusão no fluxo de bits de dados de vídeo codificados, e decodificador de vídeo 30 pode analisar os elementos de sintaxe para determinar a precisão de vetor de movimento de um determinado bloco de dados de vídeo. Quando uma CU é indicada para ter MVs de precisão de inteiro, todas as PUs dentro desta CU tem precisão de vetor de movimento de inteiro.

[0112] Em um exemplo, por modo de mesclagem / pulo, decodificador de vídeo 30 pode arredondar um vetor de movimento para uma precisão de inteiro somente para realizar a compensação de movimento. O MV não arredondado pode ser salvo para a predição de MV de blocos posteriores. Por exemplo, decodificador de vídeo 30 pode determinar um modo de codificação para um primeiro bloco é um modo de mesclagem ou um modo de pulo e determinar uma precisão de vetor de movimento para o primeiro bloco é precisão de pixel inteiro. O decodificador de vídeo 30 pode construir uma lista de candidato de mesclagem para o primeiro bloco que inclui, pelo menos, um candidato de vetor de movimento de precisão de fracionário. Decodificador de vídeo 30 pode selecionar o candidato de vetor de movimento de precisão de fracionário

para decodificar o primeiro bloco e arredondar a o candidato de vetor de movimento de precisão de fracionário para determinar um vetor de movimento de precisão de pixel inteiro. Decodificador de vídeo 30 pode localizar um bloco de referência para o primeiro bloco usando o vetor de movimento de precisão de pixel inteiro.

[0113] Para um segundo bloco (por exemplo, um bloco codificado com base nas informações do primeiro bloco), decodificador de vídeo 30 pode adicionar o candidato de vetor de movimento de precisão de inteiro a uma lista de candidato (por exemplo, uma lista de candidato de mesclagem ou uma lista de candidato AMVP) para o segundo bloco. Em outros exemplos, no entanto, decodificador de vídeo 30 pode adicionar o candidato de vetor de movimento de precisão de fracionário a uma lista de candidato para um segundo bloco.

[0114] Para o modo inter de não mesclagem / pulo, preditores de MV podem ser arredondados para precisão de inteiro, e MVD pode ser sinalizado na precisão de inteiro de modo que MV arredondado pode ser salvo para a predição de MV de bloco posterior. Alternativamente ou em adição a, MV arredondado pode ser salvo em predição de MV de bloco posterior. Em um exemplo, para este caso, o arredondamento pode ser realizado apenas para compensação de movimento apenas. Alternativamente ou em adição a, MV arredondado pode ser utilizado em compensação de movimento e pode ser gravado para a predição de MV de bloco posterior.

[0115] Por exemplo, decodificador de vídeo 30 pode determinar um modo de codificação para um primeiro bloco é diferente de um modo de mesclagem e determinar uma precisão de vetor de movimento para o primeiro bloco é precisão de

pixel inteiro. Decodificador de vídeo 30 pode determinar uma precisão de MVP de fracionário para o primeiro bloco e arredondar a precisão de MVP de fracionário para determinar uma precisão de MVP de pixel inteiro para o primeiro bloco. Decodificador de vídeo 30 pode determinar um MVD para o primeiro bloco que é precisão de pixel inteiro. Decodificador de vídeo 30 pode determinar um vetor de movimento de precisão de pixel inteiro com base na precisão de MVP de pixel inteiro e a precisão de MVD de fracionário. Decodificador de vídeo 30 pode localizar um bloco de referência para o primeiro bloco usando o vetor de movimento de precisão de pixel inteiro.

[0116] O decodificador de vídeo 30 pode, por exemplo, determinar a precisão de MVP de fracionário para o primeiro bloco através de construir uma lista de candidato AMVP para o primeiro bloco. A lista de candidato AMVP pode incluir um candidato de vetor de movimento de precisão de fracionário. Decodificador de vídeo 30 pode selecionar o candidato de vetor de movimento de precisão de fracionário como a precisão de MVP de fracionário para o primeiro bloco. O decodificador de vídeo 30 pode adicionar o candidato de vetor de movimento de precisão de fracionário à lista de candidato para um segundo bloco que deve ser predito utilizando a informação do primeiro bloco.

[0117] Em alternativa ou adicionalmente, em um exemplo, informações de precisão de MVD podem ser sinalizadas, e precisão de MV de subpixel pode ser sempre utilizada, em alguns exemplos. A precisão de MVD pode ser sinalizada ao nível LCU, ao nível CU, ou ao nível PU. Em um exemplo, quando uma PU (ou CU) está indicada para ter

precisão de MVD de inteiro, a PU (PU ou todos dentro desta CU) pode ter precisão de MVD de inteiro. Para PUs codificadas por AMVP, MVD da PU podem ter precisão de pixel inteiro, enquanto MV e MV da PU pode ter precisão de subpixel. Assim, adicionar uma precisão de MVD de inteiro a uma precisão de MVP de subpixel resulta em um vetor de movimento de subpixel.

[0118] Por exemplo, decodificador de vídeo 30 pode determinar uma precisão de MVD para um primeiro bloco é a precisão de pixel inteiro. Decodificador de vídeo 30 pode construir uma lista de candidato (por exemplo, e lista de candidato AMVP) de MVPs para o primeiro bloco que inclui pelo menos um candidato de vetor de movimento de precisão de fracionário. Decodificador de vídeo 30 pode escolher entre a lista de candidato o candidato de vetor de movimento de precisão de fracionário e determinar uma precisão de pixel vetor de movimento fracionário com base no candidato de vetor de movimento de precisão de fracionário e a precisão de MVD de pixel inteiro. Decodificador de vídeo 30 pode localizar um bloco de referência para o primeiro bloco usando a vetor de movimento de precisão de pixel fracionário.

[0119] Em um outro exemplo, o flag de precisão de vetor de movimento pode ser parcialmente aplicado a uma LCU ou uma CU. Por exemplo, o flag de precisão de inteiro de CU não é aplicado a suas PUs que são codificadas com modos de codificação predefinidos, como e fusão e pulo, ou com partições predefinidas, como as partições não-2Nx2N, ou com ferramenta de codificação especial, tais como salto de transformada ou nenhum resíduo.

[0120] Por exemplo, um decodificador de vídeo 30 pode determinar para os dados de vídeo uma precisão de vetor

de movimento padrão e, em resposta a uma PU dos dados de vídeo sendo codificados em um modo especial, localizar um bloco de referência para a PU utilizando um vetor de movimento da precisão de vetor de movimento padrão. O modo especial pode ser, por exemplo, um ou mais de um modo de pulo, um modo de mesclagem $2N \times 2N$, um modo de mesclagem, um modo de pulo de transformada, ou um modo de particionamento assimétrica. Em resposta a uma segunda PU dos dados de vídeo sendo codificados utilizando modos diferentes de um modo especial, decodificador de vídeo 30 pode determinar para segunda PU dos dados de vídeo, uma precisão de vetor de movimento sinalizada e localizar um bloco de referência para a segunda PU usando um vetor de movimento de precisão de vetor de movimento sinalizada. O decodificador de vídeo 30 pode determinar para uma CU dos dados de vídeo, um vetor de movimento de precisão sinalizado que é diferente da precisão de vetor de movimento padrão. A CU pode, por exemplo, incluir a PU e/ou a segunda PU. Em um exemplo, precisão de vetor de movimento sinalizada pode ser precisão de pixel inteiro, enquanto a precisão de vetor de movimento padrão é um vetor de movimento de precisão de fracionário. Em outros exemplos, o vetor de movimento de precisão padrão pode ser uma precisão de vetor de movimento de fracionário.

[0121] Em um exemplo, informações de precisão MV / MVD podem ser sinalizadas apenas para a PU ou CU que tem um DMV não zero. Quando informações de precisão de MV / MVD não são sinalizadas, MV de subpixel pode ser utilizado para a PU ou CU. Informações de precisão de MV / MVD podem ser sinalizadas após MVD de uma PU ou CU. Um MVD igual a zero

pode ser utilizado para significar que o componente vertical do MVD e os componentes horizontais do MVD são iguais a 0.

[0122] Por exemplo, para um bloco atual de dados de vídeo, decodificador de vídeo 30 pode receber um valor de MVD e, em resposta ao valor MVD ser igual a zero, determinar um vetor de movimento para o bloco atual tem de precisão de vetor de movimento de subpixel. O valor de MVD sendo igual a zero pode indicar que tanto um componente x do valor MVD e um componente y do valor MVD são iguais a zero. Para um segundo bloco atual de dados de vídeo, decodificador de vídeo 30 pode receber um segundo valor MVD e, em resposta ao segundo valor MVD sendo um valor diferente de zero, receber uma indicação de uma precisão de vetor de movimento para um segundo vetor de movimento para o segundo bloco atual. O decodificador de vídeo 30 pode localizar, na imagem de referência, um bloco de referência para o segundo bloco atual utilizando o segundo vetor de movimento. Para o segundo bloco atual, decodificador de vídeo 30 pode receber a indicação da precisão de vetor de movimento depois de receber o segundo valor MVD.

[0123] Quando informações de precisão de MV / MVD são sinalizadas no nível PU, a informação de precisão MV / MVD pode não ser sinalizada se um ou mais (por exemplo, houver) das seguintes condições: (1) a PU é codificada com modo de mesclagem / pulo, (2) a PU está codificada com o modo AMVP, e MVD em cada direção de predição da PU é igual a zero, ou (3) em alternativa ou adicionalmente, se uma CU poderia conter PU intracodificada e PU intercodificada em conjunto, o que não é permitido no HEVC, e quando a PU é

intracodificado, a sinalização de informações de precisão de MV / MVD em nível PU é ignorada.

[0124] O decodificador de vídeo 30 pode, por exemplo, receber, para um primeiro bloco de dados de vídeo (por exemplo, uma primeira PU), primeiro informação de precisão de vetor de movimento. Em resposta a um segundo bloco de dados de vídeo que satisfaz uma condição, decodificador de vídeo 30 pode determinar segunda informação de vetor de movimento para corresponder a uma precisão padrão. Em um exemplo, a condição pode ser o segundo bloco que está sendo codificado utilizando o modo de mesclagem ou o modo de pulo. Em outro exemplo, a condição pode ser o segundo bloco sendo codificado utilizando o modo AMVP e um MVD para cada direção de predição do segundo bloco sendo igual a zero. A precisão padrão pode, por exemplo, ser uma precisão de fracionário em alguns exemplos ou uma precisão de número inteiro em outros exemplos. As primeira e segunda informações de precisão de vetor de movimento podem, por exemplo, ser uma ou ambos de uma precisão de vetor de movimento ou uma precisão de MVD.

[0125] Quando informações de precisão de MV / MVD são sinalizada no nível CU, a informação de precisão MV / MVD pode não ser sinalizada se uma (e possivelmente uma ou mais) das seguintes condições for verdadeiro para todas as PUs dentro da CU: (1) a PU é intracodificada, (2) a PU é codificada com modo de mesclagem / pulo, ou (3) a PU é codificada com o modo AMVP, e MVD em cada direção de predição da PU é igual a zero. Alternativamente ou adicionalmente, quando a informação de precisão de vetor de movimento não é sinalizada, uma precisão de vetor de movimento padrão, tal

como a precisão de vetor de movimento de inteiro, pode ser utilizado para a PU ou CU.

[0126] O decodificador de vídeo 30 pode, por exemplo, receber, para uma primeira CU de dados de vídeo, primeira informação de precisão de vetor de movimento e, em resposta a uma segunda CU dos dados de vídeo que satisfaz uma condição, determinar segunda informação de vetor de movimento para corresponder a uma precisão padrão. A condição pode, por exemplo, ser que todas as PUs dentro da CU são intracodificadas, todas as PUs dentro da CU são codificadas utilizando o modo de mesclagem ou o modo de pulo, todas as PUs dentro da CU são codificados usando AMVP e um MVD para cada direção de todas as PUs sendo igual a zero. A precisão padrão pode, por exemplo, ser precisão de fracionário ou pode não haver precisão. Por exemplo, se um bloco é intrapredito, em seguida, o bloco não tem vetor de movimento associado e, portanto, não associado à precisão de vetor de movimento. As primeira e segunda informações de precisão de vetor de movimento podem, por exemplo, incluir um ou ambos de precisão de vetor de movimento ou precisão de MVD.

[0127] Quando PU codificada por AMVP atual é sinalizada / derivada como com precisão de vetor de movimento de pixel inteiro, um ou mais (e, em alguns exemplos, todos) os candidatos de MV de blocos vizinhos espaciais, blocos vizinhos temporais, ou ambos, podem ser arredondados para precisão de pixel inteiro antes de cortar no processo de geração de lista AMVP. Quando MV de pixel inteiro é sinalizado / derivado para ser utilizado para uma mesclagem atual, CU / PU codificada por pulo, ou ambos um ou mais (e, em alguns exemplos, todos) os candidatos de MV a partir de

blocos vizinhos temporais espaciais, blocos vizinhos temporais, ou ambos, pode ser arredondado para precisão de pixel inteiro antes de cortar no processo de geração de lista de mesclagem.

[0128] Por exemplo, decodificador de vídeo 30 pode identificar um ou mais candidatos de vetor de movimento para inclusão em uma lista de candidato (por exemplo, uma lista de candidato de mesclagem ou uma lista de candidato AMVP) para um bloco. Os um ou mais candidatos de vetor de movimento podem, por exemplo, incluir, um ou mais candidatos vizinhos espaciais e/ou um ou mais candidatos vizinhos temporais. Os um ou mais candidatos de vetor de movimento podem incluir, pelo menos, um candidato de precisão de vetor de movimento de fracionário. Em resposta a uma precisão de vetor de movimento para o bloco sendo precisão de pixel inteiro, decodificador de vídeo 30 pode arredondar um ou candidatos de vetor de movimento para determinar um ou mais candidatos de vetor de movimento de precisão de inteiro. O decodificador de vídeo 30 pode executar uma operação de remoção de um ou mais candidatos de vetor de movimento de precisão de inteiro.

[0129] Em um exemplo, o flag de precisão de vetor de movimento pode ser usado ou condicionalmente utilizado como contextos de CABAC de outros elementos de sintaxe. Isto é, diferentes modelos de contexto, dependendo do flag de precisão de vetor de movimento, podem ser utilizados para codificar determinado elemento de sintaxe. Em um exemplo, ao codificar um índice de candidato AMVP para um bloco tal como PU, o flag de precisão de vetor de movimento (s) de uma PU ou uma CU associada ou blocos espacialmente vizinhos ou blocos temporalmente vizinhos é (são) usado como o contexto

de codificação de CABAC (s). Em alternativa ou adicionalmente, em alguns exemplos, a probabilidade inicializada de índice de candidato AMVP sendo igual a 0 pode ser definida perto de 1, quando flag de precisão de vetor de movimento indicando a precisão de vetor de movimento de pixel inteiro. Em alternativa ou adicionalmente, em alguns casos, tais como apenas em fatias B, ou apenas quando a fatia está em um determinado nível temporal, ou quando o parâmetro de quantificação é maior do que um limite predefinido, o flag de precisão de vetor de movimento pode ser usado como contextos de CABAC para outros elementos de sintaxe, como AMVP índice de candidato.

[0130] Um ou mais destes exemplos podem ser combinados. Por exemplo, na prática, qualquer combinação de qualquer parte do exemplo pode ser usada como exemplo novo. Além disso, subexemplos dos exemplos acima são discutidos abaixo.

[0131] Alguns exemplos referem-se a derivação de precisão de vetor de movimento do lado do decodificador para o conteúdo da tela. Em um exemplo, pode ser utilizado em forma de L ou correspondência de modelo de outra forma em amostras reconstruídas. A precisão de vetor de movimento pode ser baseada na diferença entre o resultado do modelo correspondente, tal como SAD, uma posição de pixel inteiro e o resultado da correspondência de sua posição de subpixel vizinho. Por exemplo, quando o resultado de correspondência da posição de número inteiro pixel é muito mais baixo, a precisão de pixel inteiro se aplica. Caso contrário, a precisão de subpixel se aplica. Para definir "muito inferior", pode ser utilizado um limite. Na prática, limite

fixo, limite adaptativo, ou ambos, podem ser utilizados. Para um limite adaptativo, o limite adaptativo pode ser sinalizado no fluxo de bits ou obtido com base em outras informações, tais como o tipo de bloco, ou QP, sinalizado no fluxo de bits. Além disso, um limite para um caso "muito superior" pode também ser definido. Por conseguinte, quando o resultado de correspondência de posição de inteiro menos aquela da posição de subpixel vizinho é maior do que o limite de "muito superior", pode ser usada precisão de um quarto de pixel. Quando a diferença de correspondência está entre os limites de "muito inferior" e "muito superior", pode ser usada precisão de meio pixel. Alternativamente ou em adição, outro método de correspondência de modelo, tal como correspondência de modelo bidirecional baseado em espelho, pode ser utilizada em vez do exemplo acima.

[0132] Em um outro exemplo, a informação de vetor de movimento de precisão pode ser derivada no lado do decodificador com base na propriedade de blocos espacialmente ou temporalmente vizinhos, tais como gradiente, exatidão, ou se a transformada é ignorada para os blocos. A informação de limite pode ser sinalizada no fluxo de bits, derivada a partir do fluxo de bits, ou ambos.

[0133] Alguns exemplos referem-se à sinalização de indicador. Para ajustar de forma adaptativa para diferentes conteúdos, uma combinação de diferentes métodos de derivação de precisão de vetor de movimento do lado do decodificador (DMPD) pode ser utilizada. Para indicar qual método ou métodos estão em uso, um indicador pode ser sinalizado no fluxo de bits. Em um exemplo, o indicador pode ser sinalizado ao nível da fatia ou acima para explicitamente dizer ao

decodificador que método ou métodos de DMPD irão ser usados. Em outro exemplo, o uso de alguns métodos DMPD é sinalizado no fluxo de bits, enquanto o uso de outros métodos de DMPD é derivado com base em outras informações, tais como o tipo de fatia e nível temporal da fatia, no fluxo de bits.

[0134] Alguns exemplos referem-se à precisão de vetor de movimento adaptativa sinalizada. Em um exemplo, a precisão de vetor de movimento pode ser sinalizada no fluxo de bits, tal como em nível de PU LCU, ou CU. Um flag / valor pode ser usado para indicar a precisão de vetor de movimento, tais como a precisão de inteiro, precisão de meio pixel, precisão de um quarto de pixel, ou outras precisões. Quando a precisão de vetor de movimento é sinalizada por um bloco ou uma região / fatia, todos os blocos menores dentro deste bloco / região / fatia podem compartilhar a mesma precisão de vetor de movimento. Além disso, a informação de MVD também pode ser sinalizada na precisão sinalizada. Antes de compensação de movimento, MV (MV preditor + MVD) pode ser arredondada para a precisão sinalizada. O arredondamento pode ser em direção ao infinito positivo, infinito negativo, zero, ou infinito (valor negativo é arredondado para infinito negativo enquanto valor positivo é arredondado para infinito positivo). Alternativamente ou em adição, MV preditor pode ser primeiro arredondado como mencionado acima e, em seguida, formar o MV para um bloco. Depois de compensação de movimento, o MV do bloco é guardado para a predição de MV de blocos posteriores. Ao salvar o MV , o MV arredondado pode ser salvo, por exemplo, para ser usado mais tarde como um candidato de mesclagem ou AMVP candidato para um bloco posteriormente decodificado. Alternativamente ou em adição,

o MV não arredondado pode ser guardado em vez do vetor de movimento arredondado, que pode potencialmente manter o campo de movimento mais preciso.

[0135] Em outro exemplo, a informação de precisão de vetor de movimento não é sinalizado para o modo de pulo, modo de mesclagem $2N \times 2N$, ou ambos. Em tal um exemplo, informações de precisão de vetor de movimento também podem não ser sinalizadas para PUs unidas. Alternativamente ou adicionalmente, PUs, que são codificadas em modos de codificação especiais, tais como o modo de pulo e modo de mesclagem, ou com partições especiais, tais como partições assimétricas, ou com profundidade de transformada especial ou com pulo de transformada, pode manter a precisão de vetor de movimento padrão, tal como um quarto de pel, mesmo quando precisão de MV inteiro é sinalizada no seu nível de CU. Alternativamente ou adicionalmente, outras informações codificadas, tais como o nível temporal, QP, profundidade de CU, podem também ser consideradas como um modo de codificação ou especial uma ferramenta de codificação especial.

[0136] Ao codificar por entropia a informação de precisão de vetor de movimento com CABAC, contextos diferentes da informação de precisão de vetor de movimento em blocos / UCs espacialmente vizinhos podem ser usados para salvar armazenador de linha, tais como profundidade de CU, particionamento de PU, tamanho do bloco, nível temporal e assim por diante.

[0137] A figura 8 é um fluxograma que ilustra um método exemplar para a codificação de dados de vídeo, de acordo com as técnicas descritas nesta divulgação. As técnicas da figura 8 irão ser descritas com referência a um

codificador de vídeo, tais como codificador de vídeo 20. O codificador de vídeo 20 determina que uma precisão de vetor de movimento para um primeiro bloco é precisão de pixel inteiro (202). O codificador de vídeo 20 constrói uma lista de candidato de mesclagem para o primeiro bloco (204). Como parte de construir a lista de candidato de mesclagem, codificador de vídeo 20 pode adicionar um candidato de vetor de movimento de precisão de fracionário à lista de candidato de mesclagem. Assim, a lista de candidato de mesclagem pode incluir um candidato de vetor de movimento de precisão de fracionário. O codificador de vídeo 20 seleciona o candidato de vetor de precisão de movimento de fracionário para codificar o primeiro bloco (206). Para codificar o primeiro bloco usando um modo de mesclagem, codificador de vídeo 20 arredonda o candidato de vetor de movimento de precisão de fracionário para determinar um vetor de movimento de precisão de pixel inteiro para o primeiro bloco (208). O codificador de vídeo 20 localiza um bloco de referência para o primeiro bloco utilizando o vetor de movimento de precisão de pixel inteiro (210). O codificador de vídeo 20 codifica o primeiro bloco com base no bloco de referência (212).

[0138] A figura 9 é um fluxograma que ilustra um método exemplar para a decodificação de dados de vídeo, de acordo com as técnicas descritas nesta divulgação. As técnicas da figura 9 irão ser descritas com referência a um decodificador de vídeo, tal como decodificador de vídeo 30. O decodificador de vídeo 30 determina que um modo de codificação para um primeiro bloco é o modo de mesclagem (220). O decodificador de vídeo 30 determina que uma precisão de vetor de movimento para o primeiro bloco é precisão de

pixel inteiro (222). O decodificador de vídeo 30 constrói uma lista de candidato de mesclagem para o primeiro bloco (224). O decodificador de vídeo 30 constrói a lista de candidato de mesclagem, adicionando um candidato de vetor de movimento de precisão de fracionário à lista de candidato de mesclagem. O decodificador de vídeo 30 seleciona o candidato de vetor de movimento de precisão de fracionário para decodificar o primeiro bloco (226). Em resposta à seleção do vetor de movimento de precisão de fracionário, o decodificador de vídeo 30 arredonda o candidato de precisão de vetor de movimento de fracionário para determinar um vetor de movimento de precisão de pixel inteiro para o primeiro bloco (228). O decodificador de vídeo 30 localiza um bloco de referência para o primeiro bloco utilizando o vetor de movimento de precisão de pixel inteiro (230). O decodificador de vídeo 30 decodifica o primeiro bloco com base no bloco de referência (232).

[0139] A figura 10 é um fluxograma que ilustra um método exemplar para a codificação de dados de vídeo, de acordo com as técnicas descritas nesta divulgação. As técnicas da figura 10 irão ser descritas com referência ao decodificador de vídeo 30; No entanto, muitas das técnicas podem também ser realizadas por um codificador de vídeo, tal como um codificador de vídeo 20. Para um bloco atual sendo codificado em um quadro atual, o decodificador de vídeo 30 identifica uma posição de pixel inteiro de um bloco vizinho já codificado (234). Com base em uma localização da posição de pixel inteiro, o decodificador de vídeo 30 aplica um modelo para determinar uma pluralidade de posições de pixel inteiro na imagem (236). O decodificador de vídeo 30 aplica

o modelo a uma pluralidade de posições de subpixel para determinar uma pluralidade de posições de subpixels na imagem (238). O decodificador de vídeo 30 compara os um ou mais valores de pixel para a pluralidade de posições de pixel inteiro com os um ou mais valores de pixel para a pluralidade de posições de subpixel (240). Com base na comparação, o decodificador de vídeo 30 determina uma precisão de vetor de movimento para um vetor de movimento (242). O decodificador de vídeo 30 codifica o bloco atual utilizando o vetor de movimento com a precisão de vetor de movimento determinada (242).

[0140] A figura 11 é um fluxograma que ilustra um método exemplar para a codificação de dados de vídeo, de acordo com as técnicas descritas nesta divulgação. As técnicas da figura 11 irão ser descritas com referência ao decodificador de vídeo 30; No entanto, muitas das técnicas também podem ser realizadas por um codificador de vídeo, tal como um codificador de vídeo 20. Para um bloco atual a ser codificado, o decodificador de vídeo 30 localiza um ou mais blocos vizinhos (246). O decodificador de vídeo 30 determina uma propriedade de um ou mais blocos vizinhos (248). Com base na propriedade de um ou mais blocos vizinhos, o decodificador de vídeo 30 determina uma precisão de vetor de movimento para um vetor de movimento (250). O decodificador de vídeo 30 codifica o bloco atual utilizando o vetor de movimento com a precisão de vetor de movimento determinada.

[0141] A figura 12 é um fluxograma que ilustra um método exemplar para a codificação de dados de vídeo, de acordo com as técnicas descritas nesta divulgação. As técnicas da figura 12 irão ser descritas com referência ao

decodificador de vídeo 30; No entanto, muitas das técnicas pode também ser realizadas por um codificador de vídeo, tal como um codificador de vídeo 20. O decodificador de vídeo 30 determina que um modo de codificação para um primeiro bloco é o modo de mesclagem (254). O decodificador de vídeo 30 determina que uma precisão de vetor de movimento para o primeiro bloco é precisão de pixel inteiro (256). O decodificador de vídeo 30 constrói uma lista de candidato de mesclagem para o primeiro bloco, em que a lista de candidato de mesclagem compreende um candidato de vetor de movimento de precisão de fracionário (258). O decodificador de vídeo 30 seleciona o candidato de vetor de movimento precisão de fracionário para decodificar o primeiro bloco (260). O decodificador de vídeo 30 arredonda o candidato de vetor de movimento de precisão de fracionário para determinar um vetor de movimento de precisão de pixel inteiro (262). O decodificador de vídeo 30 localiza um bloco de referência para o primeiro bloco usando o vetor de movimento de precisão de pixel inteiro.

[0142] A figura 13 é um fluxograma que ilustra um método exemplar para a codificação de dados de vídeo, de acordo com as técnicas descritas nesta divulgação. As técnicas da figura 13 irão ser descritas com referência ao decodificador de vídeo 30; No entanto, muitas das técnicas também podem ser realizadas por um codificador de vídeo, tal como um codificador de vídeo 20. Para um bloco atual de dados de vídeo, decodificador de vídeo 30 determina um valor de diferença de vetor de movimento (266). Em resposta ao valor da diferença de vetor de movimento igual a zero, o decodificador de vídeo 30 determina que um vetor de movimento

para o bloco atual tem precisão de vetor de movimento de subpixel (268).

[0143] A figura 14 é um fluxograma que ilustra um método exemplar para a codificação de dados de vídeo, de acordo com as técnicas descritas nesta divulgação. As técnicas da figura 14 irão ser descritas com referência ao decodificador de vídeo 30; No entanto, muitas das técnicas também podem ser realizadas por um codificador de vídeo, tal como um codificador de vídeo 20. O decodificador de vídeo 30 determina os dados de vídeo para uma precisão de vetor de movimento padrão (270). Em resposta a uma PU dos dados de vídeo sendo codificados em um modo especial, o decodificador de vídeo 30 localiza um bloco de referência para a PU utilizando um vetor de movimento da precisão de vetor de movimento padrão (272).

[0144] A figura 15 é um fluxograma que ilustra um método exemplar para a codificação de dados de vídeo, de acordo com as técnicas descritas nesta divulgação. As técnicas da figura 15 irão ser descritas com referência ao decodificador de vídeo 30; No entanto, muitas das técnicas pode também ser realizadas por um codificador de vídeo, tal como um codificador de vídeo 20. O decodificador de vídeo 30 identifica um ou mais candidatos de vetor de movimento para inclusão em uma lista de candidato para um bloco, em que os um ou mais candidatos de vetor de movimento compreendem pelo menos um candidato de vetor de movimento de precisão de fracionário (274). Em resposta a uma precisão de vetor de movimento para o bloco sendo precisão de pixel inteiro, o decodificador de vídeo 30 arredonda um ou mais candidatos de vetor de movimento para determinar um ou mais candidatos de

vetor de movimento de precisão de inteiro (276). Depois de arredondar os um ou mais candidatos de vetor de movimento, o decodificador de vídeo 30 executa uma operação de remoção de um ou mais candidatos de vetor de movimento de precisão de inteiro (278).

[0145] A figura 16 é um fluxograma que ilustra um método exemplar para a codificação de dados de vídeo, de acordo com as técnicas descritas nesta divulgação. As técnicas da figura 16 irão ser descritas com referência ao decodificador de vídeo 30; No entanto, muitas das técnicas pode também ser realizadas por um codificador de vídeo, tal como um codificador de vídeo 20. O decodificador de vídeo 30 determina, para um primeiro bloco de dados de vídeo e um segundo bloco de dados de vídeo, uma precisão padrão (280). O decodificador de vídeo 30 determina, para o primeiro bloco de dados de vídeo, a primeira informação de precisão de vetor de movimento (282). Em resposta ao segundo bloco de dados de vídeo atendendo a uma condição, determinar segunda informação de vetor de movimento para corresponder à precisão padrão (284). Os primeiro e segundo blocos podem, por exemplo, ser as primeira e segunda PUs ou as primeira e segunda CUs.

[0146] A figura 17 é um fluxograma que ilustra um método exemplar para a codificação de dados de vídeo, de acordo com as técnicas descritas nesta divulgação. As técnicas da figura 17 irão ser descritas com referência ao decodificador de vídeo 30; No entanto, muitas das técnicas pode também ser realizadas por um codificador de vídeo, tal como um codificador de vídeo 20. O decodificador de vídeo 30 determina uma diferença de precisão de vetor de movimento

para um primeiro bloco é precisão de pixel inteiro (286). O decodificador de vídeo 30 constrói uma lista de candidato de preditores de vetor de movimento para o primeiro bloco, em que a lista de candidato compreende um candidato de vetor de movimento de precisão de fracionário (288). O decodificador de vídeo 30 seleciona a partir da lista de candidato o candidato de vetor de movimento de precisão de fracionário (290). O decodificador de vídeo 30 determina um vetor de movimento de precisão de pixel fracionário com base no candidato de vetor de movimento de precisão de fracionário e na diferença de vetor de movimento de precisão de pixel inteiro (292). O decodificador de vídeo 30 localiza um bloco de referência para o primeiro bloco utilizando o vetor de movimento de precisão de pixel fracionário (294).

[0147] Para facilitar a explicação, as técnicas das figuras 8 a 17 foram apresentadas separadamente, mas é contemplado que as técnicas descritas podem ser utilizadas em conjunto umas com as outras. Além disso, é contemplado que as porções de algumas técnicas podem ser utilizadas em combinação com porções de outras técnicas.

[0148] Em um ou mais exemplos, as funções descritas podem ser implementadas em hardware, software, firmware, ou qualquer combinação dos mesmos. Se implementadas em software, as funções podem ser armazenadas ou transmitidas como uma ou mais instruções ou código em um meio legível por computador e executadas por uma unidade de processamento com base em hardware. Meios legíveis por computador podem incluir meios de armazenamento legíveis por computador, que correspondem a um meio tangível, tais como mídia de armazenamento de dados ou meios de comunicação, incluindo

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

[0149] A título de exemplo, e não como limitação, tais meios de armazenamento legíveis por computador podem compreender RAM, ROM, EEPROM, CD-ROM ou outro armazenamento em disco óptico, armazenamento em disco magnético, ou outros dispositivos de armazenamento magnéticos, memória flash, ou qualquer outro meio que possa ser usado para armazenar o código de programa desejado sob a forma de instruções ou estruturas de dados, e que possa ser acessado por um computador. Além disso, qualquer conexão é adequadamente denominada um meio legível por computador. Por exemplo, se as instruções são transmitidas de um site, servidor ou outra fonte remota utilizando um cabo coaxial, cabo de fibra óptica, par trançado, linha de assinante digital (DSL) ou tecnologias sem fio, tais como infravermelhos, rádio e micro-ondas, então o cabo coaxial, cabo de fibra óptica, par trançado, DSL, ou tecnologias sem fio, tais como infravermelho, rádio e micro-ondas estão incluídos na

definição de meio. Deve ser entendido, no entanto, que mídia de armazenamento legível por computador e mídia de armazenamento de dados não incluem conexões, ondas de portadoras, sinais ou outros mídia de comunicação transitória, mas são direcionadas para mídia de armazenamento tangível não transitória. Disco e disquete, como aqui utilizado, incluem disco compacto (CD), disco laser, disco óptico, disco versátil digital (DVD), disquete e disco blu-ray onde disquetes geralmente reproduzem dados magneticamente, enquanto que discos reproduzem dados opticamente com lasers. Combinações dos anteriores também devem ser incluídas dentro do âmbito dos mídia legível por computador.

[0150] As instruções podem ser executadas por um ou mais processadores, tais como os um ou mais processadores de sinal digital (DSPs), microprocessadores de uso geral, circuitos integrados de aplicação específica (ASIC), arranjos de lógica programáveis em campo (FPGA), ou outro equivalente integrado ou conjunto de circuitos lógicos discretos. Por conseguinte, o termo "processador" tal como aqui utilizado pode referir-se a qualquer estrutura precedente ou qualquer outra estrutura adequada para aplicação das técnicas aqui descritas. Além disso, em alguns aspectos, a funcionalidade aqui descrita pode ser provida dentro de módulos de hardware e/ou software dedicado configurados para a codificação e decodificação, ou incorporados em um codec combinado. Além disso, as técnicas podem ser totalmente implementadas em um ou mais circuitos ou elementos lógicos.

[0151] As técnicas da presente divulgação podem ser implementadas em uma vasta variedade de dispositivos ou aparelhos, incluindo um aparelho telefônico sem fio, um circuito integrado (IC) ou um conjunto de ICs (por exemplo, um conjunto de chips). Vários componentes, módulos ou unidades são descritos nesta divulgação para enfatizar os aspectos funcionais de dispositivos configurados para executar as técnicas divulgadas, mas não precisam necessariamente de realização por diferentes unidades de hardware. Em vez disso, tal como descrito acima, várias unidades podem ser combinadas em uma unidade de hardware codec ou providas por um conjunto de unidades de hardware interoperativo, incluindo um ou mais processadores, conforme descrito acima, em conjunto com o software e/ou firmware adequado.

[0152] Vários exemplos foram descritos. Estes e outros exemplos estão dentro do âmbito das reivindicações seguintes.

REIVINDICAÇÕES

1. Método para decodificar dados de vídeo codificados, o método caracterizado pelo fato de que compreende:

determinar (254) que um modo de codificação para um primeiro bloco é o modo de mesclagem, em que, no modo de mesclagem, uma lista de candidato de mesclagem que compreende candidatos de vetor de movimento é construída com base em blocos já codificados, e o primeiro bloco é decodificado com o uso do vetor de movimento de candidato identificado por um índice sinalizado que identifica um dos candidatos na lista de candidato de mesclagem;

determinar (256) que uma precisão de vetor de movimento para o primeiro bloco é a precisão de pixel de número inteiro;

construir (268) uma lista de candidato de mesclagem para o primeiro bloco, em que construir a lista de candidato de mesclagem compreende adicionar um candidato de vetor de movimento de precisão fracional à lista de candidato de mesclagem;

selecionar (260) o candidato de vetor de movimento de precisão fracional para decodificar o primeiro bloco, com base em um índice sinalizado que identifica o candidato de vetor de movimento de precisão fracional na lista de candidato de mesclagem;

em resposta à seleção do vetor de movimento de precisão fracional, arredondar (262) o candidato de vetor de movimento de precisão fracional para determinar um vetor de movimento de precisão de pixel de número inteiro para o primeiro bloco;

localizar (264) um bloco de referência para o primeiro bloco com o uso do vetor de movimento de precisão de pixel de número inteiro obtido arredondando-se o candidato de vetor de movimento de precisão fracional selecionado da lista de candidato de mesclagem; e

decodificar o primeiro bloco com base no bloco de referência.

2. Método, de acordo com a reivindicação 1, caracterizado pelo fato de que compreende adicionalmente:

determinar que um modo de codificação para um segundo bloco é diferente de um modo de mesclagem;

determinar que uma precisão de vetor de movimento para o segundo bloco é a precisão de pixel de número inteiro;

determinar um preditor de vetor de movimento de precisão fracional para o segundo bloco;

arredondar o preditor de vetor de movimento de precisão fracional para determinar um preditor de vetor de movimento de precisão de pixel de número inteiro para o segundo bloco;

determinar uma diferença de vetor de movimento para o segundo bloco, em que a diferença de vetor de movimento compreende uma diferença de precisão de pixel de número inteiro;

determinar um vetor de movimento de precisão de pixel de número inteiro para o segundo bloco com base no preditor de vetor de movimento de precisão de pixel de número inteiro e na diferença de vetor de movimento; e

localizar um bloco de referência para o segundo bloco com o uso do vetor de movimento de precisão de pixel de número inteiro.

3. Método para codificar dados de vídeo, o método caracterizado pelo fato de que compreende:

determinar a codificação de um primeiro bloco no modo de mesclagem, em que, no modo de mesclagem, uma lista de candidato de mesclagem que compreende candidatos de vetor de movimento é construída com base em blocos já codificados, e um índice que identifica o candidato dentre os candidatos na lista de candidato de mesclagem é sinalizado, para codificar o primeiro bloco;

determinar que uma precisão de vetor de movimento para um primeiro bloco é a precisão de pixel de número inteiro;

construir uma lista de candidato de mesclagem para o primeiro bloco, em que a lista de candidato de mesclagem compreende um candidato de vetor de movimento de precisão fracional;

selecionar o candidato de vetor de movimento de precisão fracional para codificar o primeiro bloco;

sinalizar um índice que identifica o vetor de movimento de precisão fracional na lista de candidato de mesclagem;

para codificar o primeiro bloco com o uso de um modo de mesclagem, arredondar o candidato de vetor de movimento de precisão fracional para determinar um vetor de movimento de precisão de pixel de número inteiro para o primeiro bloco; e

localizar um bloco de referência para o primeiro bloco com o uso do vetor de movimento de precisão de pixel de número inteiro obtido arredondando-se o candidato de vetor de movimento de precisão fracional selecionado da lista de

candidato de mesclagem.

4. Método, de acordo com a reivindicação 1, caracterizado pelo fato de que determinar que a precisão de vetor de movimento para o primeiro bloco é a precisão de pixel de número inteiro compreende receber nos dados de vídeo codificados uma indicação de que a precisão de vetor de movimento para o primeiro bloco é a precisão de pixel de número inteiro, ou método, de acordo com a reivindicação 3, que compreende adicionalmente:

gerar para inclusão nos dados de vídeo codificados, uma indicação de que uma precisão de vetor de movimento para o primeiro bloco é a precisão de pixel de número inteiro.

5. Método, de acordo com a reivindicação 1 ou 3, caracterizado pelo fato de que o modo de mesclagem compreende um modo de pulo.

6. Método, de acordo com a reivindicação 3, caracterizado pelo fato de que compreende adicionalmente:

determinar que uma precisão de vetor de movimento para um segundo bloco é a precisão de pixel de número inteiro;

para codificar o segundo bloco em um modo diferente do modo de mesclagem, determinar um preditor de vetor de movimento de precisão fracional para o segundo bloco;

arredondar o preditor de vetor de movimento de precisão fracional para determinar um preditor de vetor de movimento de precisão de pixel de número inteiro para o segundo bloco;

determinar uma diferença de vetor de movimento para o segundo bloco, em que a diferença de vetor de

movimento compreende uma diferença de precisão de pixel de número inteiro;

determinar um vetor de movimento de precisão de pixel de número inteiro para o segundo bloco com base no preditor de vetor de movimento de precisão de pixel de número inteiro e na diferença de vetor de movimento; e

localizar um bloco de referência para o segundo bloco com o uso do vetor de movimento de precisão de pixel de número inteiro;

codificar o primeiro bloco com base no bloco de referência.

7. Método, de acordo com a reivindicação 2 ou 6, caracterizado pelo fato de que determinar o preditor de vetor de movimento de precisão fracional para o segundo bloco compreende:

construir uma lista de candidato de AMVP para o segundo bloco, em que a lista de candidato de AMVP compreende um segundo candidato de vetor de movimento de precisão fracional;

selecionar o segundo candidato de vetor de movimento de precisão fracional como o preditor de vetor de movimento de precisão fracional para o segundo bloco,

sendo que o método, preferencialmente, compreende adicionalmente:

adicionar o vetor de movimento de precisão de número inteiro a uma lista de candidato para um segundo bloco.

8. Método, de acordo com a reivindicação 1 ou 3, caracterizado pelo fato de que compreende adicionalmente:

adicionar o candidato de vetor de movimento de

precisão fracional a uma lista de candidato para um segundo bloco, em que a lista de candidato para o segundo bloco compreende preferencialmente uma lista de candidato de AMVP.

9. Dispositivo para decodificar vídeo, o dispositivo caracterizado pelo fato de que compreende:

uma memória configurada para armazenar dados de vídeo;

um decodificador de vídeo que compreende um ou mais processadores configurados para:

determinar que um modo de codificação para um primeiro bloco é o modo de mesclagem, em que, no modo de mesclagem, uma lista de candidato demesclagem que compreende candidatos de vetor de movimento é construída com base em blocos já codificados, e o primeiro bloco é decodificado com o uso do vetor de movimento de candidato identificado por um índice sinalizado que identifica um dos candidatos na lista de candidato de mesclagem;

determinar que uma precisão de vetor de movimento para o primeiro bloco é a precisão de pixel de número inteiro;

construir uma lista de candidato de mesclagem para o primeiro bloco com base em dados de vídeo armazenada na memória, em que a lista de candidato de mesclagem compreende um candidato de vetor de movimento de precisão fracional;

selecionar o candidato de vetor de movimento de precisão fracional para decodificar o primeiro bloco, com base em um índice sinalizado que identifica o candidato de vetor de movimento de precisão fracional na lista de candidato de mesclagem;

arredondar o candidato de vetor de movimento de

precisão fracional para determinar um vetor de movimento de precisão de pixel de número inteiro para o primeiro bloco;

localizar um bloco de referência para o primeiro bloco com o uso do vetor de movimento de precisão de pixel de número inteiro obtido arredondando-se o candidato de vetor de movimento de precisão fracional selecionado da lista de candidato de mesclagem.

10. Dispositivo, de acordo com a reivindicação 9, caracterizado pelo fato de que os um ou mais processadores são adicionalmente configurados para determinar que a precisão de vetor de movimento para o primeiro bloco é a precisão de pixel de número inteiro, os um ou mais processadores são adicionalmente configurados para receber nos dados de vídeo codificados uma indicação de que a precisão de vetor de movimento para o primeiro bloco é a precisão de pixel de número inteiro.

11. Dispositivo, de acordo com a reivindicação 9, caracterizado pelo fato de que os um ou mais processadores são adicionalmente configurados para:

determinar que um modo de codificação para um segundo bloco é diferente de um modo de mesclagem;

determinar que uma precisão de vetor de movimento para o segundo bloco é a precisão de pixel de número inteiro;

determinar um preditor de vetor de movimento de precisão fracional para o segundo bloco;

arredondar o preditor de vetor de movimento de precisão fracional para determinar um preditor de vetor de movimento de precisão de pixel de número inteiro para o segundo bloco;

determinar uma diferença de vetor de movimento

para o segundo bloco, em que a diferença de vetor de movimento compreende uma diferença de precisão de pixel de número inteiro;

determinar um vetor de movimento de precisão de pixel de número inteiro para o segundo bloco com base no preditor de vetor de movimento de precisão de pixel de número inteiro e na diferença de vetor de movimento;

localizar um bloco de referência para o segundo bloco com o uso do vetor de movimento de precisão de pixel de número inteiro.

12. Dispositivo, de acordo com a reivindicação 11, caracterizado pelo fato de que, para determinar o preditor de vetor de movimento de precisão fracional para o segundo bloco, os um ou mais processadores são adicionalmente configurados para:

construir uma lista de candidato de AMVP para o segundo bloco, em que a lista de candidato de AMVP compreende um segundo candidato de vetor de movimento de precisão fracional;

selecionar o segundo candidato de vetor de movimento de precisão fracional como o preditor de vetor de movimento de precisão fracional para o segundo bloco, em que os um ou mais processadores são preferencialmente configurados adicionalmente para:

adicionar o vetor de movimento de precisão de número inteiro a uma lista de candidato para um segundo bloco.

13. Dispositivo, de acordo com a reivindicação 9, caracterizado pelo fato de que os um ou mais processadores são configurados adicionalmente para:

adicionar o candidato de vetor de movimento de precisão fracional a uma lista de candidato para um segundo bloco, em que a lista de candidato para o segundo bloco compreende preferencialmente uma lista de candidato de AMVP.

14. Memória legível por computador caracterizada pelo fato de que compreende instruções armazenadas na mesma que, quando executadas, fazem com que um computador realize o método conforme definido em qualquer uma das reivindicações 1 a 8.

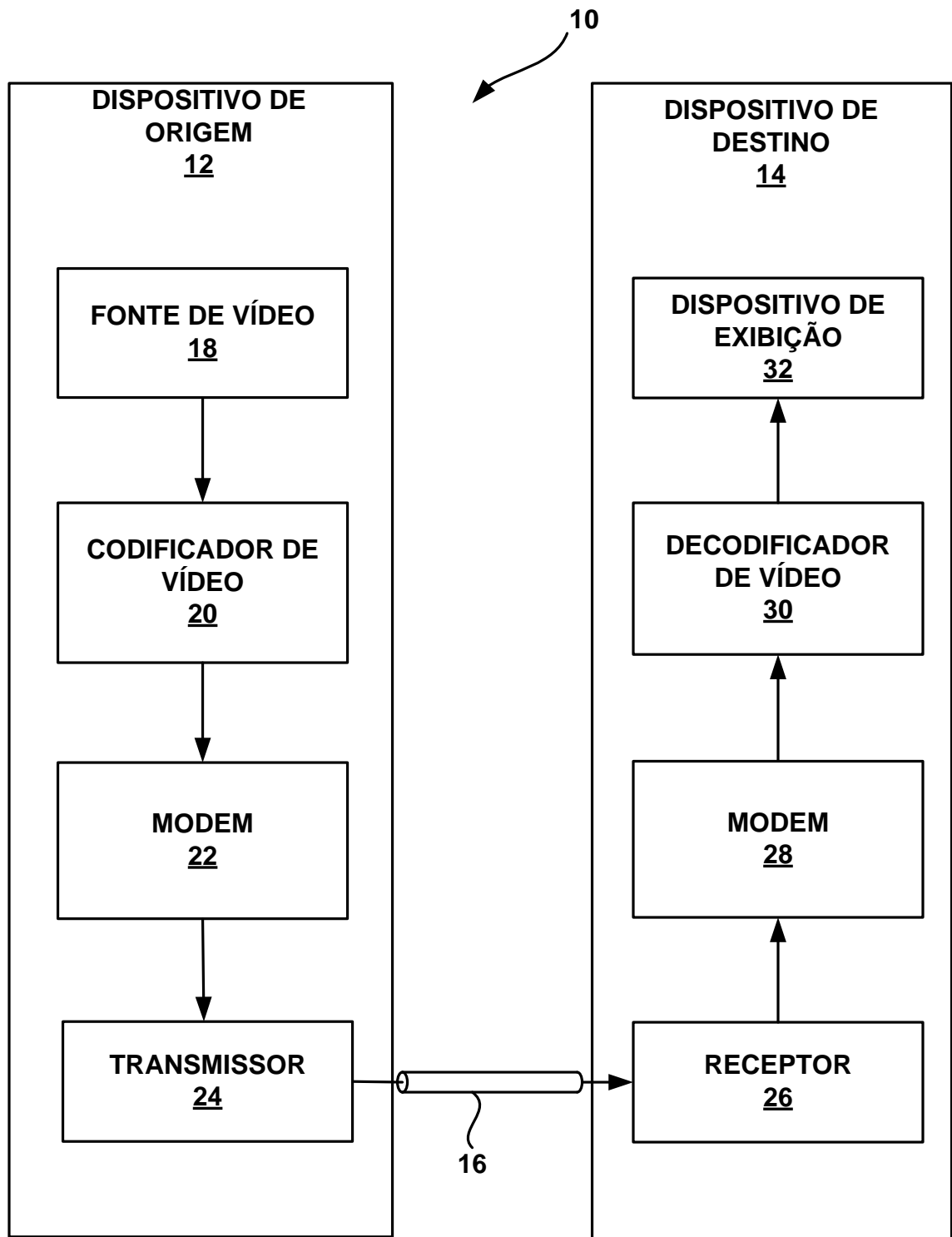


FIG. 1

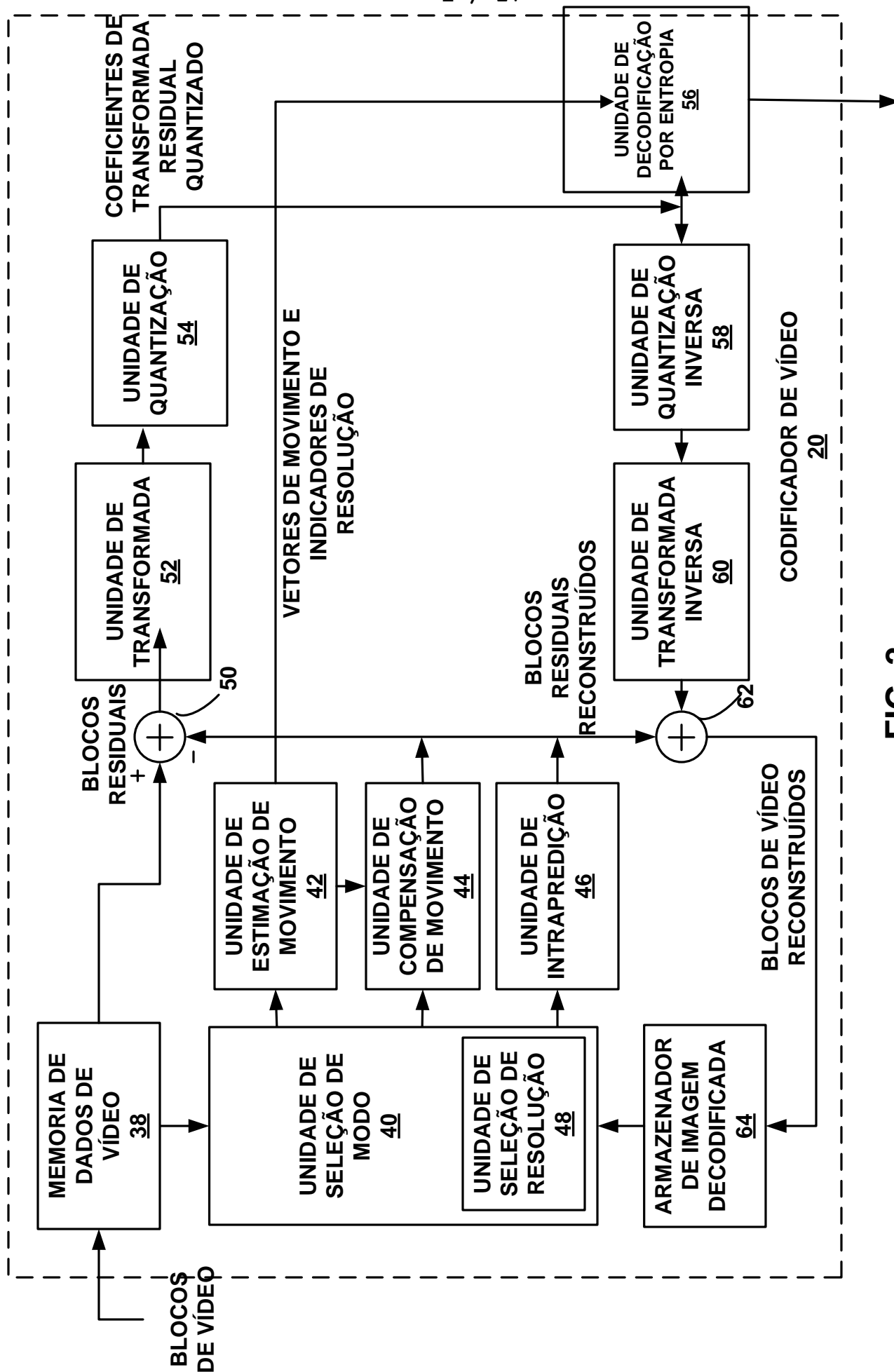


FIG. 2

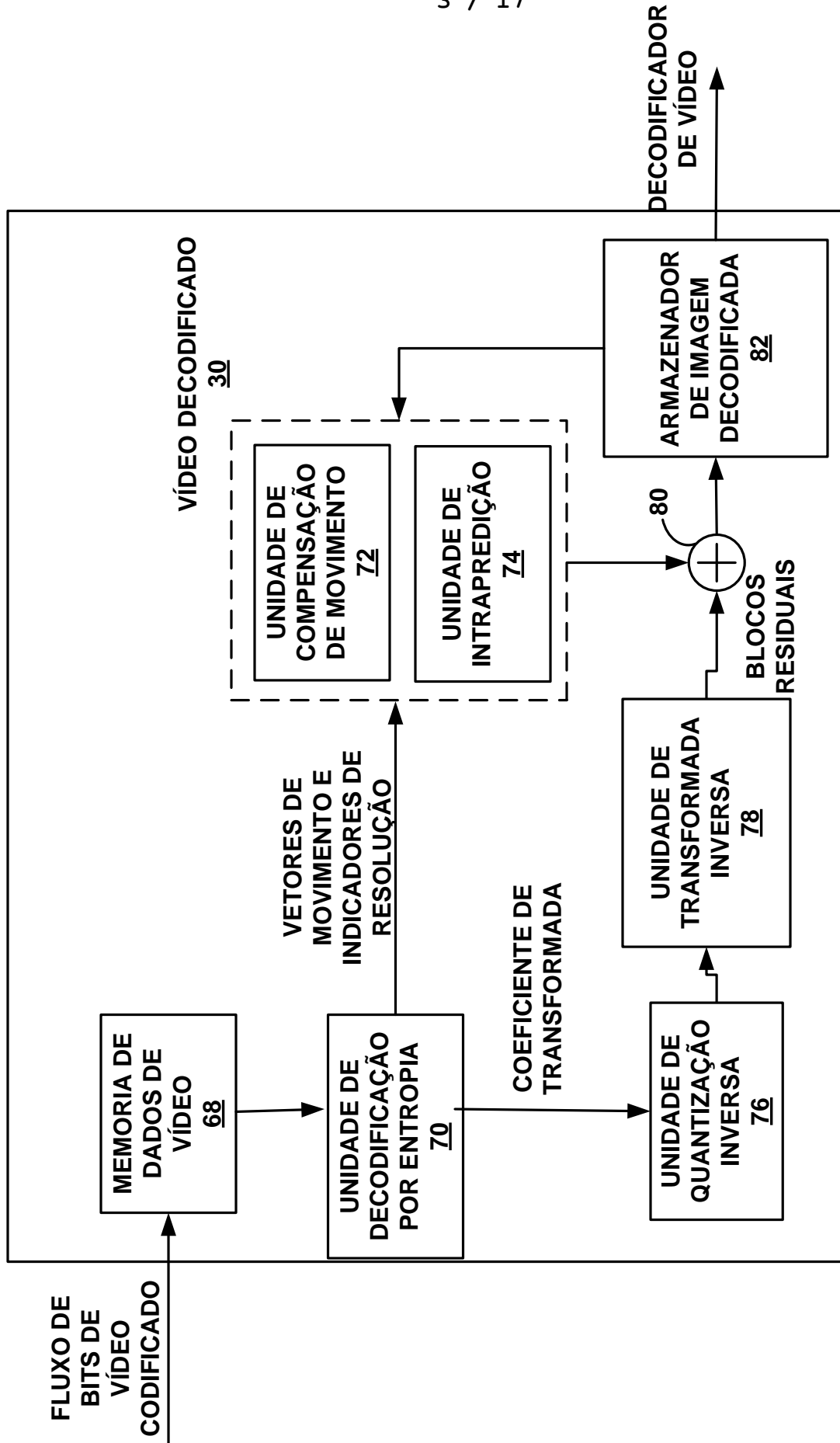


FIG. 3

PEL COMPLETO 100	UM OITAVO DE PEL 106A	UM QUARTO DE PEL 104A	UM OITAVO DE PEL 106B	MEIO PEL 102A	UM OITAVO DE PEL 106C	UM QUARTO DE PEL 104B	UM OITAVO DE PEL 106D
UM OITAVO DE PEL 106E	UM OITAVO DE PEL 106F	UM OITAVO DE PEL 106G	UM OITAVO DE PEL 106H	UM OITAVO DE PEL 106I	UM OITAVO DE PEL 106J	UM OITAVO DE PEL 106K	UM OITAVO DE PEL 106L
UM QUARTO DE PEL 104C	UM OITAVO DE PEL 106M	UM QUARTO DE PEL 104D	UM OITAVO DE PEL 106N	UM QUARTO DE PEL 104E	UM OITAVO DE PEL 106O	UM QUARTO DE PEL 104F	UM OITAVO DE PEL 106P
UM OITAVO DE PEL 106Q	UM OITAVO DE PEL 106R	UM OITAVO DE PEL 106S	UM OITAVO DE PEL 106T	UM OITAVO DE PEL 106U	UM OITAVO DE PEL 106V	UM OITAVO DE PEL 106W	UM OITAVO DE PEL 106X
MEIO PEL 102B	UM OITAVO DE PEL 106Y	UM QUARTO DE PEL 104G	UM OITAVO DE PEL 106Z	MEIO PEL 102C	UM OITAVO DE PEL 106AA	UM QUARTO DE PEL 104H	UM OITAVO DE PEL 106AB
UM OITAVO DE PEL 106AC	UM OITAVO DE PEL 106AD	UM OITAVO DE PEL 106AE	UM OITAVO DE PEL 106AF	UM OITAVO DE PEL 106AG	UM OITAVO DE PEL 106AH	UM OITAVO DE PEL 106AI	UM OITAVO DE PEL 106AJ
UM QUARTO DE PEL 104I	UM OITAVO DE PEL 106AK	UM QUARTO DE PEL 104J	UM OITAVO DE PEL 106AL	UM QUARTO DE PEL 104K	UM OITAVO DE PEL 106AM	UM QUARTO DE PEL 104L	UM OITAVO DE PEL 106AN
UM OITAVO DE PEL 106AO	UM OITAVO DE PEL 106AP	UM OITAVO DE PEL 106AQ	UM OITAVO DE PEL 106AR	UM OITAVO DE PEL 106AS	UM OITAVO DE PEL 106AT	UM OITAVO DE PEL 106AU	UM OITAVO DE PEL 106AV

FIG. 4

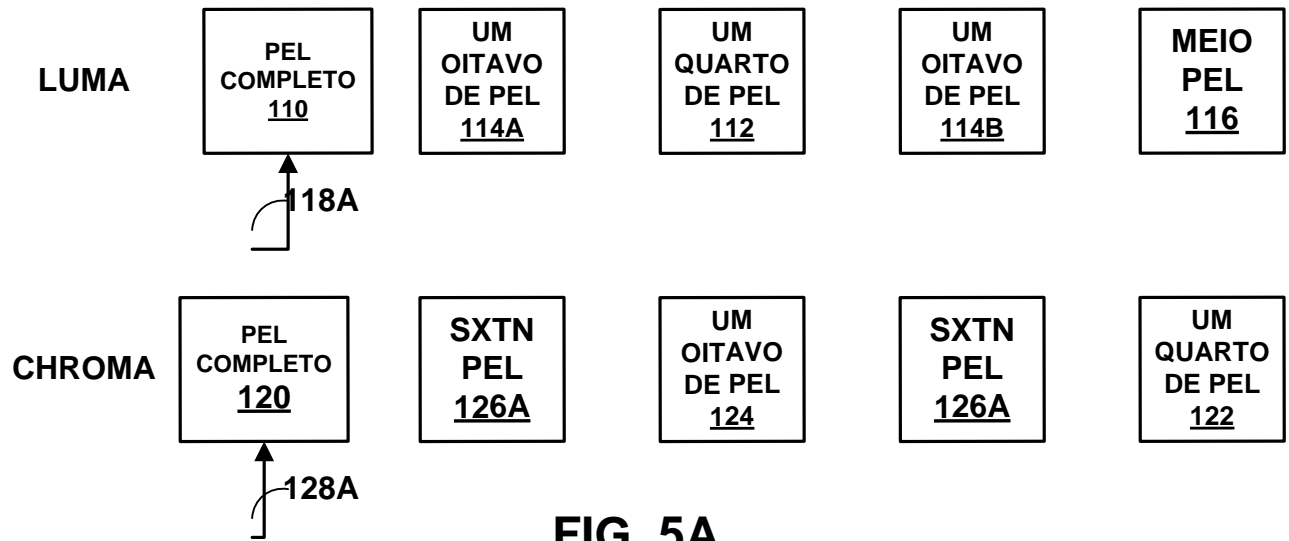


FIG. 5A

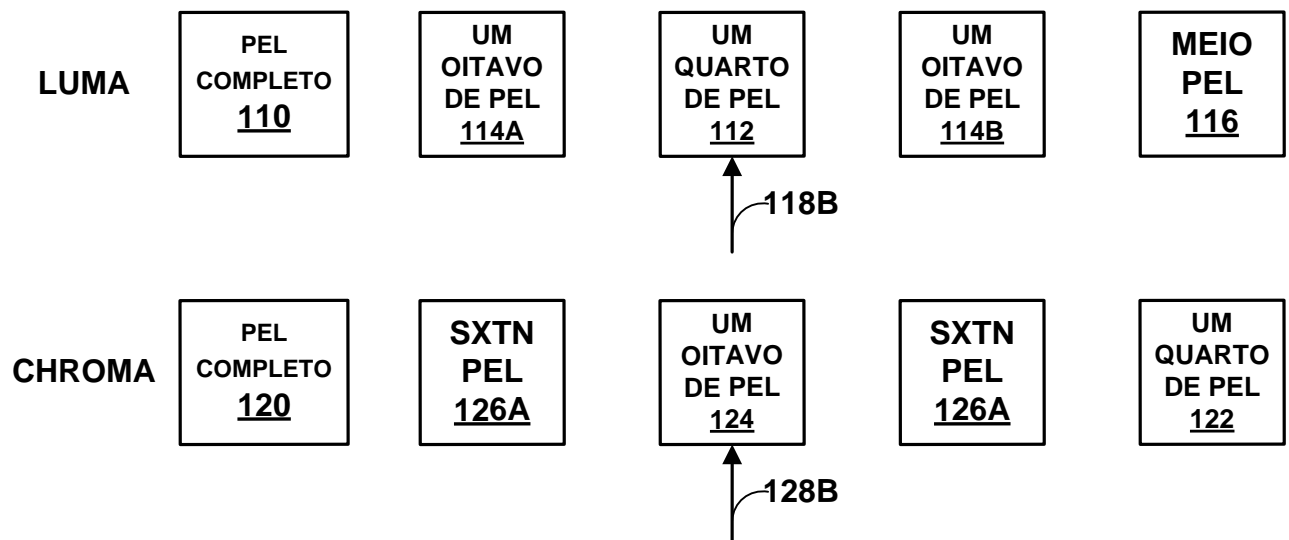


FIG. 5B

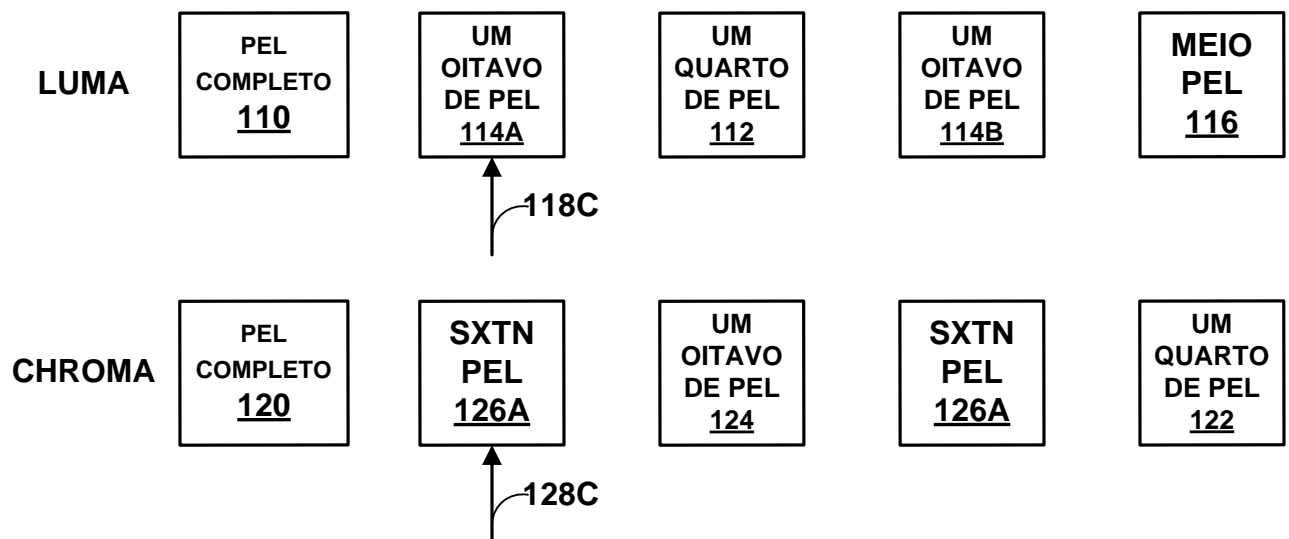


FIG. 5C

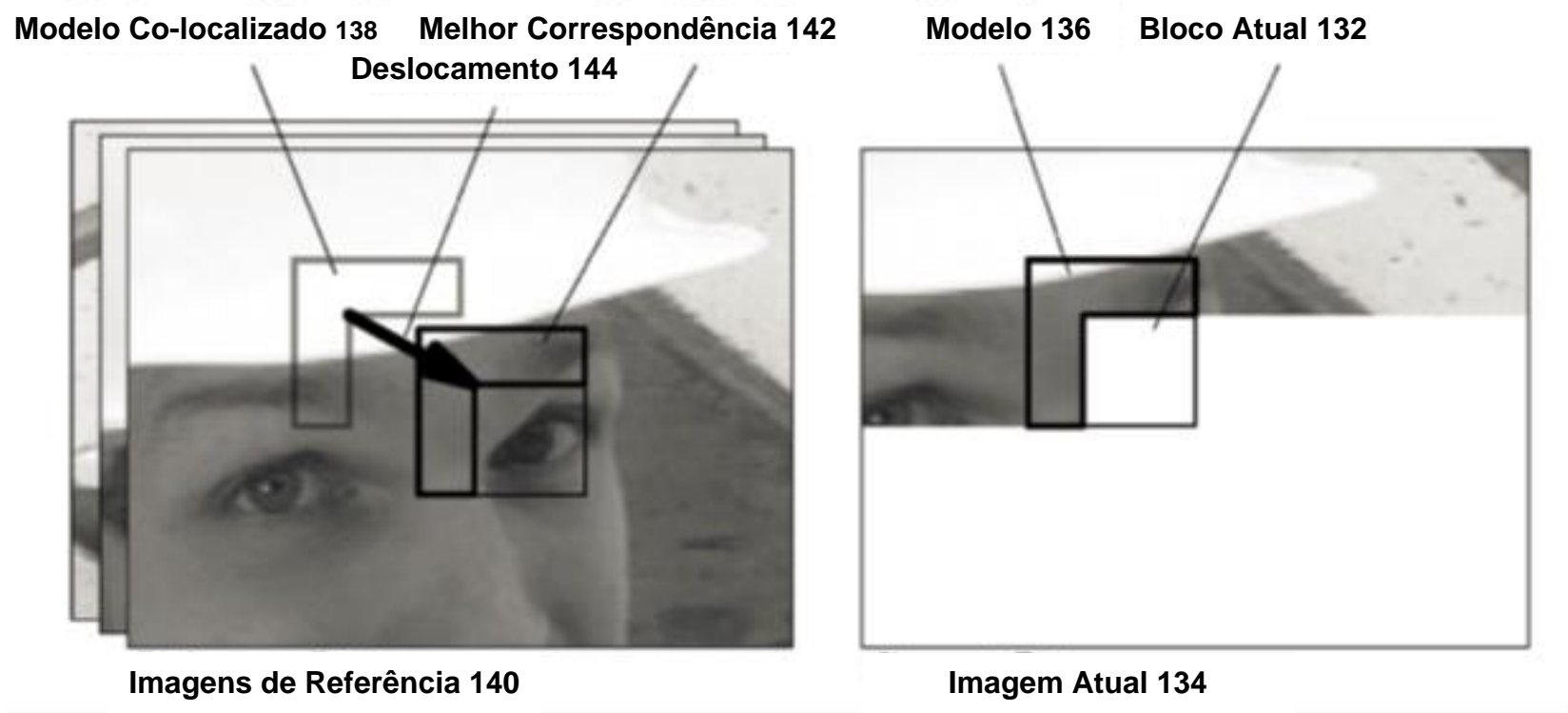


FIG. 6

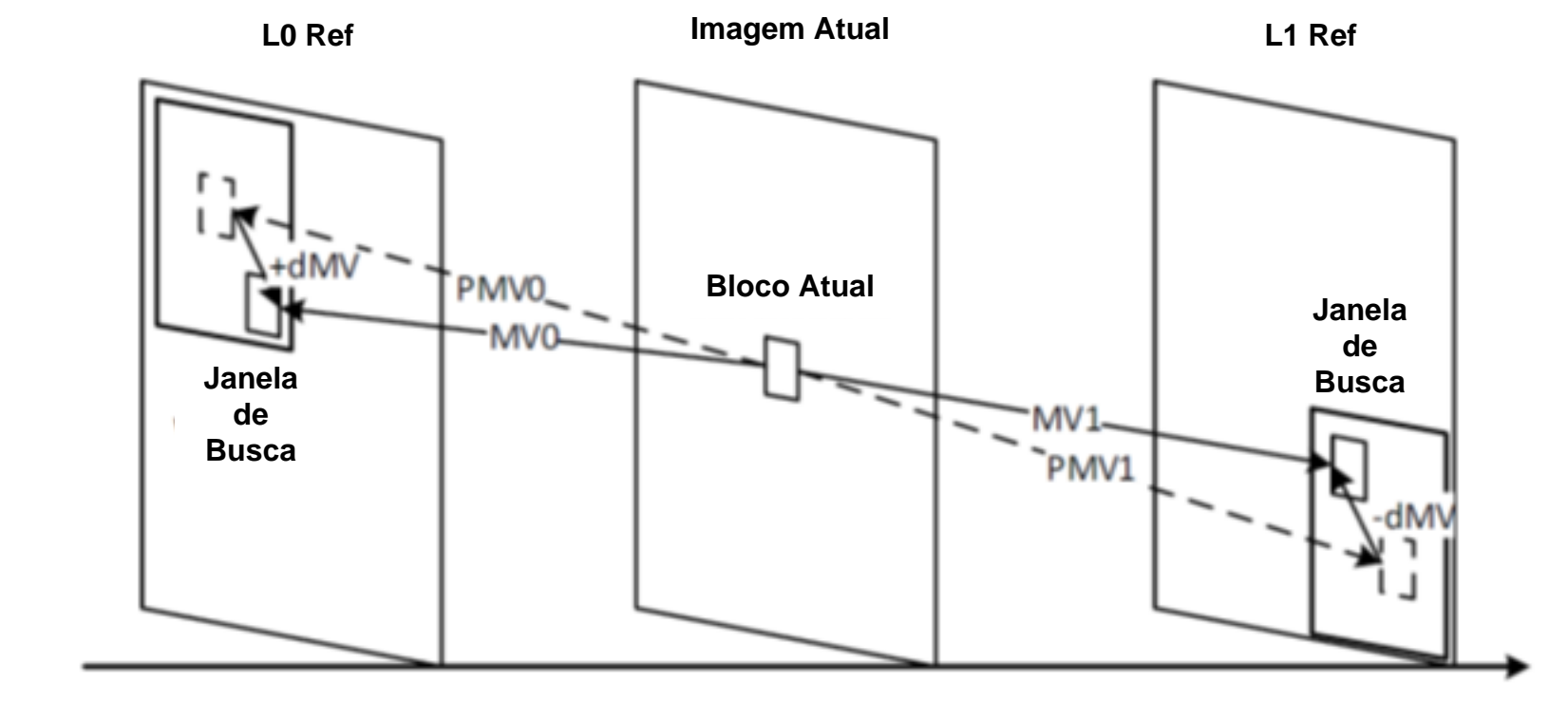


FIG. 7

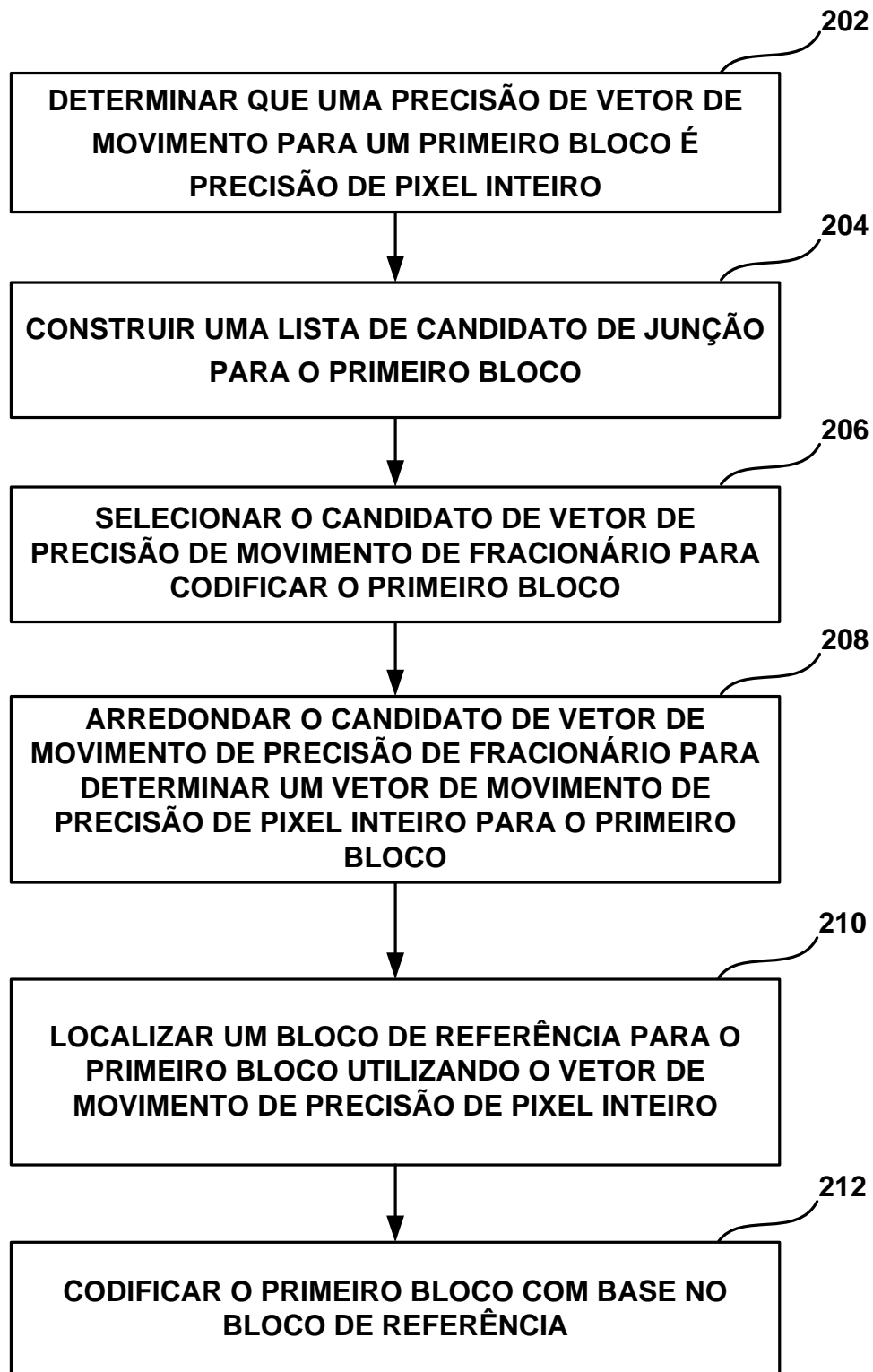


FIG. 8

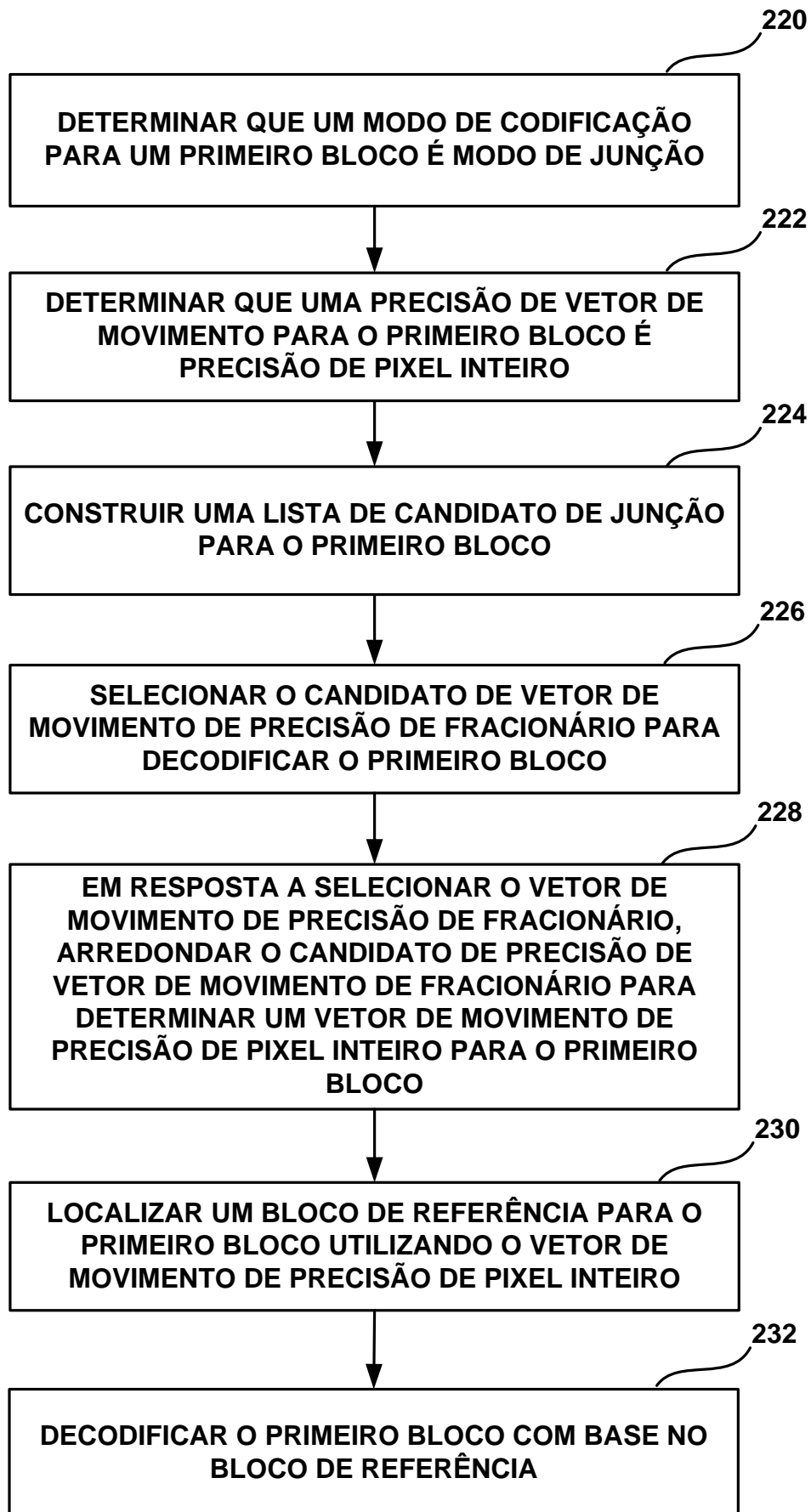


FIG. 9

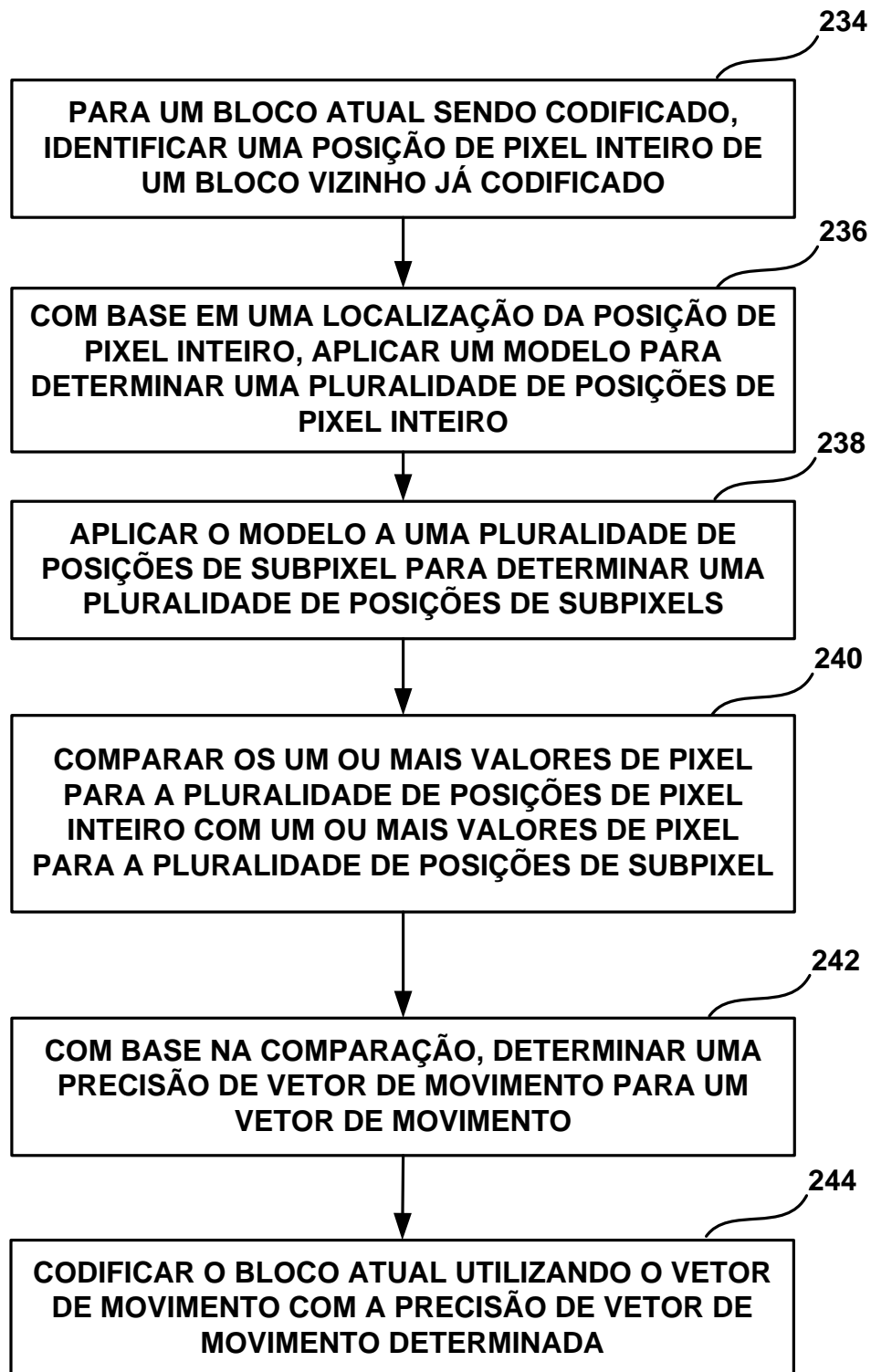


FIG. 10

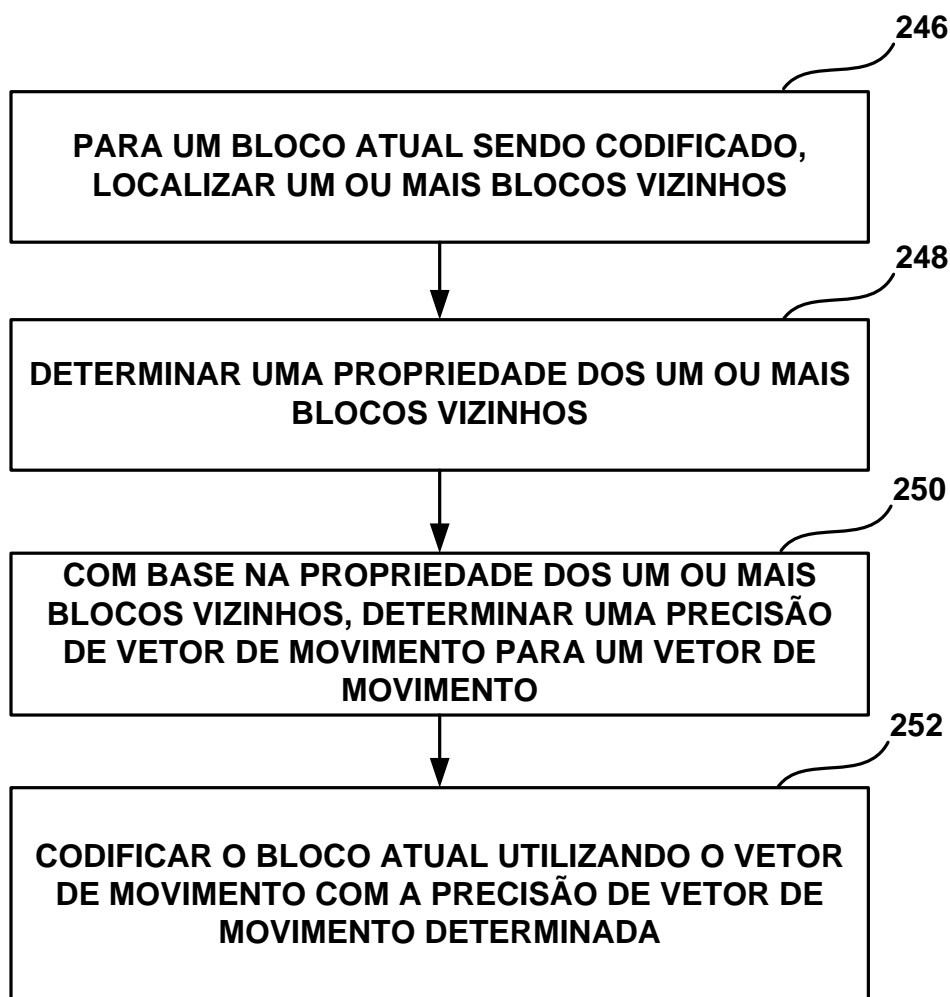


FIG. 11

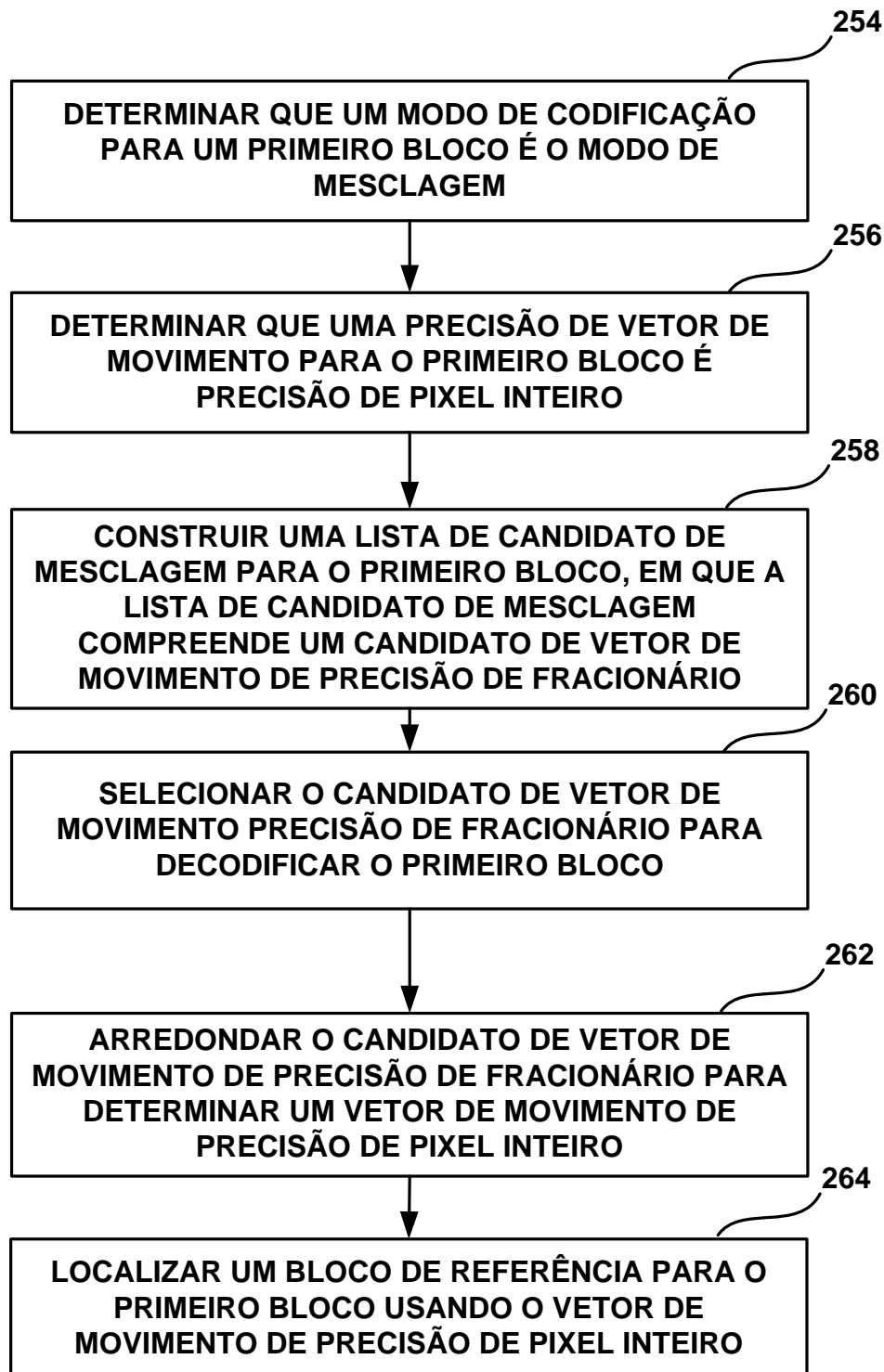


FIG. 12

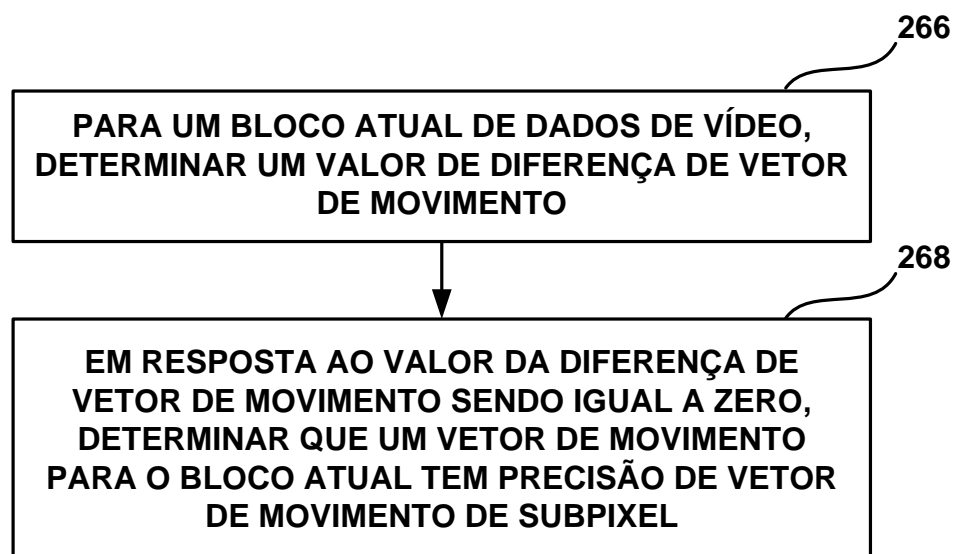


FIG. 13

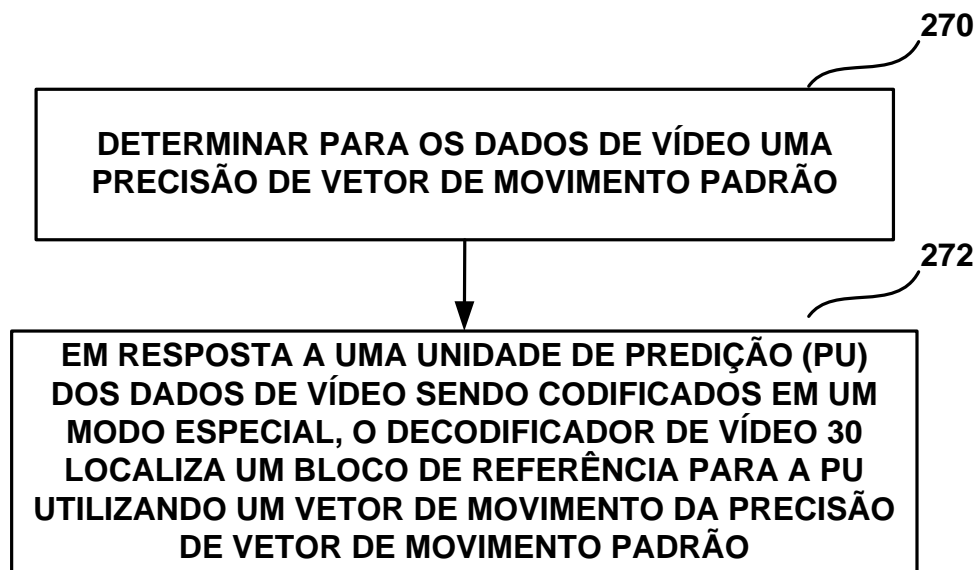
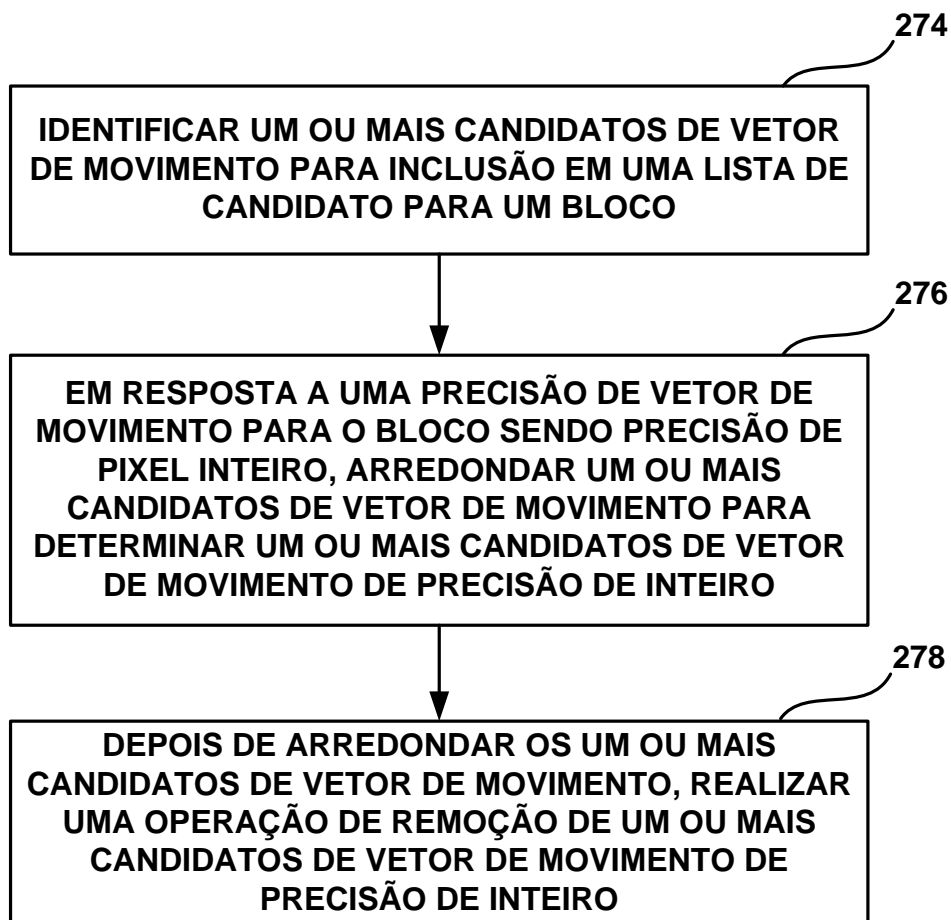


FIG. 14

**FIG. 15**

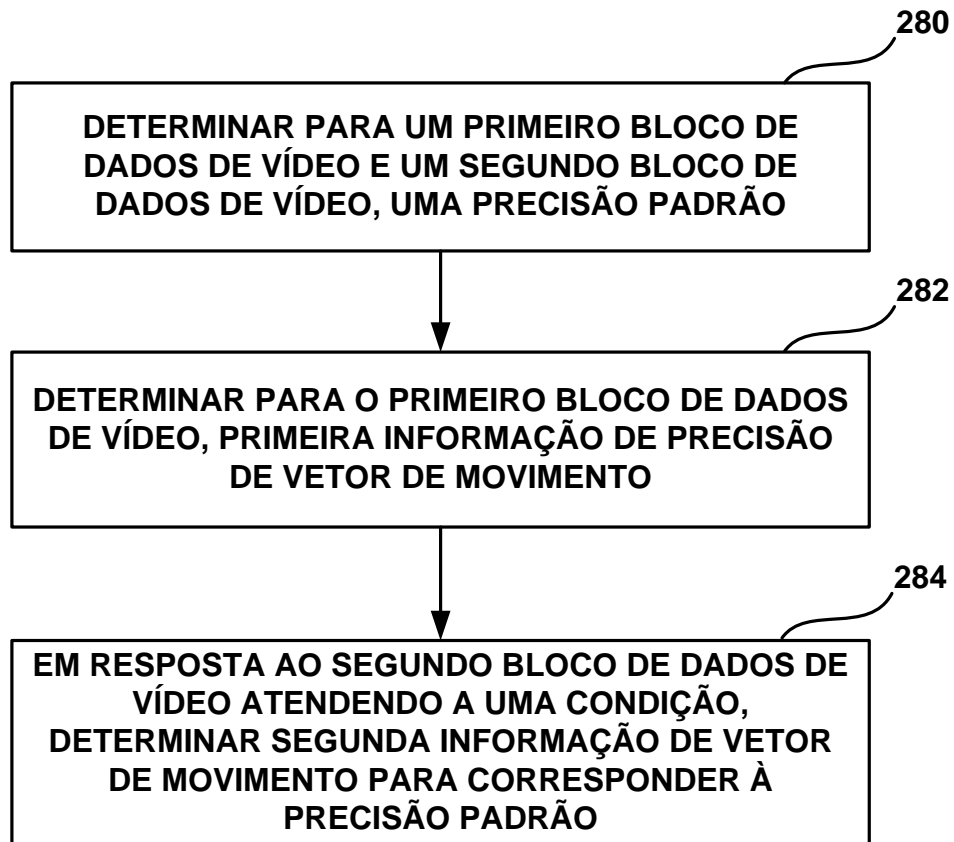


FIG. 16

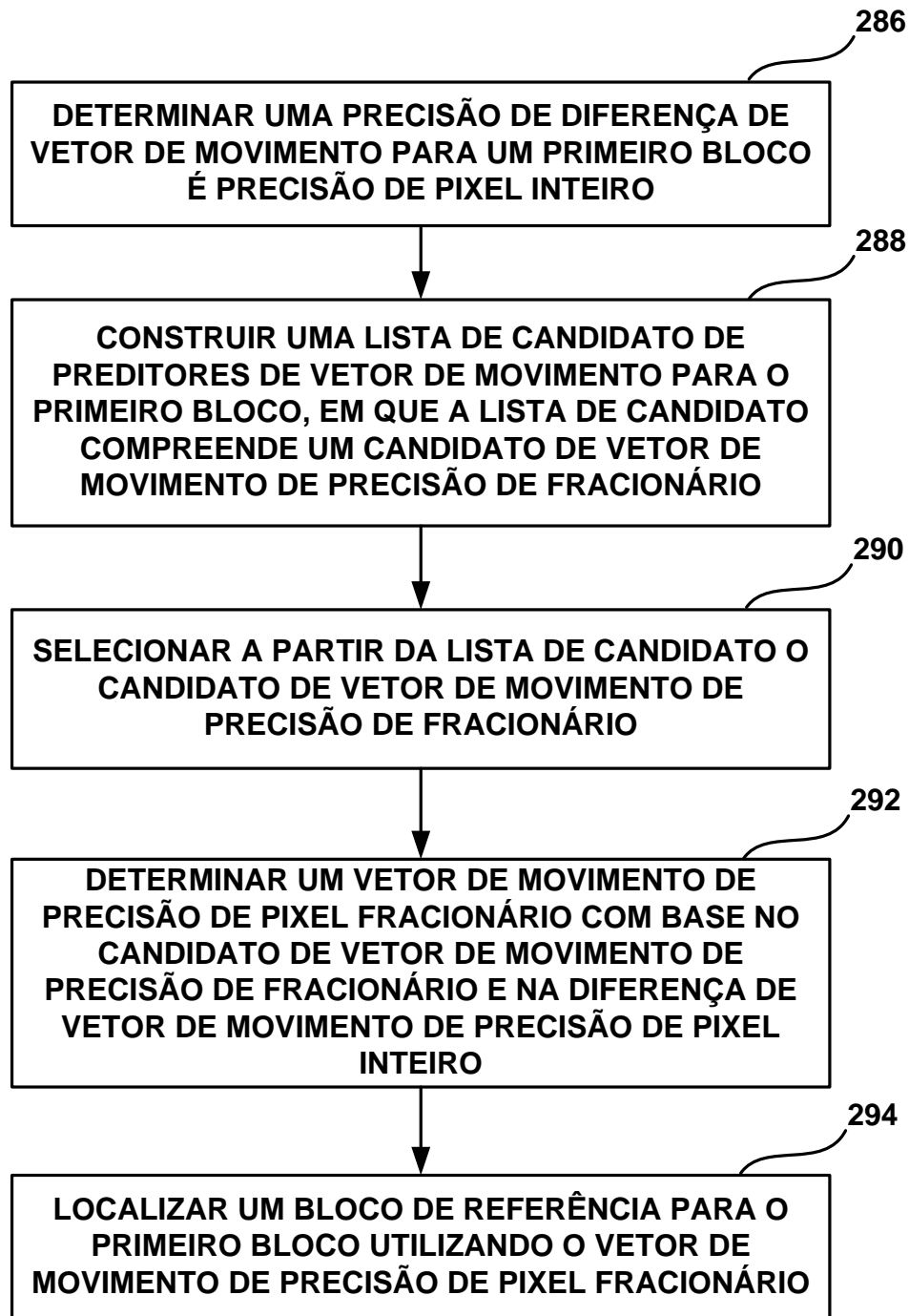


FIG. 17