

(22) Data de pedido: **2012.06.29**

(30) Prioridade(s): **2011.07.01 US**  
**201161504150 P 2011.11.01 US**  
**201161554398 P 2012.06.28 US**  
**201213536218**

(43) Data de publicação do pedido: **2014.05.07**

(45) Data e BPI da concessão: **2015.07.15**  
**208/2015**

(73) Titular(es):

**QUALCOMM INCORPORATED**  
**5775 MOREHOUSE DRIVE SAN DIEGO, CA**  
**92121-1714 US**

(72) Inventor(es):

**MARTA KARCZEWICZ US**  
**WEI-JUNGCHIEN US**  
**PEISONG CHEN US**

(74) Mandatário:

**LUÍS MANUEL DE ALMADA DA SILVA CARVALHO**  
**RUA VÍCTOR CORDON, 14 1249-103 LISBOA PT**

(54) Epígrafe: **CODIFICAÇÃO DE VÍDEO USANDO RESOLUÇÃO DE VECTORES DE MOVIMENTO ADAPTATIVO**

(57) Resumo:

UM DISPOSITIVO DE CODIFICAÇÃO DE VÍDEO COMPREENDE UM CODIFICADOR DE VÍDEO CONFIGURADO PARA CODIFICAR INFORMAÇÃO REPRESENTATIVA DE SE UM VALOR ABSOLUTO DE UM COMPONENTE X DE UM VALOR DA DIFERENÇA DO VECTOR DE MOVIMENTO PARA UM BLOCO ACTUAL DE DADOS DE VÍDEO É MAIOR DO QUE ZERO, CODIFICAR INFORMAÇÃO REPRESENTATIVA DE SE UM VALOR ABSOLUTO DE UM COMPONENTE Y DO VALOR DA DIFERENÇA DO VECTOR DE MOVIMENTO É MAIOR DO QUE ZERO, QUANDO O VALOR ABSOLUTO DO COMPONENTE X É MAIOR DO QUE ZERO, CODIFICAR INFORMAÇÃO REPRESENTATIVA DO VALOR ABSOLUTO DO COMPONENTE X, QUANDO O VALOR ABSOLUTO DO COMPONENTE Y É MAIOR DO QUE ZERO, CODIFICAR A INFORMAÇÃO REPRESENTATIVA DO VALOR ABSOLUTO DO COMPONENTE Y, QUANDO O VALOR ABSOLUTO DO COMPONENTE X É MAIOR DO QUE ZERO, CODIFICAR UM SINAL DO COMPONENTE X, E QUANDO O VALOR ABSOLUTO DO COMPONENTE Y É MAIOR DO QUE ZERO, CODIFICAR UM SINAL DO COMPONENTE Y.

**RESUMO****"CODIFICAÇÃO DE VÍDEO USANDO RESOLUÇÃO DE VECTORES DE MOVIMENTO ADAPTATIVOS"**

Um dispositivo de codificação de vídeo compreende um codificador de vídeo configurado para codificar informação representativa de se um valor absoluto de um componente x de um valor da diferença do vector de movimento para um bloco actual de dados de vídeo é maior do que zero, codificar informação representativa de se um valor absoluto de um componente y do valor da diferença do vector de movimento é maior do que zero, quando o valor absoluto do componente x é maior do que zero, codificar informação representativa do valor absoluto do componente x, quando o valor absoluto do componente y é maior do que zero, codificar a informação representativa do valor absoluto do componente y, quando o valor absoluto do componente x é maior do que zero, codificar um sinal do componente x, e quando o valor absoluto do componente y é maior do que zero, codificar um sinal do componente y.

**DESCRIÇÃO****"CODIFICAÇÃO DE VÍDEO USANDO RESOLUÇÃO DE VECTORES DE MOVIMENTO ADAPTATIVOS"**

## Campo técnico

A presente descrição refere-se à codificação de vídeo, e mais particularmente, à codificação de dados de vídeo para a compensação de movimento na codificação de vídeo.

## Antecedentes

As capacidades do vídeo digital podem ser incorporadas numa ampla gama de dispositivos, incluindo televisores digitais, sistemas de radiodifusão digital directos, sistemas de transmissão sem fios, assistentes digitais pessoais (PDAs), computadores portáteis ou de mesa, câmaras digitais, dispositivos de gravação digital, leitores multimédia digitais, dispositivos de jogos de vídeo, consolas de jogos de vídeo, radiotelefonos celulares ou via satélite, dispositivos de teleconferência de vídeo, e semelhantes. Os dispositivos de vídeo digitais implementam técnicas de compressão de vídeo, tais como aquelas descritas nas normas definidas pela MPEG-2, MPEG-4, ITU-T H.263 ou ITU-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.

As técnicas de compressão de vídeo executam a previsão espacial e/ou temporal para reduzir ou eliminar a redundância inerente às sequências de vídeo. Para a codificação de vídeo baseada em bloco, uma trama ou fatia de vídeo pode ser dividida em macroblocos. Cada macrobloco pode ser adicionalmente dividido. Os macroblocos numa trama ou fatia intra-codificada (I) são codificados usando a previsão espacial em relação a macroblocos vizinhos. Os macroblocos numa trama ou fatia inter-codificada (P ou B) podem usar a previsão espacial em relação a macroblocos vizinhos na mesma trama ou fatia ou previsão temporal em relação a outras tramas de referência.

O artigo "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard" por Marpe et al publicado em IEEE Transactions on Circuits and Systems, volume 13, número 7, páginas 620-636, de 1 de Julho de 2003 descreve uma técnica para codificar o bit de sinal para um componente x- ou y- juntamente com um valor representativo da grandeza do valor mvd para aquele componente x- ou y-.

A US 2005/038837 descreve um método e dispositivo para a binarização e codificação aritmética de um valor de dados.

O jornal "CE12: Adaptive Motion Vector Resolution from Qualcomm" 4ª reunião JCT-VC; 95. reunião MPEG número JCTVC-D394, de 16 de Janeiro de 2011 descreve um método adaptativo de resolução do vector de movimento dentro do âmbito da experiência do núcleo 12 que foi criado na 3ª reunião JCT-VC.

#### Sumário

A invenção encontra-se definida nas reivindicações às quais deve ser feita agora referência.

Os pormenores de um ou mais exemplos encontram-se estabelecidos nos desenhos anexos e na descrição seguinte. Outras características, objectos e vantagens da invenção serão visíveis da descrição e desenhos, e das reivindicações.

#### Breve descrição dos desenhos

Figura 1 diagrama de blocos que ilustra um exemplo de um sistema de codificação e descodificação de vídeo consistente com as técnicas desta descrição.

Figura 2 diagrama de blocos que ilustra um exemplo de um codificador de vídeo consistente com as técnicas desta descrição.

Figura 3 diagrama de blocos que ilustra um exemplo de um descodificador de vídeo consistente com as técnicas desta descrição.

Figura 4 fluxograma que ilustra um exemplo do método para a codificação conjunta dos componentes x e y de um valor da diferença do vector de movimento e codificação de entropia de um bloco actual.

Figura 5 fluxograma que ilustra um exemplo do método de sinalizar separadamente um valor da diferença do vector de movimento de acordo com o modelo de teste HEVC 3.0 (HM 3.0).

Figura 6 fluxograma que ilustra um exemplo do método para a codificação conjunta dos componentes x e y de um valor da diferença do vector de movimento.

Figura 7 fluxograma que ilustra um exemplo do método para a codificação conjunta dos componentes x e y de um valor da diferença do vector de movimento e codificação da resolução dos componentes x e y de um valor da diferença do vector de movimento.

Figura 8 fluxograma que ilustra um exemplo do método para a codificação conjunta dos componentes x e y de um valor da diferença do vector de movimento e codificação de dados codificados de entropia para produzir um bloco actual.

Figura 9 fluxograma que ilustra um exemplo do método para a codificação conjunta dos componentes x e y de um valor da diferença do vector de movimento.

#### Descrição pormenorizada

Em geral, esta descrição refere-se a técnicas para a codificação de dados para os valores da diferença

dos vectores de movimento durante a codificação de vídeo. Os dados de vídeo incluem uma sequência de tramas (ou imagens) reproduzidas em rápida sucessão para simular movimento. Cada trama de dados de vídeo pode ser dividida em blocos. Durante a codificação de vídeo, e mais especificamente inter-previsão, um dispositivo, tal como um codificador de vídeo ou um descodificador de vídeo, pode prever os valores de pixel de um bloco. O codificador ou descodificador de vídeo podem basear as suas previsões para o bloco sobre os valores de pixel de um bloco de outra trama ou a partir dos valores de pixel de um bloco vizinho.

Para a intra-previsão, um codificador de vídeo pode indicar o deslocamento de um bloco de referência utilizando um vector de movimento. O vector de movimento pode ter um componente  $x$  e um componente  $y$ . Os componentes  $x$  e  $y$  do vector de movimento podem indicar um deslocamento com precisão de sub-pixel, tal como a precisão de um meio-pixel, precisão de um quarto de pixel, ou precisão de um oitavo de pixel. Para alcançar a precisão de subpixel, um codificador ou descodificador de vídeo pode utilizar uma técnica, tal como interpolação, para determinar os valores de sub-pixeis nos locais indicados pelo vector de movimento. Depois de determinar os componentes  $x$  e  $y$  do vector de movimento, um codificador de vídeo pode calcular os valores de diferença do vector de movimento (MVD) para os componentes do vector de movimento relativo a um preditor de vector de movimento.

Esta descrição proporciona geralmente

técnicas para a codificação conjunta de valores MVD de movimento. Nesta descrição, a codificação conjunta pode-se referir a técnicas de codificação em que as informações relacionadas com os componentes x e y de um valor da diferença do vector de movimento estão intercaladas, ao contrário de codificação de todas as informações relacionadas a um dos componentes x ou y antes da codificação de qualquer informação relacionada com o segundo componente do valor da diferença do vector de movimento. A codificação conjunta pode também referir-se à utilização de um valor para representar valores para ambos os componentes x e y. Além disso, esta descrição também descreve técnicas para a codificação de diferenças de vector de movimento, que podem ter várias precisões de sub-pixel, por exemplo precisão de quarto de pixel ou precisão de oitavo de pixel, e indicando as precisões das diferenças do vector de movimento e vectores de movimento associados aos vectores de movimento.

Uma sequência de vídeo inclui uma ou mais tramas ou imagens. Cada uma das imagens pode ser dividida em um ou mais blocos, cada um dos quais pode ser individualmente codificado. Estão a ser feitos esforços para produzir um novo padrão de codificação, actualmente designado como codificação de vídeo de alta eficiência (HEVC), e por vezes designado como ITU H.265. Esta próxima norma refere-se a uma unidade de codificação (CU) na forma de um determinado bloco de píxeis incluindo dados de luminância e dados de crominância, onde os dados de luminância têm uma resolução

de  $2N \times 2N$  e os dados de crominância têm uma resolução de  $N \times N$ . Uma unidade de codificação pode ser dividida em quatro unidades, quadradas, de sub-codificação com o mesmo tamanho que não se sobrepõem.

Cada uma das unidades de sub-codificação pode também ser dividida desta maneira em outras unidades de sub-codificação. Uma unidade de codificação que não esteja dividida em unidades de sub-codificação é designada como unidade de codificação folha-nó terminal. As unidades de codificação folha-nó terminal podem incluir unidades de previsão (PUs) e unidades de transformação (TUs), em que as PUs representam os dados de previsão e TUs representam os dados residuais de codificação, ou seja, codificados, diferenças pixel-por-pixel entre os dados de previsão e os dados originais, não codificados, para os píxeis correspondentes a uma TU. Como exemplo, as PUs podem ser codificadas usando um modo de inter-previsão, em que um codificador de vídeo pode calcular um vector de movimento para uma PU usando um processo de estimativa de movimento. O codificador de vídeo pode sinalizar adicionalmente um modo de codificação para a PU e os valores de diferença do vector de movimento para o vector de movimento calculado, tal como descrito nesta descrição.

Da mesma forma, um descodificador de vídeo pode utilizar a informação indicadora de um modo de previsão incluído num fluxo de bits codificado para formar dados de previsão para blocos codificados. Os dados podem ainda

incluir uma precisão do vector de movimento, bem como uma indicação de uma posição fraccionada de pixel para a qual o vector de movimento aponta (por exemplo, uma posição de um oitavo de pixel de uma trama de referência ou fatia de referência).

Um dispositivo de codificação de vídeo, tal como um codificador de vídeo ou um decodificador de vídeo, pode determinar um vector de movimento preditivo para uma unidade de codificação (tal como uma trama, fatia, ou bloco). O dispositivo de codificação de vídeo pode utilizar componentes x e y do vector de movimento preditivo, (também designado como um preditor de vector de movimento) para calcular os valores de diferença do vector de movimento de vector de movimento para o bloco actual. Um dispositivo de codificação de vídeo pode implementar as técnicas desta descrição para codificar em conjunto os valores da diferença de vectores de movimento em vectores de movimento, que podem possuir uma precisão de subpixel.

Tal como descrito acima, os dados de previsão de uma PU para a formação de um bloco preditivo podem ser baseados em dados previamente codificados de CUs espacialmente vizinhas ou CUs de tramas temporalmente vizinhas que tenham sido previamente codificadas. Um codificador de vídeo pode calcular um vector de movimento de uma PU que indica a localização de um bloco preditivo para uma CU correspondente numa trama temporalmente separada, previamente codificada. O codificador de vídeo

pode ainda calcular um valor de diferença do vector de movimento para o vector de movimento para codificar o vector de movimento. O valor da diferença do vector de movimento pode em geral corresponder a uma diferença entre o vector de movimento calculado, e um preditor de vector de movimento. O vector de movimento para o bloco actual pode ter um componente x (MV\_x) para um deslocamento horizontal, e um componente y (MV\_y) para um deslocamento vertical. O preditor do vector de movimento pode ter um componente x de p\_x e um componente y de p\_y. Depois, o preditor do vector de movimento pode ser determinado como  $\langle MV_x - p_x, MV_y - p_y \rangle$ .

Esta descrição proporciona técnicas para a codificação conjunta do componente x e componente y de um valor da diferença do vector de movimento. Nesta descrição, a codificação conjunta pode-se referir a técnicas de codificação em que as informações relacionadas com os componentes x e y de um valor da diferença do vector de movimento estão intercaladas. A informação relacionada com a diferença de vector de movimento pode incluir um valor que representa se ou não um componente da diferença do vector de movimento é zero, um ou mais valores que representam o valor de sinal dos componentes x- e/ou y-, e um ou mais valores representativos do valor absoluto dos componentes x e y do valor da diferença do vector de movimento, como alguns exemplos.

De acordo com as técnicas desta descrição, a

informação relacionada com um ou ambos os componentes de um valor da diferença do vector de movimento pode ser codificada de um modo alternativo de modo a que todas as informações relativas a um componente da diferença do vector de movimento não necessitam de ser codificadas antes da codificação da informação relacionada com o outro componente da diferença do vector de movimento. Por outras palavras, um elemento do componente x, tal como se o componente x tem uma magnitude maior do que zero, pode ser intercalado com um elemento correspondente do componente y. A codificação conjunta do componente x e y da diferença do vector de movimento contrasta com a codificação separada de toda a informação relativa a um dos componentes x ou y de um valor da diferença do vector de movimento, por exemplo, pela convenção em ITU-T H.264. A codificação conjunta dos componentes x e y dos valores de diferença do vector de movimento pode reduzir a complexidade de um fluxo de bits de codificação de vídeo, o que pode resultar numa taxa de bits menor para vídeo codificado, bem como outras vantagens potenciais aqui descritas.

Na codificação de vídeo de acordo com a norma ITU-T H.264 convencional, os vectores de movimento podem ter um quarto de pixel (ou seja, um quarto do pixel) de precisão. Em alguns casos, a precisão de um oitavo pixel (isto é, um oitavo de pixel) pode proporcionar certas vantagens. O modelo de teste de codificação de vídeo de alta eficiência (HM) tem a capacidade de formar unidades de previsão usando vectores de movimento com uma precisão de

oitavo de pixel. Em particular, o HM proporciona a capacidade de seleccionar de forma adaptativa seja a precisão um quarto de pixel ou precisão de um oitavo de pixel. Deste modo, os vectores de movimento podem ter precisão adaptativa, também designada como uma resolução de vector de movimento. Esta precisão de pixel fraccionada, ou sub-inteiro pode ser utilizada para definir um vector de movimento em relação a um bloco de valores de pixel interpolados na trama anteriormente codificada.

Esta descrição também proporciona técnicas para a selecção de casos particulares em que podem ser utilizados vectores de movimento de precisão de um quarto ou um oitavo de pixel, e como a precisão do vector de movimento pode ser sinalizada para um vector de movimento em particular. Em particular, as técnicas da presente descrição são dirigidas para seleccionar uma resolução de vector de movimento, e sinalizar a resolução usando em alguns casos uma referência de resolução de movimento. Essa descrição proporciona também técnicas para a sinalização conjunta dos componentes x e y de um vector de movimento particular.

Este documento descreve várias técnicas relacionadas com a codificação do vector de movimento, tal como a codificação conjunta dos valores da diferença do vector de movimento e sinalizar a precisão, amplitude e sinal de um ou mais valores da diferença dos vectores de movimento. As técnicas desta descrição podem ser executadas durante um processo de codificação executado por um

dispositivo de codificação de vídeo, tal como um codificador de vídeo ou um descodificador de vídeo. Nesta descrição, o termo "codificação" refere-se a codificação que ocorre no codificador ou descodificação que ocorre no descodificador. De modo semelhante, o termo codificador refere-se a um codificador, um descodificador, ou um codificador/descodificador combinado (CODEC). Os termos codificador, descodificador e CODEC referem-se todos a máquinas específicas projectadas para a codificação (codificação e/ou descodificação) de dados de vídeo compatíveis com a presente descrição.

A figura 1 é diagrama de blocos que ilustra um exemplo de um sistema de codificação e descodificação de vídeo 10 que pode utilizar técnicas para sinalizar em conjunto vectores de movimento. Os vectores de movimento podem ter precisão de sub-pixel. Tal como mostrado na figura 1, o sistema 10 inclui um dispositivo de origem 12 que transmite vídeo codificado para um dispositivo de destino 14 através de um canal de comunicações 16. O dispositivo de origem 12 e dispositivo de destino 14 podem compreender qualquer de uma ampla gama de dispositivos. Em alguns casos, o dispositivo de origem 12 e dispositivo de destino 14 podem compreender dispositivos de comunicação sem fios, tais como telefones sem fios, os denominados telemóveis ou radiotelefonos de satélite, ou quaisquer dispositivos sem fios que podem comunicar informação de vídeo através de um canal de comunicação 16, sendo que neste caso o canal de comunicação 16 é sem fios. As

técnicas desta descrição, no entanto, que se referem em conjunto a codificação, por exemplo, a sinalização conjunta e/ou interpretar um valor sinalizado em conjunto, valores da diferença do vectores de movimento, que podem ter precisão de sub-pixel, não estão necessariamente limitados a aplicações ou configurações sem fios. Por exemplo, estas técnicas podem ser aplicadas a transmissões de televisão sem fios, transmissões de televisão por cabo, transmissões de televisão por satélite, transmissões de vídeo via Internet, vídeo digital codificado que é codificado num meio de armazenamento, ou outros cenários. Por conseguinte, o canal de comunicação 16 pode compreender qualquer combinação de meio sem fios, com fios, ou de armazenamento adequado para a transmissão ou armazenamento de dados de vídeo codificados.

No exemplo da figura 1, o dispositivo de origem 12 inclui uma fonte de vídeo 18, um codificador de vídeo 20, um modulador/desmodulador (modem) e um transmissor 24. O dispositivo de destino 14 inclui um receptor 26, um modem 28, um descodificador de vídeo 30, e um dispositivo de visualização 32. De acordo com esta descrição, o codificador de vídeo 20 do dispositivo de origem 12 pode ser configurado para aplicar as técnicas para sinalizar as diferenças do vector de movimento, que podem apresentar uma precisão de subpixel. Em outros exemplos, um dispositivo de origem e um dispositivo de destino podem incluir outros componentes ou dispositivos. Por exemplo, o 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 de visualização externo, em vez de incluir um dispositivo de visualização integrado.

O sistema 10 ilustrado da figura 1 é simplesmente um exemplo. As técnicas para sinalizar em conjunto a precisão das diferenças do vector de movimento podem ser executadas por qualquer dispositivo de codificação e/ou descodificação de vídeo digital. Embora geralmente as técnicas desta descrição sejam realizadas por um dispositivo de codificação de vídeo, as técnicas podem também ser realizadas por um codificador/descodificador de vídeo, tipicamente designado como um "CODEC". Além disso, as técnicas desta descrição também podem ser realizadas por um pré-processador de vídeo. O dispositivo de origem 12 e dispositivo de destino 14 são simplesmente exemplos de tais dispositivos de codificação em que 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 uma maneira substancialmente simétrica, de tal modo que cada um dos dispositivos 12, 14 inclui componentes de codificação e descodificação de vídeo. Por isso, o sistema 10 pode suportar a transmissão de vídeo unidireccional entre dispositivos de vídeo 12, 14, por exemplo, para transmissão de vídeo, reprodução de vídeo, difusão de vídeo ou telefonia de vídeo.

A fonte de vídeo 18 do dispositivo de origem 12

pode incluir um dispositivo de captura de vídeo, tal como uma câmara de vídeo, um arquivo de vídeo contendo vídeo capturado anteriormente, e/ou uma alimentação de vídeo a partir de um fornecedor de conteúdo de vídeo. Como alternativa adicional, a fonte de vídeo 18 pode gerar dados baseados em gráficos de computador como a fonte do vídeo, ou uma combinação de vídeo em directo, vídeo arquivado, e vídeo gerado por computador. Em alguns casos, se a fonte de vídeo 18 for uma câmara de vídeo, dispositivo de origem 12 e dispositivo de destino 14 podem formar os denominados telefones com câmara ou telefones de vídeo. Tal como mencionado acima, no entanto, as técnicas apresentadas nesta descrição podem ser aplicáveis à codificação de vídeo, em geral, e podem ser aplicadas a aplicações sem fios e/ou com fios. Em qualquer dos casos, o vídeo capturado, pré-capturado, ou gerado pelo computador pode ser codificado pelo codificador de vídeo 20. A informação de vídeo codificada pode depois ser modulada pelo modem 22 de acordo com uma norma de comunicação, e transmitida para o dispositivo de destino 14 através do transmissor 24. O modem 22 pode incluir vários misturadores, filtros, amplificadores ou outros componentes projectados para modulação do sinal. O transmissor 24 pode incluir circuitos projectados para transmitir dados, incluindo amplificadores, filtros, e uma ou mais antenas.

O receptor 26 do dispositivo de destino 14 recebe informação pelo canal 16, e o modem 28 desmodula a informação. Novamente, o processo de codificação de vídeo

pode implementar uma ou mais das técnicas descritas na presente para sinalizar as diferenças do vector de movimento, que podem apresentar uma precisão de subpixel. A informação comunicada pelo canal 16 pode incluir informação de sintaxe definida pelo codificador de vídeo 20, que também é usada pelo descodificador de vídeo 30, que inclui elementos de sintaxe que descrevem características e/ou processamento de macroblocos e outras unidades codificadas, por exemplo, GOPs. O dispositivo de visualização 32 exhibe a um utilizador os dados de vídeo descodificados, e pode compreender qualquer um de uma variedade de dispositivos de visualização tal como um tubo de raios catódicos (CRT), um ecrã de cristais líquidos (LCD), um ecrã de plasma, um ecrã de díodos orgânicos emissores de luz (OLED), ou outro tipo de dispositivo de visualização.

No exemplo da figura 1, o canal de comunicação 16 pode compreender qualquer meio de comunicações sem ou com fios, tal como um espectro de frequência de rádio (RF) ou uma ou mais linhas de transmissão física, ou qualquer combinação dos meios de comunicação sem e com fios. O canal de comunicação 16 pode fazer parte de uma rede baseada em pacotes, tal como uma rede de área local, uma rede alargada, ou uma rede global, tal como a Internet. O canal de comunicação 16 representa geralmente qualquer meio de comunicação adequado, ou colecção de diferentes meios de comunicação, para a transmissão de dados de vídeo do dispositivo de origem 12 para o dispositivo de destino 14, incluindo qualquer combinação apropriada de meios com ou

sem fios. O canal de comunicação 16 pode incluir routers, comutadores, estações de base, ou qualquer outro equipamento que possa ser útil para facilitar a comunicação do dispositivo de origem 12 para o dispositivo de destino 14. Em outros exemplos, o dispositivo de origem 12 pode armazenar dados codificados num meio de armazenamento, em vez de transmitir os dados. Do mesmo modo, o dispositivo de destino 14 pode ser configurado para recuperar dados codificados a partir de um meio de armazenamento.

O codificador de vídeo 20 e descodificador de vídeo 30 podem operar de acordo com uma norma de compressão de vídeo, como a próxima norma ITU-T High Efficiency Video Coding (HEVC), também designada como "H.265". A HEVC ainda não está pronta, sendo que deste modo o codificador de vídeo 20 e descodificador 30 podem funcionar de acordo com o mais recente esboço de HEVC, designado como o High Efficiency Video Coding Test Model (HM). As técnicas da presente descrição não se encontram contudo limitadas a qualquer norma de codificação particular. Outros exemplos incluem o MPEG-2 e ITU-T H.263, e ITU-T H.264. Embora não seja mostrado na figura 1, em alguns aspectos, o codificador de vídeo 20 e descodificador de vídeo 30 podem cada um ser integrado com um codificador e descodificador de áudio, e podem incluir unidades apropriadas MUX-DEMUX, ou outro hardware e software, para lidar com a codificação de ambos o áudio e vídeo num fluxo de dados comum ou fluxos de dados separados. Se aplicável, as unidades MUX-DEMUX podem estar de acordo com o protocolo multiplexador ITU

H.223, ou outros protocolos tal como o protocolo UDP.

A norma HEVC está a ser actualmente formulada pelo grupo de peritos de codificação de vídeo ITU-T (VCEG) juntamente com o grupo de peritos de imagem em movimento ISO/IEC (MPEG) como o produto de uma parceria colectiva conhecido como a equipa de colaboração conjunta sobre codificação de vídeo (JCT -VC). O HM pressupõe várias capacidades de dispositivos de codificação de vídeo sobre dispositivos de acordo com normas de codificação anteriores, tais como ITU-T H.264/AVC. Por exemplo, enquanto H.264 oferece nove modos de codificação intra-previsão, HM oferece até trinta e quatro modos de codificação intra-previsão.

O codificador de vídeo 20 e descodificador de vídeo 26 podem cada um ser implementados como qualquer um de uma variedade de circuitos codificadores apropriados, tais como um ou mais processadores, processadores de sinal digital (DSPs), circuitos integrados para aplicação específica (ASIC)s, redes de portas lógicas programáveis (FPGAs), lógica discreta, software, hardware, firmware ou qualquer combinação dos mesmos. Cada um dos codificadores de vídeo 20 e descodificadores de vídeo 30 pode ser incluído em um ou mais codificadores ou descodificadores, cada um dos quais pode ser integrado como parte de um codificador/descodificador combinado (CODEC) numa respectiva câmara, computador, dispositivo móvel, dispositivo de assinante, dispositivo de transmissão,

descodificador de televisão, servidor, ou outros semelhantes.

Uma sequência de vídeo inclui tipicamente uma série de tramas de vídeo. Um grupo de imagens (GOP) compreende geralmente uma série de uma ou mais tramas de vídeo. Um GOP pode incluir dados de sintaxe num cabeçalho do GOP, um cabeçalho de uma ou mais tramas do GOP, ou em outro lugar, que descreve uma série de tramas incluídas no GOP. Cada trama pode incluir dados de sintaxe de trama que descreve um modo de codificação para a respectiva trama. O codificador de vídeo 20 opera tipicamente em blocos de vídeo, também designados como CUs, dentro de tramas de vídeo individuais de modo a codificar os dados de vídeo. Um bloco de vídeo pode corresponder a uma LCU ou uma partição de uma LCU. Os blocos de vídeo podem ter tamanhos fixos ou variados, e pode variar em tamanho de acordo com um padrão de codificação especificado. Cada trama de vídeo pode incluir uma pluralidade de fatias. Cada fatia pode incluir uma pluralidade de LCUs, que podem ser dispostos em partições, também designadas como sub-UC.

Como exemplo, a norma ITU-T H.264 suporta intra-previsão em vários tamanhos de bloco, tais como 16 por 16, 8 por 8, ou 4 por 4 para os componentes de luminância e 8x8 para componentes de crominância, bem como inter-previsão em vários tamanhos de bloco, tais como 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 e 4x4 de componentes de luminância e tamanhos dimensionados de modo correspondente para componentes de

crominância. Na presente descrição, "NxN" e "N por N" pode ser utilizado indiferentemente para designar as dimensões de pixel do bloco em termos de dimensão vertical e horizontal, por exemplo, 16x16 píxeis ou 16 por 16 píxeis. Em geral, um bloco de 16x16 terá 16 píxeis na direcção vertical ( $y = 16$ ) e 16 píxeis na direcção horizontal ( $x = 16$ ). Do mesmo modo, um bloco NxN tem geralmente N píxeis na direcção vertical e N píxeis na direcção horizontal, em que N representa um valor inteiro não negativo. Os píxeis num bloco podem ser dispostos em filas e colunas. Além disso, os blocos não têm necessariamente de ter o mesmo número de píxeis na direcção horizontal e na direcção vertical. Por exemplo, os blocos podem compreender NxM píxeis, em que M não é necessariamente igual a N.

HEVC refere-se a um bloco de dados de vídeo como uma unidade de codificação (CU), que pode incluir uma ou mais unidades de previsão (PUs) e/ou uma ou mais unidades de transformação (TUs). Os dados da sintaxe dentro de um fluxo de bits podem definir uma unidade de codificação (LCU) maior, que é a unidade de codificação maior em termos de número de píxeis. Em geral, uma CU tem um objectivo semelhante a um macrobloco de H.264, excepto que uma CU não tem uma distinção de tamanho. Deste modo, uma CU pode ser dividida em sub-CUs. Em geral, as referências nesta descrição a uma CU pode-se referir a uma maior unidade de codificação de uma imagem ou uma sub-CU de uma LCU. Uma LCU pode ser dividida em sub-CUs, e cada sub-CU pode ser adicionalmente dividida em sub-CUs. Os dados de sintaxe

para um fluxo de bits podem definir um número máximo de vezes que uma LCU pode ser dividida, designado como profundidade da CU. Por conseguinte, um fluxo de bits pode também definir uma unidade de codificação menor (SCU). Essa descrição utiliza também o termo "bloco" para se referir a qualquer uma de uma CU, PU, ou TU.

Uma LCU pode ser associada com uma estrutura de dados de árvore quaternária. Em geral, uma estrutura de dados de árvore quaternária inclui um nó por CU, onde um nó de raiz corresponde à LCU. Se uma CU for dividida em quatro sub-CUs, o nó que corresponde à CU inclui quatro nós de folhas, cada uma das quais corresponde a uma das sub-CUs. Cada nó da estrutura de dados da árvore quaternária pode proporcionar dados de sintaxe para a CU correspondente. Por exemplo, um nó na árvore quaternária pode incluir uma referência de divisão, que indica se a CU correspondente ao nó é dividida em sub-CUs. Os elementos de sintaxe para uma CU podem ser definidos recursivamente, e podem depender do facto da CU ser dividida em sub-CUs. Se uma CU não for adicionalmente dividida, é designada como folha-CU. Na presente descrição, 4 sub-CUs de uma folha-CU serão também designadas como folha-CUs embora não haja nenhuma divisão explícita da folha-CU original. Por exemplo se uma CU de tamanho 16x16 não for adicionalmente dividida, as quatro sub-CUs serão também designadas como folha-CUs embora a CU de 16x16 não tenha sido sempre dividida.

Além disso, as TUs das folha-CUs podem também

estar associadas às respectivas estruturas de dados de árvore quaternária. Isto é, uma folha-CU pode incluir uma árvore quaternária que indica como a folha-CU é dividida em TUs. Esta descrição refere-se à árvore quaternária que indica como uma LCU é dividida como uma árvore quaternária de CU, indicando a árvore quaternária como a folha-CU é dividida em TUs como uma árvore quaternária TU. O nó raiz de uma árvore quaternária TU corresponde geralmente a uma folha-CU, enquanto o nó raiz de uma árvore quaternária CU corresponde geralmente a uma LCU. As TUs da árvore quaternária TU que não são divididas são designadas como folha-TUs.

Uma folha-CU pode incluir uma ou mais unidades de previsão (PUs). Em geral, uma PU representa toda ou uma parte da CU correspondente, e pode incluir dados para obter uma amostra de referência para a PU. Por exemplo, quando a PU é codificada inter-modo, a PU pode incluir dados que definem um vector de movimento para a PU. Os dados que definem o vector de movimento podem descrever, por exemplo, um componente horizontal do vector de movimento, um componente vertical do vector de movimento, uma resolução para o vector de movimento (por exemplo, uma precisão de um quarto de pixel ou uma precisão de um oitavo de pixel), uma trama de referência para a qual o vector de movimento aponta, e/ou uma lista de referência (por exemplo, lista 0 ou lista 1) para o vector de movimento. Os dados para a folha-CU que definem a(s) PU(s) podem também descrever, por exemplo, a divisão da CU em uma ou mais PUs. Os modos de

divisão podem diferir dependendo se a CU não está codificada, codificada em modo intra-previsão, ou codificada em modo inter-previsão. Para a intra-codificação, uma PU pode ser tratada da mesma forma que uma unidade de transformação de folha descrita a seguir.

Uma folha-CU pode incluir uma ou mais unidades de transformação (TUs). As unidades de transformação podem ser especificadas utilizando uma estrutura de árvore quaternária TU, tal como descrito acima. Isto é, uma referência de divisão pode indicar se uma folha-CU está dividida em quatro unidades de transformação. Em seguida, cada unidade de transformação pode ser dividida ainda mais em 4 sub-TUs. Quando uma TU não se encontra adicionalmente dividida, pode ser designada como uma folha-TU. No geral, para a intra-codificação, todas as folha-TUs que pertencem a uma folha-CU partilham o mesmo modo de intra-previsão. Isto é, o mesmo modo intra-previsão é geralmente aplicado para calcular os valores previstos para todas as TUs de uma folha-CU. Para a intra-codificação, um codificador de vídeo pode calcular um valor residual para cada folha-TU utilizando o modo de intra-previsão, como uma diferença entre a parte dos valores preditivos que correspondem à TU e bloco original. O valor residual pode ser transformado, quantificado, e verificado com o scanner. Para inter-codificação, um codificador de vídeo pode executar a previsão ao nível da PU e pode calcular um residual para cada PU. Os valores residuais que correspondem a uma folha CU podem ser transformados, quantificados, e verificados

com o scanner. Para inter-codificação, uma folha-TU pode ser maior ou menor do que uma PU. Para a intra-codificação, uma PU pode ser colocada com uma folha-TU correspondente. Em alguns exemplos, a dimensão máxima de uma folha-TU pode ser o tamanho da folha-CU correspondente.

Em geral, esta descrição usa os termos CU e TU para se referir a folha-CU e folha-TU, respectivamente, salvo indicação em contrário. Em geral, as técnicas da presente descrição referem-se a transformação, quantificação, verificação com o scanner, e entropia e codificação de dados de uma CU. Como um exemplo, as técnicas da presente descrição incluem a selecção de uma transformação para usar a transformação de um valor residual de um bloco intra-previsto com base num modo intra-previsão utilizado para prever o bloco. Essa descrição também usa o termo "transformação direccional" ou "transformação projectada" para se referir a tal uma transformação que depende da direcção do modo intra-previsão. Quer dizer, um codificador de vídeo pode seleccionar uma transformação direccional para aplicar a uma unidade de transformação (TU). Tal como observado acima, a intra-previsão inclui prever uma TU de uma CU actual de uma imagem de CUs previamente codificadas e TUs da mesma imagem. Mais especificamente, um codificador de vídeo pode intra-prever uma TU actual de uma imagem utilizando um modo particular intra-previsão.

Durante os modos inter-previsão, o codificador de

vídeo 20 pode determinar uma resolução, como 1/4 (um quarto) ou 1/8 (um oitavo) de pixel de resolução, para um vector de movimento particular de uma PU. Esta resolução pode ser determinada usando uma técnica que tenta minimizar a quantidade de erros entre um preditor de vector de movimento, e um vector de movimento calculado para a PU da CU, que pode ser um vector de movimento utilizado para prever uma deslocação de uma trama previamente codificada.

O codificador de vídeo 20 pode determinar uma precisão para o preditor do vector de movimento, bem como uma deslocação do vector, que pode incluir um componente x,  $p_x$ , e um componente y,  $p_y$ . O codificador de vídeo 20 pode calcular uma diferença do vector de movimento para uma diferença entre o preditor do vector de movimento e o vector de movimento calculado. A diferença do vector de movimento (designado como um "MVD"), pode também ter um componente x,  $mvd_x$ , e um componente y,  $mvd_y$ . Com base na magnitude do  $mvd_x$ , e  $mvd_y$ , a precisão dos vectores (isto é, um quarto, ou um oitavo de precisão do pixel), e um valor limiar, o codificador de vídeo 20 pode ser configurado para codificar em conjunto um ou mais valores referentes aos componentes x e y do valor da diferença do vector de movimento.

De acordo com as técnicas da presente descrição, o codificador de vídeo 20 e/ou descodificador de vídeo 30 podem ser configurados para codificar em conjunto a informação relacionada com as diferenças do vector de

movimento de um bloco. A codificação conjunta pode-se referir a técnicas de codificação em que as informações relacionadas com os componentes x e y de um valor da diferença do vector de movimento estão intercaladas. A informação pode incluir informação indicativa se uma grandeza, por exemplo, um valor absoluto, dos componentes é maior do que zero, um sinal para os componentes quando o valor absoluto do componente correspondente é maior do que zero, e informação indicativa do valor absoluto do componente quando o valor absoluto é maior do que zero. A codificação conjunta dos um ou mais valores da diferença de vectores de movimento pode reduzir a complexidade de um fluxo de bits de codificação de vídeo, que pode proporcionar as vantagens aqui descritas. De modo semelhante ao codificador de vídeo 20, descodificador de vídeo 30 pode receber um fluxo de bits codificado de vídeo e pode operar de uma maneira essencialmente recíproca para o codificador de vídeo 20. Por exemplo, o descodificador de vídeo 30 pode receber valores codificados para uma MVD, descodificar os valores codificados em conjunto, e calcular um vector de movimento de um bloco como uma soma da MVD e o preditor do vector de movimento para o bloco.

Em geral, o componente x de um valor de diferença de um vector de movimento pode ser designado como "MVD\_x", enquanto que o componente y do valor de diferença de um vector de movimento pode ser designado como "MVD\_y". O valor de diferença do vector de movimento para um bloco de dados de vídeo, por exemplo, uma PU, pode ser descrito por

$\langle \text{MVD}_x, \text{MVD}_y \rangle$ . Em geral,  $\text{MVD}_x$  corresponde à diferença entre um componente  $x$  de um vector de movimento para o bloco e um componente  $x$  de um preditor de vector de movimento seleccionado para o bloco. Do mesmo modo,  $\text{MVD}_y$  corresponde à diferença entre um componente  $y$  de um vector de movimento para o bloco e um componente  $y$  de um preditor de vector de movimento seleccionado para o bloco. Deste modo, o codificador de vídeo 20 pode calcular  $\text{MVD}_x$  através do cálculo da diferença entre o componente  $x$  do vector de movimento para o bloco e o componente  $x$  do preditor do vector de movimento para o bloco, e  $\text{MVD}_y$  através do cálculo da diferença entre o componente  $y$  do vector de movimento para o bloco e o componente  $y$  do preditor do vector de movimento para o bloco. Da mesma forma, o decodificador de vídeo 30 pode reconstruir o vector de movimento para o bloco, adicionando  $\text{MVD}_x$  ao componente  $x$  do preditor do vector de movimento, e adicionando  $\text{MVD}_y$  ao componente  $y$  do preditor do vector de movimento. Por outras palavras, deixando  $\text{MV}_x$  e  $\text{MV}_y$  representar o componente  $x$  e o componente  $y$  do vector de movimento, respectivamente:

$$\text{MVD}_x = \text{MV}_x - \text{p}_x; \text{ and} \quad (1)$$

$$\text{MVD}_y = \text{MV}_y - \text{p}_y. \quad (2)$$

Em alguns exemplos, o codificador de vídeo 20 pode codificar, por exemplo, sinalizar, em conjunto, informação para o valor da diferença do vector de movimento para o bloco. Por exemplo, o codificador de vídeo 20 pode codificar informação indicativa de se um valor absoluto de

MVD\_x, isto é,  $|MVD_x|$ , é maior do que zero, seguido por informação indicativa de se um valor absoluto de MVD\_y, isto é,  $|MVD_y|$ , é maior do que zero. Tal informação pode incluir, por exemplo, as referências representativas de se os componentes x e y do valor MVD são maiores que zero, respectivamente. O codificador de vídeo 20 pode também sinalizar um valor representativo de um sinal para MVD\_x após a sinalização da referência que representa se MVD\_y é maior do que zero, assumindo que a referência que representa se MVD\_x é maior do que zero indica que MVD\_x é, na verdade, maior do que zero. O codificador de vídeo 20 não necessita de sinalizar um sinal para MVD\_x quando  $|MVD_x|$  é igual a zero.

A seguir ao valor indicativo do sinal para MVD\_x (quando sinalizado) ou a informação que indica se  $|MVD_y|$  é maior do que zero (quando o sinal de MVD\_x não está sinalizado), o codificador de vídeo 20 pode sinalizar um para MVD\_y, assumindo mais uma vez que  $|MVD_y|$  é maior do que zero. Além disso, o codificador de vídeo 20 pode então sinalizar informação representativa dos valores absolutos de MVD\_x e/ou MVD\_y, dependendo se uma ou ambas de MVD\_x e MVD\_y têm valores absolutos maiores que zero. Novamente, a informação representativa dos valores absolutos de MVD\_x e MVD\_y não necessita de ser sinalizada quando os valores anteriormente assinalados indicam que os valores absolutos de cada uma ou ambas de MVD\_x e MVD\_y são iguais a zero.

De um modo semelhante, o descodificador de vídeo

30 pode codificar (por exemplo, descodificar e interpretar) dados representativos de se  $MVD_x$  tem um valor absoluto maior que zero, seguido pelo representante de dados de se  $MVD_y$  tem um valor absoluto maior que zero. Quando os dados indicam que o valor absoluto de  $MVD_x$  é maior do que zero, o descodificador de vídeo 30 pode ser configurado para, em seguida, analisar a informação representativa de um sinal para  $MVD_x$ . Depois de analisar a informação representativa do sinal para  $MVD_x$ , ou após a determinação que  $|MVD_x|$  é igual a zero, o descodificador de vídeo 30 pode ser configurado para analisar a informação representativa de um sinal para  $MVD_y$ , quando os dados indicam que o valor absoluto de  $MVD_y$  é maior do que zero. De modo semelhante, o descodificador de vídeo 30 pode depois analisar a informação representativa de valores absolutos para  $MVD_x$  e  $MVD_y$ , respectivamente, assumindo novamente que os valores absolutos de  $MVD_x$  e  $MVD_y$  são maiores do que zero.

O descodificador de vídeo 30 pode ser configurado para não esperar informações sobre o sinal e o valor absoluto de qualquer componente para o qual os dados indicam que o componente tem um valor absoluto não maior do que zero (isto é, igual a zero). Isto é, o descodificador de vídeo 30 pode ser configurado para analisar outros dados de vídeo, por exemplo, outros dados para o bloco, sem receber ou codificar dados para os sinais e valores absolutos dos componentes MVD que têm valores absolutos iguais a zero, tal como indicado pela informação anteriormente sinalizada.

A codificação conjunta de componentes  $x$  e  $y$  desta forma, pode alcançar certas vantagens em relação a codificação de dados para componentes  $x$  e  $y$ , separadamente. Por exemplo, a codificação conjunta dos componentes  $x$  e  $y$  pode aumentar o débito durante a codificação de entropia. Como um exemplo particular, os valores da diferença de vectores de movimento da codificação conjunta podem aumentar o débito da codificação da entropia quando o codificador 20 ou descodificador 30 de vídeo utiliza CABAC para executar a codificação de entropia. Quando a informação representativa do valor absoluto dos componentes  $x$  e  $y$  de valores da diferença do vector de movimento é codificada em conjunto, o codificador de vídeo 20 ou descodificador 30 pode ser passível de entropia o código dos componentes  $x$  e  $y$  do valor da diferença do vector de movimento juntamente com a utilização do modo de desvio de CABAC. O modo de desvio CABAC pode melhorar o débito da codificação de entropia. A codificação de entropia da informação codificada em conjunto dos componentes  $x$  e  $y$  podem permitir que o codificador 20 ou descodificador 30 de vídeo realizem a codificação da entropia ambos os elementos de sintaxe em sucessão usando o modo de desvio de CABAC, o que pode melhorar o desempenho da codificação de entropia em relação à codificação de entropia separada dos componentes  $x$  e  $y$ . Embora descrito no exemplo acima em relação a CABAC, codificar em conjunto os componentes  $x$  e  $y$  dos valores da diferença do vector de movimento pode também aumentar a eficiência da codificação da entropia e/ou débito quando se utiliza outras técnicas de codificação de

entropia, tais como VLC, e CAVLC.

Após a codificação intra-preditiva ou inter-preditiva para produzir dados preditivos e dados residuais, e após todas as transformações (tais como as transformações inteiras 4x4 ou 8x8 utilizado em H.264/AVC ou uma transformação discreta de co-senos DCT) para produzir coeficientes de transformação, podem ser realizada a quantificação dos coeficientes de transformação. A quantificação refere-se geralmente a um processo em que os coeficientes de transformação são quantificados para possivelmente reduzir a quantidade de dados utilizados para representar os coeficientes. O processo de quantificação pode reduzir a profundidade do bit associada a alguns ou todos os coeficientes. Por exemplo, um valor de  $n$  bits pode ser arredondado para baixo para um valor de  $m$  bits durante a quantização, em que  $n$  é maior do que  $m$ .

Após a quantificação, pode ser executada a codificação de entropia dos dados quantificados, por exemplo de acordo com a codificação de comprimento variável adaptativo ao contexto (CAVLC), codificação aritmética binária adaptativa ao contexto (CABAC), ou outra metodologia de codificação de entropia. Uma unidade de processamento configurada para codificação de entropia, ou outra unidade de processamento, podem exercer outras funções de processamento, tais como codificação do comprimento executar zero de coeficientes quantificados e/ou geração de informação de sintaxe, tais como valores de

padrão de bloco codificado (CBP), tipo de macrobloco, modo de codificação, tamanho máximo de macrobloco para uma unidade codificada (tal como uma trama, fatia, macrobloco, ou sequência), ou semelhantes.

O codificador de vídeo 20 pode ainda enviar dados de sintaxe, tais como dados de sintaxe baseados em blocos, dados de sintaxe baseados em trama, e dados de sintaxe baseados em GOP, para o decodificador de vídeo 30, por exemplo, num cabeçalho de trama, um cabeçalho de bloco, um cabeçalho de fatia, ou um cabeçalho GOP. Os dados de sintaxe GOP podem descrever um número de tramas no respectivo GOP, sendo que os dados de sintaxe de trama podem indicar um modo de codificação/previsão utilizado para codificar a trama correspondente.

O codificador de vídeo 20 e decodificador de vídeo 26 podem ser implementados como qualquer um de uma variedade de circuitos codificadores ou decodificadores apropriados, se aplicável, tais como um ou mais microprocessadores, processadores de sinal digital (DSPs), circuitos integrados para aplicação específica (ASIC)s, redes de portas lógicas programáveis (FPGAs), circuito de lógica discreta, software, hardware, firmware ou qualquer combinação dos mesmos. Cada 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). Um dispositivo, incluindo o codificador

de vídeo 20 e/ou descodificador de vídeo 30 pode compreender um circuito integrado, um microprocessador, e/ou um dispositivo de comunicação sem fios, tal como um telemóvel.

A figura 2 é diagrama de blocos que ilustra um exemplo de um codificador de vídeo 20 que pode implementar técnicas para sinalizar em conjunto vectores de movimento, que podem ter precisão de sub-pixel. O codificador de vídeo 20 pode realizar a intra- e inter-codificação de blocos dentro de tramas de vídeo, incluindo Cus, ou sub-CUs de CUs. A intra-codificação baseia-se na previsão espacial para reduzir ou remover a redundância espacial em vídeo dentro de uma dada trama de vídeo. A inter-codificação baseia-se na previsão temporal para reduzir ou remover a redundância temporal no vídeo dentro de tramas contíguas de uma sequência de vídeo. O intra-modo (I-mode) pode-se referir a qualquer um dos vários modos de compressão baseados no espaço e inter-modos tais como a previsão unidireccional (modo P) ou previsão bidireccional (modo B) pode-se referir a qualquer um dos vários modos de compressão baseados no tempo. Embora os componentes para a codificação inter-modo estejam representados na figura 2, deverá ser entendido que o codificador de vídeo 20 pode ainda incluir componentes para codificação intra-modo. No entanto, tais componentes não estão ilustrados por razões de brevidade e clareza.

Tal como mostrado na figura 2, o codificador de

vídeo 20 recebe um bloco de vídeo actual dentro de uma trama de vídeo a ser codificada. No exemplo da figura 2, o codificador de vídeo 20 inclui a unidade de compensação de movimento 44, a unidade de estimativa de movimento 42, a memória de trama de referência 64, o somador 50, a unidade de processamento de transformação 52, a unidade de quantização 54, e unidade codificação de entropia 56. Para a reconstrução do bloco de vídeo, o codificador de vídeo 20 também inclui a unidade de quantização inversa 58, a unidade de transformação inversa 60, e somador 62. Um filtro de desbloqueio (não mostrado na figura 2) pode também ser incluído para filtrar limites de bloco para remover artefactos de bloqueio de vídeo reconstruído. Se desejado, o filtro de desbloqueio iria tipicamente filtrar a saída do somador 62.

Durante o processo de codificação, o codificador de vídeo 20 recebe uma trama de vídeo, fatia, ou CU a ser codificada. A trama ou fatia pode ser dividida em vários blocos de vídeo. A unidade de estimativa de movimento 42 e unidade de compensação de movimento 44 executam a codificação inter-preditiva do bloco de vídeo recebido relativamente a um ou mais blocos em uma ou mais tramas de referência para proporcionar a compressão temporal. A unidade de intra-previsão 46 pode executar a codificação inter-preditiva do bloco de vídeo recebido relativamente a um ou mais blocos vizinhos na mesma trama ou fatia como o bloco a ser codificado para proporcionar a compressão temporal. De acordo com as técnicas descritas abaixo, a

unidade de estimativa de movimento 42 pode determinar a precisão de um preditor de vector de movimento, o que pode ser determinada com base num modo intra ou inter-previsão que a unidade de selecção de modo 40 já determinou previamente. A unidade de estimativa de movimento 42, ou uma outra unidade de codificador de vídeo 20, tal como a unidade de codificação de entropia 56, pode ainda determinar uma diferença do vector de movimento que corresponde a uma diferença entre um preditor do vector de movimento e um vector de movimento calculado para uma PU da trama recebida, fatia, ou CU. A unidade de codificação de entropia 56 pode ainda codificar informação relacionada com a resolução, o sinal, e a amplitude de uma diferença de vector de movimento, bem como outras informações relacionadas com o vector de movimento de codificação conjunta, tal como descrito abaixo.

A unidade de selecção de modo 40 pode seleccionar um dos modos de codificação, intra ou inter, por exemplo, com base em resultados de erro, e proporciona o bloco intra ou inter-codificado resultante para o somador 50 para gerar os dados de bloco residuais e para o somador 62 para reconstruir o bloco codificado para utilizar como uma trama de referência. Além disso, a unidade de selecção de modo 40 pode seleccionar uma precisão de vector de movimento para um vector de movimento. Por exemplo, a unidade de selecção de modo 40 pode seleccionar uma precisão de vector de movimento para o vector de movimento com base na optimização de taxa de distorção (RDO) associada a um

quarto de subpixel e a um oitavo de subpixel de vectores de movimento de precisão. Os vectores de movimento com um quarto de pixel de precisão podem exigir menos bits para codificar em relação a vectores de movimento com uma precisão de um oitavo de pixel. No entanto, um bloco preditivo indicado com um vector de movimento codificado com uma precisão de um quarto de pixel pode resultar numa maior quantidade de informação residual em comparação com um bloco preditivo codificado com uma precisão de um oitavo de sub-pixel. O codificador de vídeo 20 pode executar RDO de modo a optimizar o número de bits (isto é, a taxa de bits) utilizado para codificar um vector de movimento particular contra a informação residual (distorção) associada a uma diferença entre o bloco preditivo indicado pelo vector de movimento, e o bloco que está actualmente a ser codificado. Com base no resultado do RDO, a unidade de selecção de modo 40 pode seleccionar a precisão do vector de movimento, por exemplo uma precisão de subpixel de um quarto ou um oitavo, que optimiza a troca da distorção de taxa.

A unidade de compensação de movimento 44 pode ser necessária para calcular os sub-pixeis, tais como sub-pixeis das tramas de referência, a várias precisões, por exemplo, a um oitavo e a um quarto de sub-pixeis. De modo a interpolar os sub-pixeis, a unidade de compensação de movimento 44 pode utilizar uma variedade de técnicas. Como exemplos, a unidade de compensação de movimento 44 pode utilizar a interpolação bilinear ou utilizar filtros de

resposta finitos (FIR) N-tap para interpolar um sub-pixel. Quando um dispositivo tal como a unidade de compensação de movimento 44 calcula um valor para um pixel fraccionário fazendo a média de dois píxeis ou sub-píxeis, pode arredondar, e/ou dimensionar o valor resultante. Em alguns casos, a unidade de compensação de movimento 44 pode calcular a média dos valores para dois sub-píxeis que são o resultado de uma média para um sub-inteiro de pixel

A unidade de compensação de movimento 44 pode calcular os valores para mais posições de pixel sub-inteiras, tais como posições de um oitavo de pixel, através da aplicação de filtros de interpolação a conjuntos de apoio. O suporte refere-se geralmente a valores para um ou mais píxeis de referência, por exemplo, píxeis numa linha ou região comum. Os píxeis podem corresponder a posições completas de píxeis ou posições de pixel sub-inteiras que foram previamente calculadas. Em alguns exemplos, a unidade de compensação de movimento 44 pode calcular valores para píxeis sub-inteiros utilizando a interpolação bilinear, e podem utilizar filtros de interpolação bilineares semelhantes para calcular os valores para duas ou mais diferentes posições de pixel sub-inteiras através da aplicação de um ou mais dos filtros de interpolação bilinear para diferentes conjuntos de suporte para as respectivas posições de pixel sub-inteiras.

Em alguns outros casos, a unidade de compensação de movimento 44 pode utilizar um filtro de resposta finita

(FIR) N-tap para interpolar valores sub-píxeis. Um FIR, tal como um filtro de Wiener 6 tomadas 12 tomadas, pode utilizar valores de pixel de apoio próximos para interpolar um valor de pixel sub-inteiro. Um pixel de apoio é um valor de pixel ou sub-pixel usado como entrada para o FIR. Um FIR pode ter uma ou mais dimensões. Num FIR unidimensional, um dispositivo tal como uma unidade de compensação de movimento 44 pode aplicar um filtro a um número píxeis ou sub-píxeis de apoio numa linha, por exemplo, na horizontal, vertical ou num ângulo. Em contraste a um FIR unidimensional, que pode usar píxeis de apoio numa linha recta, um FIR bidimensional, pode utilizar píxeis de apoio próximos ou sub-píxeis que formam um quadrado ou rectângulo para calcular o valor interpolado de píxeis.

A unidade de estimativa de movimento 42 e unidade de compensação de movimento 44 podem ser altamente integradas, mas são ilustradas separadamente para fins conceptuais. A estimativa do movimento é o processo de geração de vectores de movimento, que estimam o movimento para blocos de vídeo. Um vector de movimento, por exemplo, pode indicar que a deslocação de um bloco preditivo dentro de uma trama de referência preditiva (ou outra unidade codificada) em relação ao bloco actual a ser codificado dentro da trama actual (ou outra unidade de codificada). Um bloco preditivo é um bloco que é encontrado para corresponder de perto ao bloco a ser codificado, em termos de diferença de píxeis, que pode ser determinado pela soma da diferença absoluta (SAD), soma da diferença quadrado

(SSD), ou outras métricas de diferença. Um vector de movimento pode também indicar um deslocamento de uma partição de um macrobloco. A compensação do movimento pode envolver buscar ou gerar o bloco preditivo com base no vector de movimento determinado por estimativa de movimento. Novamente, a unidade de estimativa de movimento 42 e unidade de compensação de movimento 44 podem ser funcionalmente integradas, em alguns exemplos.

A unidade de estimativa de movimento 42 calcula um vector de movimento para o bloco de vídeo de uma trama inter-codificada através da comparação do bloco de vídeo com blocos de vídeo de uma trama de referência na memória de trama de referência 64. A unidade de compensação de movimento 44 pode também interpolar píxeis sub-inteiros da trama de referência, por exemplo, uma trama I ou uma trama P. Como exemplo, os vectores de movimento podem ser previstos a partir de duas listas de tramas de referência: a lista 0, que inclui tramas de referência que tenham uma ordem de exibição mais cedo do que uma trama actual a ser codificada, e a lista 1, que inclui tramas de referência que tenham uma ordem de exibição mais tarde do que a trama actual a ser codificada. Por isso, os dados armazenados na memória da trama de referência 64 podem ser organizados de acordo com estas listas.

A unidade de estimativa de movimento 42 compara os blocos de uma ou mais tramas de referência da memória de trama de referência 64 com um bloco a ser codificado de uma

trama actual, por exemplo, uma trama P ou uma trama B. Quando as tramas de referência na memória da trama de referência 64 incluem valores de píxeis de sub-inteiro, um vector de movimento calculado pela unidade de estimativa de movimento 42 pode referir-se a uma localização de pixel de sub-inteiro de uma das tramas de referência armazenadas na memória da trama de referência 64. A unidade de estimativa de movimento 42 e/ou a unidade de compensação de movimento 44 podem também ser configuradas para calcular os valores para as posições de pixel de sub-inteiro de tramas de referência armazenadas na memória da trama de referência 64, se não estiverem armazenados quaisquer valores para as posições de pixel de sub-inteiros na memória da trama de referência 64. A unidade de estimativa de movimento 42 envia o vector de movimento calculado para a unidade de codificação de entropia 56 e unidade de compensação de movimento 44. O bloco da trama de referência identificado por um vector de movimento pode ser designado como um bloco preditivo. O vector de movimento que indica o deslocamento do bloco preditivo em relação ao bloco actual a ser codificado na trama actual pode ter um componente x de  $mvd\_x$  e um componente y de  $mvd\_y$ .

A unidade de compensação de movimento 44 pode calcular os dados de previsão com base no bloco de previsão. O codificador de vídeo 20 forma um bloco de vídeo residual subtraindo os dados de previsão da unidade de compensação de movimento 44 do bloco de vídeo original que está a ser codificado. O somador 50 representa o componente

ou componentes que executam esta operação de subtração. A unidade de processamento de transformação 52 aplica uma transformação, tal como uma transformação discreta de cossenos (DCT) ou uma transformação conceptualmente semelhante, para o bloco residual, produzindo um bloco de vídeo que compreende valores de coeficientes de transformação residual. A unidade de processamento de transformação 52 pode executar outras transformações, tais como aquelas definidas pela norma H.264, que são conceptualmente semelhantes à DCT. Podem também ser utilizadas a transformação da onda, transformação do número inteiro, transformações de sub-banda ou outros tipos de transformações. Em qualquer dos casos, a unidade de processamento de transformação 52 aplica a transformação ao bloco residual, produzindo um bloco de coeficientes de transformação residuais. A transformação pode converter a informação residual a partir de um domínio de valores de pixel para um domínio de transformação, tal como um domínio de frequência. A unidade de quantificação 54 quantifica os coeficientes de transformação residual para reduzir ainda mais a taxa de bits. O processo de quantificação pode reduzir a profundidade do bit associada a alguns ou todos os coeficientes. O grau de quantificação pode ser modificado por ajuste de um parâmetro de quantificação.

A seguir à quantificação, a unidade de codificação de entropia 56 codifica em entropia coeficientes de transformação quantificados. Por exemplo, a unidade de codificação de entropia 56 pode executar a

codificação de comprimento variável adaptativo ao contexto (CAVLC), codificação aritmética binária adaptativa ao contexto (CABAC), ou outra técnica de codificação de entropia. A codificação conjunta dos componentes x e y dos valores da diferença do vector de movimento pode permitir à unidade de codificação de entropia 56 alcançar um débito de codificação de entropia mais elevado quando se utiliza CABAC. A unidade de codificação de entropia 56 pode usar o modo de desvio de CABAC para aumentar o débito da codificação de entropia dos componentes x e y codificados em conjunto dos valores da diferença do vector de movimento relativamente à codificação de entropia de componentes x e y codificados separadamente de valores da diferença do vector. 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 transmissão ou recuperação posterior. No caso da codificação aritmética binária adaptativa ao contexto, o contexto pode ser baseado em macroblocos vizinhos.

Em alguns casos, a unidade de codificação de entropia 56 ou outra unidade de codificação de vídeo 20 pode ser configurada para executar outras funções de codificação, para além da codificação de entropia. Por exemplo, a unidade de codificação de entropia 56 pode ser configurada para determinar os valores CBP para os macroblocos e partições. Além disso, em alguns casos, a unidade de codificação entropia 56 pode executar a codificação do comprimento dos coeficientes num macrobloco

ou partição do mesmo. Em particular, a unidade de codificação de entropia 56 pode aplicar um varrimento de zig-zague ou outro padrão de varrimento para varrer os coeficientes de transformação num macrobloco ou partição e codificar execuções de zeros para mais compressão. A unidade de codificação de entropia 56 pode também construir informação de cabeçalho com elementos de sintaxe apropriados para a transmissão no fluxo de bits codificado.

De acordo com as técnicas desta descrição, a unidade de quantificação 54 e unidade de codificação de entropia 56 podem codificar em conjunto os vectores de movimento, que podem ter apresentar uma precisão de subpixel, tal como descrito a seguir. Em alguns exemplos, a unidade de quantificação 45 e unidade de codificação de entropia 56 podem sinalizar um ou mais valores de referência que representam se os componentes x e/ou y de um valor da diferença do vector de movimento são iguais a zero, um ou mais valores que representam o sinal dos componentes x e/ou y quando os componentes têm valores absolutos não iguais a zero, e informação que representa o valor absoluto da magnitude dos componentes x e y do valor da diferença do vector de movimento quando os componentes têm valores absolutos não igual a zero. A unidade de quantificação 45 e unidade de codificação de entropia 56 podem também sinalizar uma referência de resolução de movimento, o que pode representar a resolução de subpixel do vector de movimento, bem como um valor de limiar relacionado com ambas a magnitude e a precisão do vector de

movimento.

A unidade de quantificação inversa 58 e a unidade de transformação inversa 60 aplicam a quantificação inversa e a transformação inversa, respectivamente, para reconstruir o bloco residual no domínio do pixel, por exemplo para a 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 uma das tramas da memória da trama de referência 64. A unidade de compensação de movimento 44 pode também aplicar um ou mais filtros de interpolação para o bloco residual reconstruído para calcular os valores dos píxeis sub-inteiros para utilização na estimativa de movimento. A soma 62 acrescenta o bloco residual reconstruído ao bloco de previsão compensado de movimento produzido pela unidade de compensação de movimento 44 para produzir um bloco de vídeo reconstruído para armazenamento em memória de trama de referência 64. O bloco de vídeo reconstruído pode ser utilizado pela unidade de estimativa de movimento 42 e unidade de compensação de movimento 44 como um bloco de referência para inter-codificar um bloco numa trama de vídeo seguinte.

O codificador de vídeo 20 representa um exemplo de um codificador de vídeo configurado para codificar os dados de vídeo representativos se um valor absoluto de um componente  $x$  de um valor da diferença do vector de movimento de um bloco corrente de dados de vídeo é maior do

que zero e para codificar o representante dos dados de vídeo se um valor absoluto de um componente y do valor da diferença do vector de movimento é maior do que zero. O codificador de vídeo 20 pode ser configurado para codificar representantes dos dados de vídeo do valor absoluto do componente x do valor da diferença do vector de movimento quando o valor absoluto do componente x do valor da diferença do vector de movimento é maior do que zero, e para codificar os dados de vídeo representativos do valor absoluto do componente y do valor da diferença do vector de movimento quando o valor absoluto do componente y do valor de diferença do vector de movimento é maior do que zero. O codificador de vídeo 20 pode também ser configurado para codificar representantes dos dados de vídeo de um sinal do componente x do valor da diferença do vector de movimento quando o valor absoluto do componente x do valor da diferença do vector de movimento é maior do que zero, e para codificar um sinal do componente y do valor da diferença do vector de movimento quando o valor absoluto do componente y do valor de diferença do vector de movimento é maior do que zero.

A figura 3 é um diagrama de blocos que ilustra pormenorizadamente um exemplo do descodificador de vídeo 30, que descodifica uma sequência de vídeo codificada. No exemplo da figura 3, o descodificador de vídeo 30 inclui uma unidade de descodificação de entropia 70, unidade de compensação de movimento 72, unidade de intraprevisão 74, unidade de quantificação inversa 76, unidade de

transformação inversa 78, memória de trama de referência 82 e somador 80. O descodificador de vídeo 30 pode, em alguns exemplos, executar uma passagem de descodificação no geral recíproca 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 previsão com base em vectores de movimento recebidos da unidade de descodificação de entropia 70.

A unidade de compensação de movimento 72 pode utilizar os vectores de movimento recebidos no fluxo de bits, incluindo os vectores de movimento sinalizados de acordo com as técnicas aqui descritas, para identificar um bloco de previsão numa das tramas de referência armazenadas na memória da trama de referência 82. A unidade de previsão 74 pode utilizar os modos intra-previsão recebidos no fluxo de bits para formar um bloco de previsão a partir de blocos espacialmente adjacentes. A unidade de quantificação inversa 76 quantifica inversamente, isto é, desquantifica, os coeficientes de bloco quantificados proporcionados no fluxo de bits e descodificados pela unidade de descodificação de entropia 70. O processo de quantificação inversa pode incluir um processo convencional, por exemplo tal como definido pela norma de descodificação H.264. O processo de quantificação inversa pode também incluir a utilização de um parâmetro de quantificação QPY calculado pelo codificador 50 para cada macrobloco para determinar um grau de quantificação e, do mesmo modo, um grau de quantificação inversa, que deve ser aplicado.

A unidade de descodificação de entropia 70 e a unidade de quantificação inversa 76 pode também reconstruir valores da diferença de vetores de movimento codificados em conjunto pela interpretação vários elementos de sintaxe incluindo referências e outros valores no fluxo de bits codificado. Como um exemplo, a unidade de descodificação de entropia 70 e unidade de quantificação inversa 76 pode receber um ou mais valores que representam se os componentes  $x$  e/ou  $y$  de uma diferença de vector de movimento são maiores do que zero, uma referência de resolução de movimento, o que pode indicar a resolução de subpixel do vector de movimento, e um valor de limiar, o qual pode indicar uma magnitude máxima e/ou resolução do valor de diferença do vector de movimento. Num exemplo, a unidade de descodificação de entropia 70 pode descodificar os componentes  $x$  e  $y$  codificados conjuntamente do valor da diferença do vector de movimento utilizando CABAC no modo de desvio. Usar o modo de desvio de CABAC pode permitir que a unidade de codificação de entropia 70 descodifique os valores da diferença do vector de movimento com débito cada vez maior, relativamente à descodificação separada de entropia dos componentes  $x$  e  $y$  dos valores da diferença do vector de movimento. A unidade de descodificação de entropia 70 e a unidade de quantificação inversa 76 podem também receber um ou mais valores que representam o sinal dos componentes  $x$  e/ou  $y$  do valor de diferença do vector de movimento, e valores que representam o valor absoluto da magnitude de cada componente do valor da diferença do vector de movimento. A unidade de descodificação de

entropia 70 e a unidade de quantificação inversa 76 podem usar os valores que representam se os componentes x e y do valor de diferença do vector de movimento são maiores do que zero, assim como os valores que representam o valor absoluto da magnitude de cada componente do valor da diferença do vector de movimento para reconstruir a amplitude, sinal, precisão, componentes x e y dos valores da diferença do vector de movimento.

A unidade de transformação inversa 58 aplica uma transformação inversa, por exemplo, uma DCT inversa, transformação inteira inversa, ou um processo de transformação inversa conceptualmente semelhante, para os coeficientes de transformação de modo a produzir blocos residuais no domínio do píxel. A unidade de compensação de movimento 72 produz blocos compensados no movimento, executando possivelmente a interpolação com base em filtros de interpolação. Os identificadores para filtros de interpolação a serem usados para a estimativa de movimento com precisão de sub-píxel podem ser integrados nos elementos de sintaxe. A unidade de compensação de movimento 72 pode usar filtros de interpolação tal como usado pelo codificador de vídeo 20 durante a codificação do bloco de vídeo para calcular os valores interpolados para píxeis sub-inteiros 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 informação de sintaxe

recebida e usa os filtros de interpolação para produzir blocos preditivos.

A unidade de compensação de movimento 72 utiliza alguma da informação de sintaxe para determinar tamanhos de macroblocos utilizados para codificar trama(s) da sequência de vídeo codificada, informação sobre a partição que descreve como cada macrobloco de uma trama da sequência de vídeo codificada é dividido, modos indicando como cada partição é codificada, uma ou mais tramas de referência (e listas de tramas de referência) para cada macrobloco ou partição inter-codificado, e outra informação para descodificar a sequência de vídeo codificada. Como um exemplo, a unidade de compensação de movimento 72 pode receber um valor MVD e um preditor do vector de movimento. A unidade de compensação de movimento 72 pode ainda determinar um vector calculado para um bloco actual como uma soma do valor MVD e o preditor do vector de movimento do bloco actual. O vector de movimento calculado para o bloco actual pode adicionalmente indicar a localização de um bloco preditivo. Em alguns exemplos, a unidade de compensação de movimento 72 pode calcular os valores dos píxeis para o bloco de previsão, por exemplo, quando o vector de movimento tem precisão de subpixel, tal como precisão de um quarto ou um oitavo de pixel.

O somador 80 soma os blocos residuais aos blocos de previsão correspondentes gerados pela unidade de compensação de movimento 72 ou unidade intra-previsão para

formar blocos descodificados. Se desejado, pode também ser aplicado um filtro de desbloqueio para filtrar os blocos descodificados a fim de remover as perturbações de bloqueio. Os blocos de vídeo descodificados são depois armazenados na memória da trama de referência 82, que proporciona blocos de referência para compensação de movimento subsequente e também produz vídeo descodificado para apresentação num dispositivo de visualização (tal como o dispositivo de visualização 32 da figura 1).

Com base no vector de movimento para o bloco actual, a unidade de compensação de movimento pode determinar um bloco preditivo a partir de uma ou mais tramas armazenadas na memória da trama de referência 82. A unidade de compensação de movimento 72 pode ser necessária para calcular sub-pixeis do bloco preditivo que a unidade de compensação de movimento 72 gera. Os blocos de previsão, tais como os sub-pixeis das tramas de referência, a várias precisões, por exemplo, a um oitavo e a um quarto de sub-pixeis. De modo a interpolar os sub-pixeis, a unidade de compensação de movimento 72 pode utilizar uma variedade de técnicas. Como exemplos, a unidade de compensação de movimento 72 pode utilizar a interpolação bilinear ou utilizar filtros de resposta finitos (FIRs) N-tap para interpolar um sub-pixel. Quando um dispositivo tal como a unidade de compensação de movimento 72 calcula um valor para um pixel fraccionário fazendo a média de dois pixeis ou sub-pixeis, pode arredondar, e/ou dimensionar o valor resultante. Em alguns casos, a unidade de compensação de

movimento 72 pode calcular a média dos valores para dois sub-pixels que são o resultado de uma média para um sub-inteiro de pixel

A unidade de compensação de movimento 72 pode calcular os valores para posições de pixel sub-inteiras, tais como as posições de um oitavo de pixel, através da aplicação de filtros de interpolação a conjuntos de apoio. O suporte refere-se geralmente a valores para um ou mais pixels de referência, por exemplo, pixels numa linha ou região comum. Os píxeis podem corresponder a posições completas de pixels ou posições de pixel sub-inteiras que foram previamente calculadas. Em alguns exemplos, a unidade de compensação de movimento 72 pode calcular valores para pixels sub-inteiros utilizando a interpolação bilinear, e podem utilizar filtros de interpolação bilineares semelhantes para calcular os valores para duas ou mais diferentes posições de pixel sub-inteiro através da aplicação de um ou mais dos filtros de interpolação bilinear para diferentes conjuntos de suporte para as respectivas posições de pixel sub-inteiras.

Em alguns outros casos, a unidade de compensação de movimento 42 pode utilizar um filtro de resposta finita (FIR) N-tap para interpolar valores sub-píxeis. Um FIR, tal como um filtro de Wiener 6 tomadas 12 tomadas, pode utilizar valores de pixel de apoio próximos para interpolar um valor de pixel sub-inteiro. Um pixel de apoio é um valor de pixel ou sub-pixel usado como entrada para o FIR. Um FIR

pode ter uma ou mais dimensões. Num FIR unidimensional, um dispositivo tal como uma unidade de compensação de movimento 72 pode aplicar um filtro a um número pixels ou sub-pixels de apoio numa linha, por exemplo, na horizontal, vertical ou num ângulo. Em contraste a um FIR unidimensional, que pode usar pixels de apoio numa linha recta, um FIR bidimensional, pode utilizar pixels de apoio próximos ou sub-pixels que formam um quadrado ou rectângulo para calcular o valor interpolado de pixels.

O descodificador de vídeo 30 representa um exemplo de um codificador de vídeo configurado para descodificar os dados de vídeo representativos se um valor absoluto de um componente  $x$  de um valor da diferença do vector de movimento de um bloco corrente de dados de vídeo é maior do que zero e para descodificar o representante dos dados de vídeo se um valor absoluto de um componente  $y$  do valor da diferença do vector de movimento é maior do que zero. O descodificador de vídeo 30 pode ser configurado para descodificar representantes dos dados de vídeo do valor absoluto do componente  $x$  do valor da diferença do vector de movimento quando o valor absoluto do componente  $x$  do valor da diferença do vector de movimento é maior do que zero, e para descodificar os dados de vídeo representativos do valor absoluto do componente  $y$  do valor da diferença do vector de movimento quando o valor absoluto do componente  $y$  do valor de diferença do vector de movimento é maior do que zero. O descodificador de vídeo 30 pode também ser configurado para descodificar representantes dos dados de

vídeo de um sinal do componente x do valor da diferença do vector de movimento quando o valor absoluto do componente x do valor da diferença do vector de movimento é maior do que zero, e para descodificar um sinal do componente y do valor da diferença do vector de movimento quando o valor absoluto do componente y do valor de diferença do vector de movimento é maior do que zero.

A figura 4 é um fluxograma que ilustra um exemplo do método para a codificação conjunta dos componentes x e y de um valor da diferença do vector de movimento e codificação de entropia de um bloco actual. No método 150, a unidade de estimativa de movimento 42 do codificador de vídeo 20 pode prever um bloco actual (152). Como parte da previsão do bloco actual, a unidade de estimativa de movimento 42 pode calcular um vector de movimento. O vector de movimento pode indicar um deslocamento de um bloco preditivo em relação ao bloco actual. A deslocação do vector de movimento pode ter componentes x e y,  $MV_x$ , e  $MV_y$ . O bloco preditivo pode corresponder de perto ao bloco actual em termos de diferença de píxeis, que pode ser determinada pela soma da diferença absoluta (SAD), soma da diferença quadrado (SSD), ou outras métricas de diferença. No caso em que a trama actual é inter-prevista, o bloco preditivo pode compreender um bloco de uma outra trama, tal como aquela de uma ou mais tramas de referência na memória da trama de referência 64 da figura 2.

A unidade de compensação de movimento 44, ou uma

outra unidade de codificador de vídeo 20 pode então determinar um preditor de vector de movimento para o vector de movimento do bloco actual (154). Um preditor de vector de movimento é um vector de movimento que a unidade de compensação de movimento 44 utiliza para calcular os valores da diferença do vector de movimento para o bloco actual. Em alguns exemplos, o preditor do vector de movimento pode ser o vector de movimento de um de uma pluralidade de blocos que se encontram na vizinhança da PU do bloco actual. A unidade de compensação de movimento 44 pode também determinar o preditor do vector de movimento de um bloco co-localizado de uma das tramas de referência armazenadas na memória da trama de referência 64. O preditor do vector de movimento pode ter um componente x de  $p_x$  e um componente y de  $p_y$ . O codificador de vídeo 20 pode determinar o preditor do vector de movimento de tal modo que o preditor do vector de movimento minimiza a diferença do vector de movimento do bloco actual e o preditor do vector de movimento.

Baseado no preditor do vector de movimento, unidade de compensação de movimento 44 do codificador de vídeo 20, ou outra unidade do codificador de vídeo 20, tal como a unidade de codificação de entropia 56, pode calcular um valor da diferença do vector de movimento para o bloco actual (156). A unidade de compensação de movimento 44 pode calcular a diferença do vector de movimento para o bloco actual como uma diferença do vector de movimento para o bloco actual e o preditor do vector de movimento para o

bloco actual. Deste modo, a unidade de compensação de movimento 44 pode calcular o preditor do vector de movimento para o bloco actual como  $\langle p_x - MV_x, p_y - MV_y \rangle$ , em que  $p_x$  e  $p_y$  são os componentes x e y do preditor do vector de movimento, e  $MV_x$  e  $MV_y$  são os componentes x e y do vector de movimento para o bloco actual.

A unidade de codificação de entropia 56 pode então codificar em conjunto os elementos do valor da diferença do vector de movimento do bloco actual (158). Por exemplo, a unidade de codificação de entropia 56 pode codificar em conjunto elementos do valor da diferença do vector de movimento, que inclui os componentes x e y da diferença do vector de movimento tal como explicado em relação às figuras 6 e 7 pormenorizadamente a seguir. Os elementos da diferença do vector de movimento podem também incluir um ou mais valores representativos de se ou não os componentes x e y da diferença do vector de movimento são igual a zero, um valor representativo do valor do sinal dos componentes x e/ou y do valor de diferença do vector de movimento, e um valor representativo do valor absoluto do componente do valor da diferença do vector de movimento, como alguns exemplos não limitativos.

A unidade de processamento de transformação 52 ou unidade de compensação de movimento 44 pode calcular um bloco residual para o bloco actual (160). O bloco residual pode compreender as diferenças de píxeis entre o bloco de

vídeo preditivo, o qual é indicado pelo vector de movimento para o bloco actual, e o bloco de vídeo actual. A unidade de processamento de transformação 52 pode depois aplicar uma transformação, tal como uma transformação discreta de co-senos (DCT), ao bloco residual para o bloco actual, enquanto que a unidade de quantificação 54 pode quantificar os coeficientes de transformação da unidade de processamento de transformação 52 (162).

A unidade de codificação de entropia 56 pode digitalizar os coeficientes transformados resultantes da transformação de 162. Em alguns casos a unidade de codificação de entropia 56 pode aplicar um varrimento em ziguezague ou outro padrão de varrimento para varrer os coeficientes de transformação do bloco residual transformado (164). A unidade de codificação de entropia 56 pode codificar por entropia os coeficientes que compreendem execuções de zeros que resultam do varrimento de 164, por exemplo usando a codificação de execução de comprimento, para compressão adicional (166). A unidade de codificação de entropia 56 pode então emitir os coeficientes codificados de entropia e o valor MVD codificado em (168). Isto é, a unidade de codificação de entropia 56 pode emitir dados codificados em conjunto para x e y do valor MVD, adicionalmente aos coeficientes de transformação quantificados codificados em entropia.

As técnicas da figura 4 podem, geralmente, ser realizadas por qualquer unidade de processamento ou

processador, sejam implementadas em hardware, software, firmware, ou uma combinação das mesmas, e quando implementadas em software ou firmware, pode ser proporcionado o hardware correspondente para executar instruções para o software ou firmware. Com objectivos de exemplo, as técnicas da figura 4 estão descritas em relação a um dispositivo de codificação de vídeo, que pode incluir componentes substancialmente semelhantes aos do codificador de vídeo 20 (figuras 1 e 2), embora deva ser compreendido que outros dispositivos podem ser configurados para executar técnicas semelhantes. Além disso, os passos ilustrados na figura 4 podem ser realizados numa ordem diferente ou em paralelo, podendo ser adicionados passos adicionais e certos passos omitidos, sem fugir das técnicas desta descrição.

A figura 5 é um fluxograma que ilustra um método de sinalização separada do vector de movimento de um valor da diferença do vector de movimento de acordo com as técnicas descritas no modelo de teste HEVC 3.0 (HM3.0, também designado como HM3). De acordo com o método proposto em HM3, e ilustrado na figura 5, os valores  $mvd_x$  e  $mvd_y$ , que correspondem aos componentes  $x$  e  $y$  de um valor de diferença do vector de movimento, são sinalizados separadamente. O método 180 ilustra como um ou mais elementos de sintaxe referentes a  $mvd_x$  podem ser sinalizados, e o método 192 ilustra como um ou mais elementos de sintaxe referentes a  $mvd_y$  podem ser sinalizados.

No método 180, uma unidade do codificador de vídeo 20, tal como a unidade de compensação de movimento 44 ou unidade de codificação de entropia 56, pode sinalizar uma referência  $mvd_x == 0$  (182). A referência  $mvd_x == 0$  representa se a diferença entre os componentes x de um vector de movimento calculado para uma PU de uma CU e um preditor de vector de movimento para o vector de movimento são iguais e, por conseguinte, se a diferença correspondente entre os dois componentes x é igual a zero.

Se a referência  $mvd_x == 0$  estiver definida, então  $mvd_x$ , o componente x de MVD, é igual a zero (derivação "Sim" de 184), e o método termina (190). No entanto, se  $mvd_x$  for diferente de zero (derivação "Não" de 184), o codificador de vídeo 20 pode sinalizar  $abs\_mvd\_x1$  (186).  $Abs\_mvd\_x-1$  pode representar o valor absoluto do valor  $mvd_x$  menos um. O codificador de vídeo 20 pode adicionalmente sinalizar um valor que representa o sinal de  $mvd_x$  (188). O sinal pode ser, por exemplo, positivo ou negativo.

Da mesma forma, no método 192, um codificador de vídeo, tal como o codificador de vídeo 20 da figura 1, pode sinalizar uma referência  $mvd_y == 0$  (194). A referência  $mvd_y == 0$  representa se a diferença entre os componentes y de um vector de movimento calculado para uma PU de uma CU e um preditor de vector de movimento para o vector de movimento são iguais e, por conseguinte, se a diferença correspondente entre os dois componentes y é igual a zero.

Se a referência  $mvd\_y=0$  estiver definida, então  $mvd\_y$ , o componente  $y$  de MVD, é igual a zero (derivação "Sim" de 196), e o método termina (202). No entanto, se  $mvd\_y$  for diferente de zero (derivação "Não" de 196), o codificador de vídeo 20 pode sinalizar  $abs\_mvd\_y1$  (198).  $Abs\_mvd\_y-1$  pode ser um valor que representa o valor absoluto do valor  $mvd\_y$  menos um. O codificador de vídeo 20 pode adicionalmente sinalizar um valor que representa o sinal de  $mvd\_y$  (200). O sinal pode ser, por exemplo, positivo ou negativo.

A figura 6 é um fluxograma que ilustra um exemplo do método para a codificação conjunta dos componentes  $x$  e  $y$  de um valor da diferença do vector de movimento. Em geral, o método 220 da figura 6 corresponde ao passo 158 da figura 4. Assim, o método 220 da figura 6 representa um exemplo de método para a codificação conjunta de dados de componentes  $x$  e  $y$  de um valor da diferença do vector de movimento. Além disso, presume-se que antes de desempenho do exemplo do método da figura 6, unidade de compensação de movimento 44 do codificador de vídeo 20, ou outra unidade do codificador de vídeo 20, tal como uma unidade de codificação de entropia 56 do codificador de vídeo 20 pode calcular um vector de movimento para o bloco actual de dados de vídeo e calcular o valor da diferença do vector de movimento como uma diferença entre o vector de movimento do bloco actual de dados de vídeo e um preditor de vector de movimento determinado para o bloco actual de dados de vídeo, por exemplo tal como descrito em relação ao passo 156 da figura

4. Para codificar dados em conjunto para os componentes x e y do valor da diferença do vector de movimento, a unidade de codificação de entropia 56 pode codificar informação representativa se o valor absoluto de  $mvd\_x$ , o componente x do valor da diferença do vector de movimento, é maior do que zero (222). A unidade de codificação de entropia 56 pode então codificar informação representativa de se um valor absoluto de um componente y ( $mvd\_y$ ) do valor da diferença do vector de movimento é maior do que zero (224).

Quando o valor absoluto do componente x do valor da diferença do vector de movimento é maior do que zero, a unidade de codificação de entropia 56 pode codificar informação representativa do valor absoluto do componente x do valor da diferença do vector de movimento (226). E quando o valor absoluto do componente y do valor da diferença do vector de movimento é maior do que zero, a unidade de codificação de entropia 56 pode codificar informação representativa do valor absoluto do componente y do valor da diferença do vector de movimento (228). Em alguns exemplos a unidade de codificação de entropia 56 pode codificar a informação representativa do valor absoluto dos componentes x e y dos valores da diferença do vector como os valores absolutos de  $mvd\_x/2 - 1$ , e  $mvd\_y/2 - 1$ , respectivamente.

Quando o valor absoluto do componente x do valor da diferença do vector de movimento é maior do que zero, a unidade de codificação de entropia 56 pode codificar um

sinal do componente x do valor da diferença do vector de movimento (230). Quando o valor absoluto do componente y do valor da diferença do vector de movimento é maior do que zero, a unidade de codificação de entropia 56 pode codificar um sinal do componente y do valor da diferença do vector de movimento (232).

As técnicas da figura 6 podem, geralmente, ser realizadas por qualquer unidade de processamento ou processador, sejam implementadas em hardware, software, firmware, ou uma combinação das mesmas, e quando implementadas em software ou firmware, pode ser proporcionado o hardware correspondente para executar instruções para o software ou firmware. Com objectivos de exemplo, as técnicas da figura 6 estão descritas em relação a um dispositivo de codificação de vídeo, que pode incluir componentes substancialmente semelhantes aos do codificador de vídeo 20 (figuras 1 e 2), embora deva ser compreendido que outros dispositivos podem ser configurados para executar técnicas semelhantes. Além disso, os passos ilustrados na figura 6 podem ser realizados numa ordem diferente ou em paralelo, podendo ser adicionados passos adicionais e certos passos omitidos, sem fugir das técnicas desta descrição.

Deste modo, o método da figura 6 representa um exemplo de um método de que inclui codificar informação representativa de se um valor absoluto de um componente x de um valor da diferença do vector de movimento para um

bloco actual de dados de vídeo é maior do que zero, codificar informação representativa de se um valor absoluto de um componente  $y$  do valor da diferença de vector de movimento é maior do que zero, quando o valor absoluto do componente  $x$  do valor da diferença do vector de movimento é maior do que zero, codificar informação representativa do valor absoluto do componente  $x$  do valor da diferença do vector de movimento, quando o valor absoluto do componente  $y$  do valor de diferença do vector de movimento é maior do que zero, codificar a informação representativa do valor absoluto do componente  $y$  do valor da diferença de vector de movimento, quando o valor absoluto do componente  $x$  do valor de diferença do vector de movimento é maior do que zero, codificar um sinal do componente  $x$  do valor de diferença do vector de movimento, e quando o valor absoluto do componente  $y$  do valor da diferença do vector de movimento é maior do que zero, codificar um sinal do componente  $y$  do valor de diferença do vector de movimento. Deste modo, o método da figura 6 representa um exemplo de elementos de codificação conjunta de componentes  $x$  e  $y$  de um valor da diferença do vector de movimento calculado para um vector de movimento de um bloco actual de dados de vídeo.

A figura 7 é um fluxograma que ilustra um método para sinalizar valores para valores  $mvd_x$  e  $mvd_y$ . Ao contrário dos métodos 180 e 192 da figura 5, que ilustram a codificação dos componentes  $x$  e  $y$  de um MVD separadamente, e semelhante ao método da figura 6, o método da figura 7 também pode permitir que um codificador codifique

conjuntamente os componentes x e y de um valor da diferença do vector de movimento, isto é, `mvd_x` e `mvd_y`. Neste exemplo, um elemento de sintaxe sinalizado no fluxo de bits por um codificador, tal como o codificador de vídeo 20 das figuras 1 e 2, podem com magnitude, sinal e/ou outra informação relativa aos componentes x ou y de um valor da diferença do vector de movimento.

No exemplo da figura 7, tal como explicado a seguir, o codificador de vídeo 20 pode codificar uma referência de resolução de movimento, que é um valor representativo de se um vector de movimento do bloco actual de dados de video tem uma primeira precisão de sub-pixeis, por exemplo, precisão de quarto de pixel, ou uma segunda precisão de sub-pixel, por exemplo precisão de oitavo de pixel. Neste exemplo, a referência de resolução de movimento tem um valor de um para indicar que o vector de movimento do bloco actual tem precisão de quarto de pixel, e tem um valor de zero para indicar que a referência da resolução do movimento tem uma precisão de oitavo de pixel. O codificador de vídeo 20 pode também codificar um valor, `"abs_mvd_equal_2"`, que representa se a magnitude de um ou ambos os componentes da diferença do vector de movimento tem um valor absoluto de dois. O codificador de vídeo 20 pode atribuir um valor a `abs_mvd_equal_2` com base no exemplo da tabela 1 a seguir.

No exemplo de método 240 da figura 7, a unidade de compensação de movimento 44 ou outra unidade do

codificador de vídeo 20 pode sinalizar um  $mvd_x=0$  referência, (242) e um  $mvd_y=0$  referência (244). Estas referências podem ser definidas quando os valores  $mvd_x$  ou  $mvd_y$  correspondentes são iguais a zero. Se ambos  $mvd_y$  e  $mvd_x$  forem zero ("Sim" derivação 246), não precisam de ser sinalizados mais dados, e o método 240 termina (266).

Se tanto  $mvd_x$  ou  $mvd_y$  são diferentes de zero (derivação "Não" de 246), a unidade de compensação de movimento 44 pode sinalizar ou codificar uma referência de resolução de movimento (246). A referência de resolução de movimento ( $mvres\_flag$ ) é um exemplo de um elemento de sintaxe que pode indicar uma resolução de vector de movimento (também designado como precisão de vector de movimento) para um vector de movimento que corresponde aos valores da diferença de vectores de movimento. A referência de resolução de movimento pode indicar se o vector de movimento do bloco actual de dados de vídeo tem uma primeira precisão de sub-pixeis, ou uma segunda precisão de sub-pixel, por exemplo, precisão de um quarto de pixel ou precisão de um oitavo de pixel.

O codificador de vídeo 20 pode também determinar um limiar para uma trama actual, que também está associada ao vector de movimento para o bloco actual, e que pode ser sinalizado no cabeçalho da fatia ou outra estrutura de dados, por exemplo, um conjunto de parâmetros de fatia (SPS) ou conjunto de parâmetros de imagem (PPS). O codificador de vídeo 20 pode utilizar o limiar para

restringir a amplitude dos valores de diferença do vector de movimento de oitavo de pixel e pode indicar que o vector de movimento tem a primeira precisão de subpixel quando tanto o componente x do valor de diferença do vector de movimento (mvd\_x) como o componente y do valor de diferença do vector de movimento (mvd\_y) são maiores do que o limiar. Por exemplo, tal como mostrado a seguir no que diz respeito à tabela 1 abaixo, o codificador de vídeo (por exemplo, o codificador de vídeo 20) pode restringir a magnitude dos componentes x e y de um valor da diferença de um vector de movimento de precisão de oitavo de pixel para valores de um ou dois. O codificador de vídeo pode sinalizar o valor do limiar usando uma referência de limiar.

Se a MVD calculada para um componente x ou y de um vector de movimento para um PU for maior do que o limiar determinado, a referência da resolução do movimento pode ser definida como verdadeira. Se a referência da resolução do movimento estiver definida como verdadeira, a referência que representa que o vector de movimento tem uma precisão de um quarto de pixel. Se o valor MVD for menor ou igual ao limiar seleccionado, o codificador de vídeo pode definir um valor para a referência de resolução de movimento para indicar se um vector de movimento correspondente tem um quarto ou um oitavo de precisão de pixel.

O método 240 pode adicionalmente compreender o codificador de vídeo 20 que determina se a referência da resolução do movimento é igual a um (250). A referência de

resolução movimento pode ser igual a um, por exemplo se a precisão de vector de movimento tiver uma precisão de um quarto. Se a referência da resolução do movimento estiver definida para um, o codificador de vídeo 20 pode sinalizar ou codificar um valor representativo do valor  $\text{abs\_mvd\_x}/2 - 1$  se  $\text{mvd\_x}$  for diferente de zero, onde  $\text{abs\_mvd\_x}$  é o valor absoluto do componente x da MVD (252). O codificador de vídeo 20 pode de modo semelhante codificar um valor representativo de  $\text{abs\_mvd\_y}/2 - 1$  se  $\text{mvd\_x}$  for diferente de zero, onde  $\text{abs\_mvd\_y}$  é o valor absoluto do componente y da MVD (254). Em alguns exemplos, o codificador de vídeo 20 pode sinalizar  $\text{abs\_mvd\_y}/2 - 1$  se  $\text{mvd\_y}$  for diferente de zero, em vez de se  $\text{mvd\_x}$  for diferente de zero.

Se a referência da resolução do movimento não for igual a um (derivação "Não" de 250), o codificador de vídeo 20 pode determinar se o limiar é igual a um. O limiar pode ter um valor de um ou dois neste exemplo. Num exemplo, o limiar pode ter um valor de dois. Dado que o vector de movimento pode ter apenas um oitavo de pixel de precisão quando os valores  $\text{mvd\_x}$  e  $\text{mvd\_y}$  não excedem o limiar, os valores de  $\text{mvd\_x}$  e  $\text{mvd\_y}$  estão restritos a um ou dois neste exemplo. Além disso, se o limiar for igual a um (derivação "Sim" de 256), e o vector de movimento tiver uma precisão quarto de pixel (tal como indicado pela referência de resolução do movimento, que neste exemplo tem de ser verdadeiro a seguir ao passo 250), o codificador de vídeo não precisa de sinalizar qualquer informação adicional para os valores  $\text{mvd}$ . Isto é, os valores de  $\text{mvd\_x}$  e  $\text{mvd\_y}$  podem

ser determinados como sendo zero ou um usando outra informação. Em particular, quer `mvd_x` e `mvd_y` sejam zero ou um, pode ser indicado com base em se os valores dos sinais são sinalizados nos passos 248 e 264. Se o limiar for igual a dois (derivação "Não" de 256), o codificador de vídeo 20 pode determinar se quer os valores `mvd_x` ou `mvd_y` são iguais a zero. Se `mvd_x` ou `mvd_y` forem igual a zero (derivação "Sim" de 260), o codificador de vídeo 20 pode avançar para a sinalização dos sinais de diferente de zero uma de `mvd_x` e `mvd_y` (258, 264). Em alguns exemplos, codificador de vídeo 20 pode adicionalmente sinalizar um valor para o diferente de zero uma das `mvd_x` e `mvd_y`, e assim, em vez disso pode avançar para o passo 252, em vez de 258.

No entanto, se ambos `mvd_x` e `mvd_y` forem diferentes de zero, isto é, nem `mvd_x` nem `mvd_y` têm um valor de zero (derivação "Não" de 260), então o codificador de vídeo 20 pode sinalizar uma referência `abs_mvd_equal_2`. A referência `abs_mvd_equal_2` é um exemplo de um valor codificado em conjunto para `mvd_x` e `mvd_y`. Um codificador de vídeo pode atribuir um valor a `abs_mvd_equal_2` com base na tabela 1 a seguir.

Tabela 1

<code>abs_mvd_equal_2</code>	<code>abs_mvd_x</code>	<code>abs_mvd_y</code>
0	1	1
1	1	2
2	2	1

Com base nos valores de `abs_mvd_x` e `abs_mvd_y`, o codificador de vídeo 20 pode sinalizar o valor de `abs_mvd_equal_2` (262), por exemplo, de acordo com a tabela 1. O codificador de vídeo 20 pode sinalizar o valor de `abs_mvd_equal_2` usando binarização unária. O codificador de vídeo 20 pode adicionalmente sinalizar o sinal de `mvd_x` se `mvd_x` for diferente de zero (258) e de modo semelhante sinalizar o sinal de `mvd_y` se `mvd_x` for diferente de zero (264). Em alguns exemplos, o codificador de vídeo 20 pode sinalizar o sinal de `mvd_y` se `mvd_y` for diferente de zero.

Em alguns exemplos, a referência de resolução de movimento, referência `mvd_x==0`, referência `mvd_y==0` são codificadas em conjunto. Opcionalmente, e como um exemplo, códigos VLC podem ser utilizados de acordo com a tabela 2 a seguir:

Tabela 2

<code>mvd_x=0</code>	<code>mvd_y==0</code>	Referência de resolução de movimento	Chave VLC
verdadeiro	verdadeiro	X	00
verdadeiro	falso	1	101
verdadeiro	falso	0	1110
falso	verdadeiro	1	100
falso	verdadeiro	0	110
falso	falso	0	1111
falso	falso	1	01

O codificador de vídeo 20 pode sinalizar a referência de resolução de movimento para indicar uma resolução de vector de movimento de um quarto, e um oitavo de pixel. Por exemplo, o codificador de vídeo 20 pode sinalizar a resolução da diferença do vector de movimento entre  $1/4$  de pixel ou  $1/8$  de pixel. O codificador de vídeo 20 pode sinalizar a referência de resolução de movimento para indicar uma diferença de precisão do vector de movimento entre um quarto e um oitavo de pixel quando a MVD é menor do que um valor limiar. O codificador de vídeo 20 pode sinalizar a referência de resolução de movimento para indicar uma diferença de precisão do vector de movimento entre um quarto e um oitavo de pixel quando um ou ambos os componentes da MVD é maior do que um valor limiar.

Um dispositivo descodificador de vídeo, tal como um descodificador de vídeo 30 pode operar de uma maneira essencialmente recíproca para o codificador de vídeo 20 para executar as técnicas ilustradas na figura 7. Deste modo, o descodificador de vídeo 30 pode adaptativamente seleccionar a precisão do vector de movimento para cada vector de movimento, baseado na recepção e descodificação de um valor que representa o vector de movimento, tal como a referência de resolução de movimento. O descodificador de vídeo 30 pode também receber e descodificar o valor limiar de (256). O descodificador de vídeo 30 pode receber e descodificar o valor limiar para cada trama de referência para restringir a amplitude dos valores da diferença do vector de movimento de um oitavo de subpixel para a trama.

O limiar pode ser recebido no cabeçalho da fatia. O limiar pode ter valores de zero, um ou dois. O limiar não é restrito, e pode ser alargado a outros valores. Ao codificar a informação sobre os componentes x e y de valores da diferença de vectores de movimento, pode ser reduzida a complexidade de um fluxo de bits de vídeo codificado, o que pode resultar em uma taxa de bits total menor para dados de vídeo codificados.

Num exemplo, um dispositivo de codificação de vídeo, tal como o codificador de vídeo 20 ou descodificador de vídeo 30, pode codificar um valor, tal como a referência de resolução de movimento da figura 7, que é representativa de se um vector de movimento do bloco actual de dados de vídeo tem uma primeira precisão de sub-pixeis ou uma segunda precisão de sub-pixeis, tal como uma precisão de um quarto de pixel ou um oitavo de pixel. A segunda precisão de sub-pixeis pode ser maior do que a primeira precisão de sub-pixeis. Quando o vector de movimento tem a segunda precisão de sub-pixeis, por exemplo, precisão de um oitavo do pixel, quando o componente x do valor da diferença do vector de movimento não é igual a zero, e quando o componente y do valor da diferença do vector de movimento é não igual a zero, o codificador de vídeo 20 ou o descodificador de vídeo 30 podem codificar informação representativa do valor absoluto do componente x do valor de diferença do vector de movimento e informação codificada representativa do valor absoluto do componente y do valor da diferença do vector de movimento. O codificador de vídeo

20 ou descodificador de vídeo 30 podem codificar um valor codificado em conjunto que representa ambos o componente x do valor de diferença do vector de movimento e o componente y do valor de diferença do vector de movimento.

As técnicas da figura 7 podem, geralmente, ser realizadas por qualquer unidade de processamento ou processador, sejam implementadas em hardware, software, firmware, ou uma combinação das mesmas, e quando implementadas em software ou firmware, pode ser proporcionado o hardware correspondente para executar instruções para o software ou firmware. Com objectivos de exemplo, as técnicas da figura 7 estão descritas em relação a um dispositivo de codificação de vídeo, que pode incluir componentes substancialmente semelhantes aos do codificador de vídeo 20 (figuras 1 e 2). O descodificador de vídeo 30 das figuras 1 e 3 podem executar operações geralmente recíprocas daqueles ilustrados e descritos na figura 7 para descodificar um fluxo de bits codificados de acordo com as técnicas ilustradas na figura 7. Deverá também ser entendido que outros dispositivos podem ser configurados para executar técnicas semelhantes. Além disso, os passos ilustrados na figura 7 podem ser realizados numa ordem diferente ou em paralelo, podendo ser adicionados passos adicionais e certos passos omitidos, sem fugir das técnicas desta descrição.

O método da figura 7 representa um exemplo de um método que inclui a determinação se se deve utilizar um

vector de movimento tendo uma precisão de primeiro sub-pixel ou precisão de um segundo sub-pixel para codificar uma unidade de previsão de uma unidade de codificação de dados de vídeo, em que a segunda precisão de sub-pixel é maior do que a primeira precisão de sub-pixel, calcular um componente x e um componente y do vector de movimento que apresenta quer a primeira precisão de sub-pixel ou a segunda precisão de sub-pixel com base na determinação, quando o componente x e o componente y têm a segunda precisão de sub-pixeis, determinar os valores da diferença dos vectores de movimento com base nos valores do componente x e componente y do vector de movimento e valores de um componente x previsto e um componente y previsto para o vector de movimento, e codificar um valor codificado em conjunto representativo dos valores da diferença dos vectores de movimento para o componente x e o componente y para o vector de movimento.

Um método recíproco pode ser realizado por um descodificador de vídeo. O método recíproco pode, por exemplo, incluir determinar se um vector de movimento de uma unidade de previsão de uma unidade de codificação de dados de vídeo tem uma primeira precisão de sub-pixel ou uma segunda precisão de sub-pixel usando um valor de uma referência de resolução de movimento para o vector de movimento, em que a segunda precisão de sub-pixel é maior do que a primeira precisão de sub-pixeis, quando o vector de movimento tem a segunda precisão de sub-pixeis, descodificar um valor conjuntamente codificado

representativo de valores da diferença do vector de movimento para um componente x e um componente y do vector de movimento, e descodificar o vector de movimento com base nos valores da diferença do vector de movimento e valores de componente x previsto e um componente y previsto para o vector de movimento, e que formam os dados de previsão para a unidade de codificação utilizando o vector de movimento descodificado.

O codificador de vídeo 20 ou descodificador de vídeo 30 podem determinar a resolução do vector de movimento para o bloco actual com base no valor limiar ilustrado na figura 7. Mais especificamente, o codificador de vídeo 20 ou descodificador de vídeo 30 podem determinar se o vector de movimento para o actual bloco tem a primeira precisão de sub-pixel ou o segundo precisão de sub-pixel por determinação de um valor de limiar relacionado com o vector de movimento. O codificador de vídeo 20 ou descodificador de vídeo 30 podem determinar que o vector de movimento tem a primeira precisão de subpixel quando tanto o componente x do valor da diferença do vector de movimento como o componente y do valor de diferença do vector de movimento são maiores do que o limiar.

Quando pelo menos um dos componentes x do valor de diferença do vector de movimento e o componente y do valor da diferença do vector de movimento é inferior a ou igual ao limiar, o codificador de vídeo 20 ou descodificador de vídeo 30 pode determinar se o vector de

movimento tem a primeira precisão de subpixel ou a segunda precisão de sub-pixel com base no valor da referência da resolução do vector de movimento.

Deverá ser reconhecido que, dependendo do exemplo, certos actos ou ocorrências de qualquer das técnicas aqui descritas podem ser realizadas numa sequência diferente, podem ser adicionados, fundidos, ou deixados de fora (por exemplo, nem todas as acções ou ocorrências descritas são necessária para a prática das técnicas). Além disso, em certos exemplos, podem ser realizados simultaneamente actos ou ocorrências, por exemplo, através do multiprocessamento, processamento de interrupção, ou múltiplos processadores, em vez de sequencialmente.

A figura 8 é um fluxograma que ilustra um exemplo do método para a codificação conjunta dos componentes x e y de um valor da diferença do vector de movimento e codificação de dados codificados de entropia para produzir um bloco actual. Os passos do método 280 da figura 8 são geralmente recíprocos ao método 150 da figura 4. No método 280, a unidade de compensação de movimento 72 ou uma outra unidade de codificador de vídeo 30 podem prever um bloco actual (282). Como parte de previsão do bloco actual, a unidade de compensação de movimento 72 pode descodificar os componentes codificados conjuntamente de um valor da diferença do vector de movimento (284). O descodificador de vídeo pode descodificar os componentes codificados conjuntamente da diferença do vector de movimento de acordo

com as técnicas do método 320 ilustrado na figura 9. O valor da diferença do vector de movimento para o bloco actual pode ter um componente x de `mvd_x` e um componente y de `mvd_y`. Os componentes codificados em conjunto do valor da diferença do vector de movimento podem incluir componentes x e y do valor da diferença do vector de movimento para o bloco actual, bem como outra informação relacionada com o valor da diferença do vector de movimento tal como um ou mais valores representativos de se ou não os componentes x e y da diferença do vector de movimento são zero, um valor representativo do valor do sinal dos componentes x e/ou y do valor da diferença do vector de movimento, e um valor representativo do valor absoluto do componente do valor da diferença do vector de movimento, como alguns exemplos não limitativos.

A unidade de compensação de movimento 44 do codificador de vídeo 20 pode também receber um preditor de vector de movimento para o vector de movimento do bloco actual (286). Um preditor de vector de movimento é um vector de movimento que a unidade de compensação de movimento 72 utiliza para calcular os valores da diferença do vector de movimento para o bloco actual. Em alguns exemplos, o preditor do vector de movimento pode ser o vector de movimento de um de uma pluralidade de blocos na vizinhança da PU do bloco actual. A unidade de compensação de movimento 72 pode também determinar o preditor do vector de movimento de um bloco co-localizado de uma das tramas de referência armazenadas na memória da trama de referência

82. O preditor do vector de movimento pode ter um componente x de  $p_x$  e um componente y de  $p_y$ . O preditor do vector de movimento pode ter sido determinado pelo codificador de vídeo 20 durante a fase de codificação de tal modo que o preditor do vector de movimento minimiza a diferença do vector de movimento do bloco actual e o preditor do vector de movimento. Baseado no preditor do vector de movimento, a unidade de compensação de movimento 72 do codificador de vídeo 20 pode calcular um vector de movimento para o bloco actual (286). O descodificador de vídeo 30 pode calcular o vector de movimento como uma soma dos componentes da diferença do vector de movimento e preditor do vector de movimento,  $\langle p_x + mvd_x, p_y + mvd_y \rangle$  (288).

A unidade de descodificação de entropia 70 pode receber dados codificados de entropia para o bloco actual (290), e descodificar dados, tais como os coeficientes codificados do comprimento do percurso transformado de um bloco para reproduzir os coeficientes transformados do bloco (292). A unidade de descodificação de entropia 70 pode varrer inversamente os coeficientes reproduzidos para rearranjar os coeficientes a partir de um ou mais vectores unidimensionais de coeficientes num bloco bidimensional de coeficientes transformados (294).

A unidade de quantificação inversa 76 pode quantificar inversamente o bloco bidimensional de coeficientes de transformação, e a unidade de transformação

inversa 78 pode inversamente transformar os coeficientes de coeficientes inversamente quantificados para produzir um bloco de pixels não transformados (296). O bloco residual pode compreender as diferenças de píxeis entre o bloco de vídeo preditivo, o qual é indicado pelo vector de movimento para o bloco actual, e o bloco de vídeo actual. O descodificador de vídeo 30 pode então combinar o bloco residual com o bloco preditivo para formar o bloco actual da trama actual (298).

As técnicas da figura 8 podem, geralmente, ser realizadas por qualquer unidade de processamento ou processador, sejam implementadas em hardware, software, firmware, ou uma combinação das mesmas, e quando implementadas em software ou firmware, pode ser proporcionado o hardware correspondente para executar instruções para o software ou firmware. Com objectivos de exemplo, as técnicas da figura 8 estão descritas em relação a um dispositivo de descodificação de vídeo, que pode incluir componentes substancialmente semelhantes aos do descodificador de vídeo 30 (figuras 1 e 3), embora deva ser compreendido que outros dispositivos podem ser configurados para executar técnicas semelhantes. Além disso, os passos ilustrados na figura 8 podem ser realizados numa ordem diferente ou em paralelo, podendo ser adicionados passos adicionais e certos passos omitidos, sem fugir das técnicas desta descrição.

A figura 9 é um fluxograma que ilustra um exemplo

do método para a codificação conjunta dos componentes x e y de um valor da diferença do vector de movimento. Os passos do método 320 da figura 9 são geralmente recíprocos aos passos do método 220 da figura 6. Do mesmo modo, o método 320 pode geralmente corresponder ao passo 284 da figura 8. No método 320 da figura 9, a unidade de compensação de movimento 72 do descodificador 30 de vídeo ou outra unidade do descodificador de vídeo 30 podem receber uma representação codificada em conjunto de um vector de movimento para o bloco actual de dados de vídeo. O descodificador de vídeo 30 pode também determinar um preditor do vector de movimento, que um descodificador de vídeo, tal como o codificador de vídeo 20 pode ter sido previamente determinado e codificado. O descodificador vídeo 30 pode calcular um vector de movimento para o bloco actual como a diferença da soma do preditor do vector de movimento para o bloco de dados de vídeo actual e o preditor do vector de movimento determinado para o bloco actual de dados de vídeo.

Para determinar os componentes do valor da diferença do vector de movimento, a unidade de compensação de movimento 72 do descodificador de vídeo 30 pode descodificar informação representativa se o valor absoluto de  $mvd_x$ , o componente x do valor da diferença do vector de movimento, é maior do que zero (322). A unidade de compensação de movimento 72 pode então descodificar informação representativa de se um valor absoluto de um componente y ( $mvd_y$ ) do valor da diferença do vector de

movimento é maior do que zero (324).

Quando o valor absoluto do componente x do valor da diferença do vector de movimento é maior do que zero, a unidade de compensação de movimento 72 pode descodificar informação representativa do valor absoluto do componente x do valor da diferença do vector de movimento (326). E quando o valor absoluto do componente y do valor da diferença do vector de movimento é maior do que zero, a unidade de compensação de movimento 72 pode descodificar informação representativa do valor absoluto do componente y do valor da diferença do vector de movimento (328). Em alguns exemplos, o descodificador de vídeo 30 pode descodificar a informação representativa do valor absoluto dos componentes x e y dos valores da diferença do vector de movimento como os valores absolutos de  $mvd_x/2 - 1$ , e  $mvd_y/2 - 1$ , respectivamente.

Quando o valor absoluto do componente x do valor da diferença do vector de movimento é maior do que zero, a unidade de compensação de movimento 72 pode descodificar um sinal do componente x do valor da diferença do vector de movimento (330). Quando o valor absoluto do componente y do valor da diferença do vector de movimento é maior do que zero, a unidade de compensação de movimento 72 pode descodificar um sinal do componente y do valor da diferença do vector de movimento (332).

As técnicas da figura 9 podem, geralmente, ser

realizadas por qualquer unidade de processamento ou processador, sejam implementadas em hardware, software, firmware, ou uma combinação das mesmas, e quando implementadas em software ou firmware, pode ser proporcionado o hardware correspondente para executar instruções para o software ou firmware. Com objectivos de exemplo, as técnicas da figura 9 estão descritas em relação a um dispositivo de codificação de vídeo, que pode incluir componentes substancialmente semelhantes aos do decodificador de vídeo 30 (figuras 1 e 3), embora deva ser compreendido que outros dispositivos podem ser configurados para executar técnicas semelhantes. Além disso, os passos ilustrados na figura 9 podem ser realizados numa ordem diferente ou em paralelo, podendo ser adicionados passos adicionais e certos passos omitidos, sem fugir das técnicas desta descrição.

Deste modo, o método da figura 9 representa um exemplo de um método de que inclui codificar informação representativa de se um valor absoluto de um componente  $x$  de um valor da diferença do vector de movimento para um bloco actual de dados de vídeo é maior do que zero, codificar informação representativa de se um valor absoluto de um componente  $y$  do valor da diferença de vector de movimento é maior do que zero, quando o valor absoluto do componente  $x$  do valor da diferença do vector de movimento é maior do que zero, codificar informação representativa do valor absoluto do componente  $x$  do valor da diferença do vector de movimento, quando o valor absoluto do componente

y do valor de diferença do vector de movimento é maior do que zero, codificar a informação representativa do valor absoluto do componente y do valor da diferença de vector de movimento, quando o valor absoluto do componente x do valor de diferença do vector de movimento é maior do que zero, codificar um sinal do componente x do valor da diferença do vector de movimento, e quando o valor absoluto do componente y do valor da diferença do vector de movimento é maior do que zero, codificar um sinal do componente y do valor de diferença do vector de movimento. Assim, o método da figura 9 representa um outro exemplo de um método para a codificação de dados em conjunto do valor da diferença do vector de movimento. Além disso, o descodificador de vídeo 30 pode também ser configurado para executar um método de descodificação de vídeo substancialmente semelhante ao método da figura 7. Certas técnicas desta descrição foram avaliadas experimentalmente. Especificamente, as técnicas desta descrição com a sinalização adaptativa de precisão de subpixel foram implementadas para HM3.0 e simulada em condições de ensaio comuns. O desempenho é comparado a âncora HM3.0 em termos de taxa de bits Bjøntegaard Delta (BD). Os resultados das simulações mostraram uma economia média de 0,4% BD da taxa nas configurações de alta eficiência e uma economia média de 0,6% BD da taxa nas configurações de baixa complexidade. A tabela 3 a seguir resume os resultados experimentais:

Tabela 3

	Acesso aleatório HE			Acesso aleatório LC		
	Y	U	V	Y	U	V
Classe A	0,0	-0,3	0,0	0,0	-0,4	-0,1
Classe B	-0,3	-0,3	-0,4	-0,3	-0,2	-0,2
Classe C	-0,5	-0,4	-0,3	-0,6	-0,7	-0,7
Classe D	-0,9	-0,7	-1,0	-1,3	-0,9	-1,0
Classe E						
Global	-0,4	-0,4	-0,4	-0,6	-0,6	-0,5
Tem.cod. [%]		110%			116%	
Tem.desc. [%]		101%			102%	

	Baixo atraso B HE			Baixo atraso B LC		
	Y	U	V	Y	U	V
Classe A						
Classe B	-0,1	0,1	-0,1	-0,1	-0,4	-0,4
Classe C	-0,6	-0,6	-0,6	-0,8	-0,4	-0,5
Classe D	-1,2	-1,4	-1,7	-1,8	0,9	2,2
Classe E	0,4	-0,4	-0,8	0,7	1,1	0,5
Global	-0,4	-0,4	-0,4	-0,6	0,2	0,4
Tem.cod. [%]		110%			117%	
Tem.desc. [%]		101%			103%	

Deverá ser reconhecido que, dependendo do exemplo, certos actos ou ocorrências de qualquer das técnicas aqui descritas podem ser realizadas numa sequência diferente, podem ser adicionados, fundidos, ou deixados de fora (por exemplo, nem todas as acções ou ocorrências descritas são necessária para a prática das técnicas). Além disso, em certos exemplos, podem ser realizados simultaneamente actos ou ocorrências, por exemplo, através do multiprocessamento, processamento de interrupção, ou múltiplos processadores, em vez de sequencialmente.

Em um ou mais exemplos, as funções descritas podem ser implementadas em hardware, software, firmware, ou qualquer combinação destes. Se implementadas em software, as funções podem ser armazenadas em, ou transmitidas como, uma ou mais instruções de código num meio legível em computador e executadas por uma unidade de processamento baseada em hardware. Os suportes legíveis em computador podem incluir os meios legíveis em computador, que correspondem a um meio tangível tal como meio de armazenamento de dados, ou meio de comunicação incluindo qualquer meio que facilite a transferência de um programa de computador de um lugar para o outro, por exemplo, de acordo com um protocolo de comunicações. Deste modo, os meios legíveis por computador podem geralmente corresponder a (1) um meio tangível de armazenamento legível por computador que é não transitório ou (2) um meio de comunicação tal como um sinal ou onda portadora. Um meio de armazenamento de dados pode ser qualquer meio disponível

que pode ser acedido por um ou mais computadores ou um ou mais processadores para recuperar as instruções, código e/ou estruturas de dados para a realização das técnicas descritas na presente memória descritiva. Um produto de programas informáticos pode incluir um suporte legível por computador.

A título de exemplo, e não como limitação, tais suportes 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 suporte que pode ser usado para armazenar o código de programa desejado sob a forma de instruções ou estruturas de dados, e que pode ser acedido por um computador. Deste modo, qualquer ligação é apropriadamente denominada de suporte legível em computador. Por exemplo, se forem transmitidas instruções 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 infravermelho, rádio, microondas, depois o cabo coaxial, cabo de fibra óptica, par trançado, DSL ou tecnologias sem fio tal como infravermelhos, rádio, e microondas, encontram-se incluídos na definição de suporte. Deverá ser entendido, no entanto, que os suportes de armazenamento legíveis em computador e suportes de armazenamento de dados não incluem ligações, ondas portadoras, sinais, ou outros suportes transitórios, mas estão em vez disso direccionados

para suportes de armazenamento tangíveis, não transitórios. Os discos ópticos e discos magnéticos, tal como utilizados na presente, incluem o disco compacto (CD), o disco laser, o disco óptico, o DVD, sendo que os discos ópticos reproduzem os dados opticamente com lasers, sendo que a disquete e o disco rígido reproduzem os dados magneticamente. A combinação do acima mencionado deve também ser incluída dentro do âmbito dos suportes legíveis em computador.

As instruções podem ser executadas por um ou mais processadores, tal como um ou mais processadores de sinais digitais (DSPs), microprocessadores de uso geral, circuitos integrados para aplicação específica (ASIC), matrizes de lógica programáveis pelo utilizador (FPGA), ou outros circuitos lógicos integrados ou discretos equivalentes. Por conseguinte, o termo "processador", tal como aqui utilizado pode referir-se a qualquer uma das estruturas anteriores ou qualquer outra estrutura adequada para aplicação das técnicas aqui descritas. Adicionalmente, em alguns aspectos, a funcionalidade descrita na presente pode ser fornecida dentro de módulos dedicado de hardware e/ou software configurados para a codificação e descodificação, ou incorporados num codec combinado. Além disso, as técnicas podem ser totalmente implementadas em um ou mais circuitos ou elementos lógicos.

As técnicas da presente memória descritiva podem ser implementadas numa ampla variedade de dispositivos ou

aparelhos, incluindo um microtelefone sem fios, um circuito integrado (CI) ou um conjunto de CIs (por exemplo, um conjunto de chips). Vários componentes, módulos ou unidades são descritos na presente memória descritiva para realçar os aspectos funcionais de dispositivos configurados para executarem as técnicas descritas, mas não precisam necessariamente de realização por diferentes unidades de hardware. Em vez disso, tal como descrito acima, podem ser combinadas várias unidades numa unidade de hardware de codec ou fornecida por um conjunto de unidades de hardware interoperativas, incluindo um ou mais processadores, tal como descrito acima, em conjugação com software e/ou firmware adequado.

Foram descritos vários exemplos. Estes e outros exemplos encontram-se dentro do escopo das reivindicações anexas.

Lisboa, 2 de Outubro de 2015

**REIVINDICAÇÕES**

1. Método (220; 320) de codificação por entropia de dados de vídeo, que compreende:

quando um valor absoluto de um componente x de um valor da diferença do vector de movimento para um bloco actual é maior do que zero (226; 326), e quando um valor absoluto de um componente y do valor da diferença do vector de movimento para o bloco actual é maior do que zero (228; 328), informação representativa da codificação de entropia representativa de um valor da diferença do vector de movimento em que a codificação da entropia da informação representativa do valor da diferença do vector de movimento compreende:

intercalar informação representativa de se o valor absoluto do componente x de um valor da diferença do vector de movimento para o bloco actual de dados de vídeo é maior do que zero, e informação representativa de se o valor absoluto do componente y do valor da diferença do vector de movimento é maior do que zero; e intercalar informação representativa do valor absoluto do componente x do valor da diferença do vector de movimento, informação representativa do valor absoluto do componente y do valor da diferença do vector de movimento, um sinal do componente x do valor da diferença do vector de movimento (230; 330), e um sinal do

componente  $y$  do valor da diferença do vector de movimento (232; 332).

2. Método de acordo com a reivindicação 1, em que a codificação de acordo com a codificação de entropia compreende codificação aritmética binária contexto-adaptativa, CABAC, compreendendo adicionalmente:

calcular um vector de movimento para o bloco actual de dados de vídeo; e

calcular (156) o valor da diferença do vector de movimento como a diferença entre o vector de movimento para o bloco actual dos dados de vídeo actual e um preditor do vector de movimento determinado para o bloco actual de dados de vídeo.

3. Método de acordo com a reivindicação 1, em que a descodificação de acordo com a codificação de entropia compreende codificação aritmética binária contexto-adaptativa, CABAC, compreendendo o método adicionalmente:

determinar (286) um preditor de vector de movimento para o bloco actual de dados de vídeo; e

calcular (288) o vector de movimento para o bloco actual de dados de vídeo como uma soma do valor da diferença do vector de movimento e do preditor do vector de movimento.

4. Método de acordo com a reivindicação 1, que compreende adicionalmente:

codificar um valor representativo de se um vector de movimento do bloco actual de dados de video tem uma primeira precisão de sub-pixel, ou uma segunda precisão de sub-pixel, em que a segunda precisão de sub-pixel é maior do que a primeira precisão de sub-pixel; e

quando o vector de movimento tem a segunda precisão de sub-pixel, quando o componente x do valor da diferença do vector de movimento não é igual a zero, e quando o componente y do valor da diferença do vector de movimento não é igual a zero, em que a informação de codificação representativa do valor absoluto do componente x do valor da diferença do vector de movimento e informação de codificação representativa do valor absoluto do componente y do valor da diferença do vector de movimento compreende codificar um valor codificado em conjunto representativo de ambos o componente x do valor da diferença do vector de movimento e o componente y do valor da diferença do vector de movimento.

5. Método de acordo com a reivindicação 4, compreendendo adicionalmente determinar se o vector de movimento tem a primeira precisão de sub-pixel ou a segunda precisão de sub-pixel, compreendendo:

determinar um valor limiar associado ao vector de movimento; e

determinar que o vector de movimento tem a primeira precisão de subpixel quando tanto o componente x do

valor da diferença do vector de movimento como o componente  $y$  do valor da diferença do vector de movimento são maiores do que o limiar.

6. Método de acordo com a reivindicação 5, em que a determinação se o vector de movimento tem a primeira precisão de sub-pixel ou a segunda precisão de sub-pixel compreende, quando pelo menos um dos componentes  $x$  do valor da diferença do vector de movimento e o componente  $y$  do valor da diferença do vector de movimento é menor ou igual ao limiar, determinar se o vector de movimento tem a primeira precisão de subpixel ou a segunda precisão de sub-pixel com base num valor de uma referência da resolução do vector de movimento.

7. Método de acordo com a reivindicação 1, em que a informação de codificação representativa do valor absoluto do componente  $x$  dos valores da diferença do vector de movimento compreende codificar o valor absoluto do componente  $x$  do valor da diferença do vector de movimento como  $mvd\_x/2 - 2$ , em que  $mvd\_x$  compreende o componente  $x$  do valor da diferença do vector de movimento; e em que a informação de codificação representativa do valor absoluto do componente  $y$  dos valores da diferença do vector de movimento compreende codificar o valor absoluto do componente  $y$  dos valores da diferença do vector de movimento como  $mvd\_y/2 - 1$ , em que  $mvd\_y$  compreende o componente  $y$  do valor da diferença do vector de movimento.

8. Dispositivo (56; 70) de codificação por entropia de dados de vídeo, compreendendo:

meios para a codificação por entropia de informação representativa de um valor da diferença do vector de movimento quando o valor absoluto do componente x do valor da diferença do vector de movimento é maior do que zero e quando o valor absoluto do componente y do valor da diferença do vector de movimento é maior do que zero, em que os meios para a codificação por entropia da informação representativa do valor da diferença do vector de movimento compreendem:

meios para intercalar informação representativa se um valor absoluto de um componente x de um valor da diferença do vector de movimento para um bloco actual de dados de vídeo é maior do que zero e informação representativa de se um valor absoluto de um componente y do valor da diferença do vector de movimento é maior do que zero;

meios para intercalar informação representativa do valor absoluto do componente x do valor da diferença do vector de movimento, informação representativa do valor absoluto do componente y do valor da diferença do vector de movimento,

um sinal do componente x do valor da diferença do vector de movimento, e um sinal do componente y do valor da diferença do vector de movimento.

9. Dispositivo de acordo com a reivindicação 8, em que os meios para a codificação de entropia

compreenderem meios para a codificação aritmética binária contexto-adaptativa, CABAC, o dispositivo compreendendo adicionalmente:

meios para calcular um vector de movimento para o bloco actual de dados de vídeo; e  
meios para calcular o valor da diferença do vector de movimento como uma diferença entre o vector de movimento para o bloco actual dos dados de vídeo e um preditor do vector de movimento determinado para o bloco actual de dados de vídeo.

10. Dispositivo de acordo com a reivindicação 8, em que os meios para a codificação de entropia compreenderem meios para a descodificação de acordo com a codificação aritmética binária contexto-adaptativa, CABAC, o dispositivo compreendendo adicionalmente:

meios para determinar um preditor de vector de movimento para o bloco actual de dados de vídeo; e  
meios para calcular o vector de movimento para o bloco actual de dados de vídeo como uma soma do valor da diferença do vector de movimento e o preditor do vector de movimento.

11. Dispositivo de acordo com a reivindicação 8, compreendendo adicionalmente:

meios para codificar um valor representativo de se um vector de movimento do bloco actual de dados de vídeo tem uma primeira precisão de sub-pixel, ou uma segunda precisão de sub-pixel, em que a segunda precisão de

sub-pixel é maior do que a primeira precisão de sub-pixel; e  
meios para codificar um valor codificado em conjunto representativo de ambos o componente x do valor da diferença do vector de movimento e o componente y do valor da diferença do vector de movimento quando o vector de movimento tem a segunda precisão de sub-pixel, quando o componente x do valor da diferença do vector de movimento não é igual a zero, e quando o componente y do valor da diferença do vector de movimento não é igual a zero.

12. Dispositivo de acordo com a reivindicação 11, compreendendo adicionalmente meios para determinar se o vector de movimento tem a primeira precisão de sub-pixel ou a segunda precisão de sub-pixel, compreendendo:

meios para determinar um valor limiar associado ao vector de movimento; e  
meios para determinar que o vector de movimento tem a primeira precisão de subpixel quando ambos o componente x do valor da diferença do vector de movimento e o componente y do valor da diferença do vector de movimento são maiores do que o limiar.

13. Dispositivo de acordo com a reivindicação 12, em que os meios para determinar se o vector de movimento tem a primeira precisão de sub-pixel ou a segunda precisão de sub-pixel compreendem meios para determinar se o vector de movimento tem a primeira precisão de sub-pixel

ou a segunda precisão de sub-pixel com base num valor da referência da resolução do vector de movimento quando pelo menos um dos componentes x do valor da diferença do vector de movimento e o componente y do valor da diferença do vector de movimento é menor ou igual ao limiar.

14. Dispositivo de acordo com a reivindicação 8, em que os meios para codificar informação representativa do valor absoluto do componente x dos valores da diferença do vector de movimento compreendem codificar o valor absoluto do componente x do valor da diferença do vector de movimento como  $mvd\_x/2 - 1$ , em que  $mvd\_x$  compreende o componente x do valor da diferença do vector de movimento; e em que os meios para codificar a informação representativa do valor absoluto do componente y dos valores da diferença do vector de movimento compreendem meios para codificar o valor absoluto do componente y dos valores da diferença do vector de movimento como  $mvd\_y/2-1$ , em que  $mvd\_y$  compreende o componente y do valor da diferença do vector de movimento.

15. Produto de programas informáticos, compreendendo um suporte de armazenamento legível por computador, que apresenta armazenado no mesmo instruções que, quando executadas, fazem com que um ou mais processadores de um dispositivo para a codificação de dados

de vídeo executem o método de qualquer das reivindicações 1 a 7.

Lisboa, 2 de Outubro de 2015

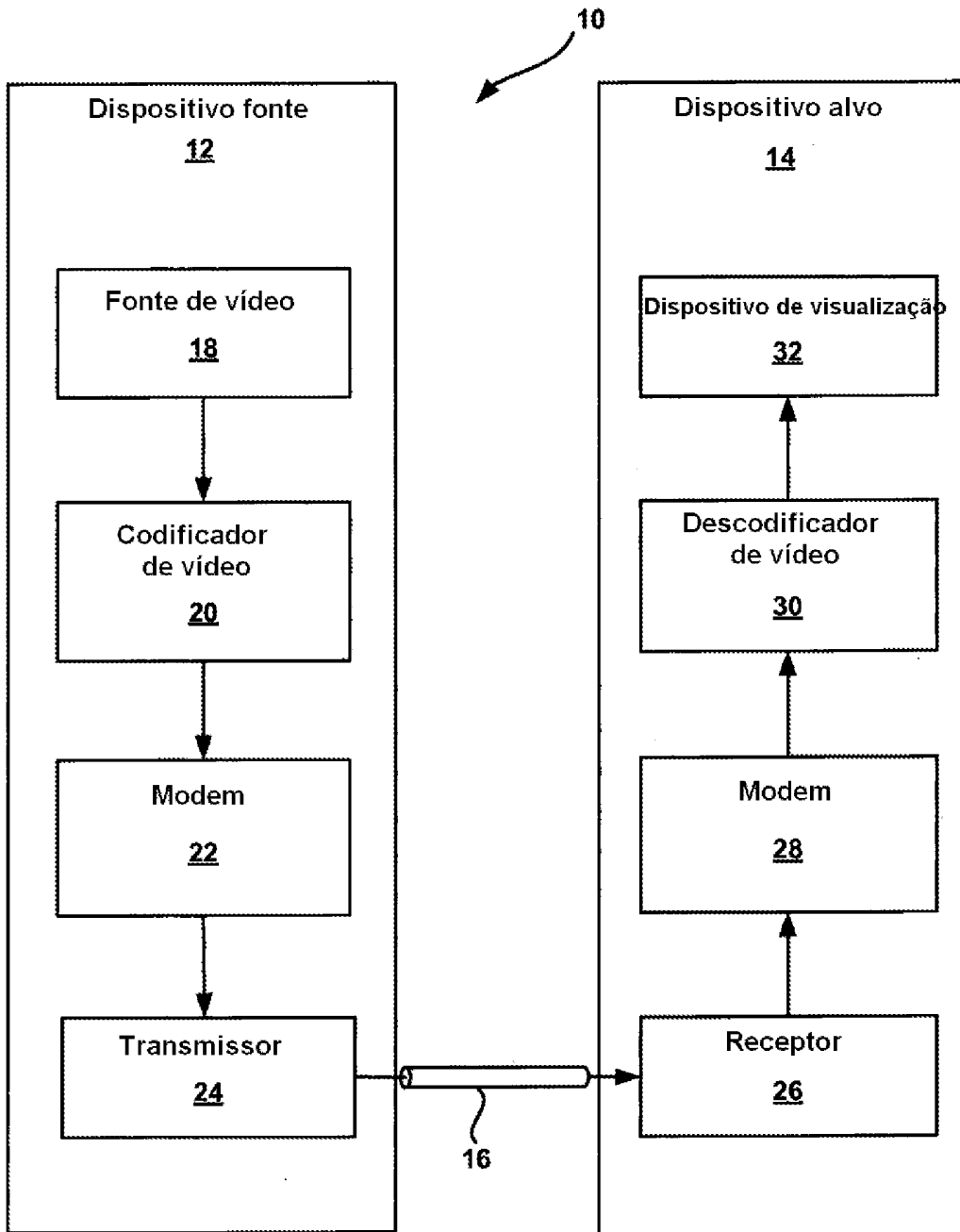


FIG. 1

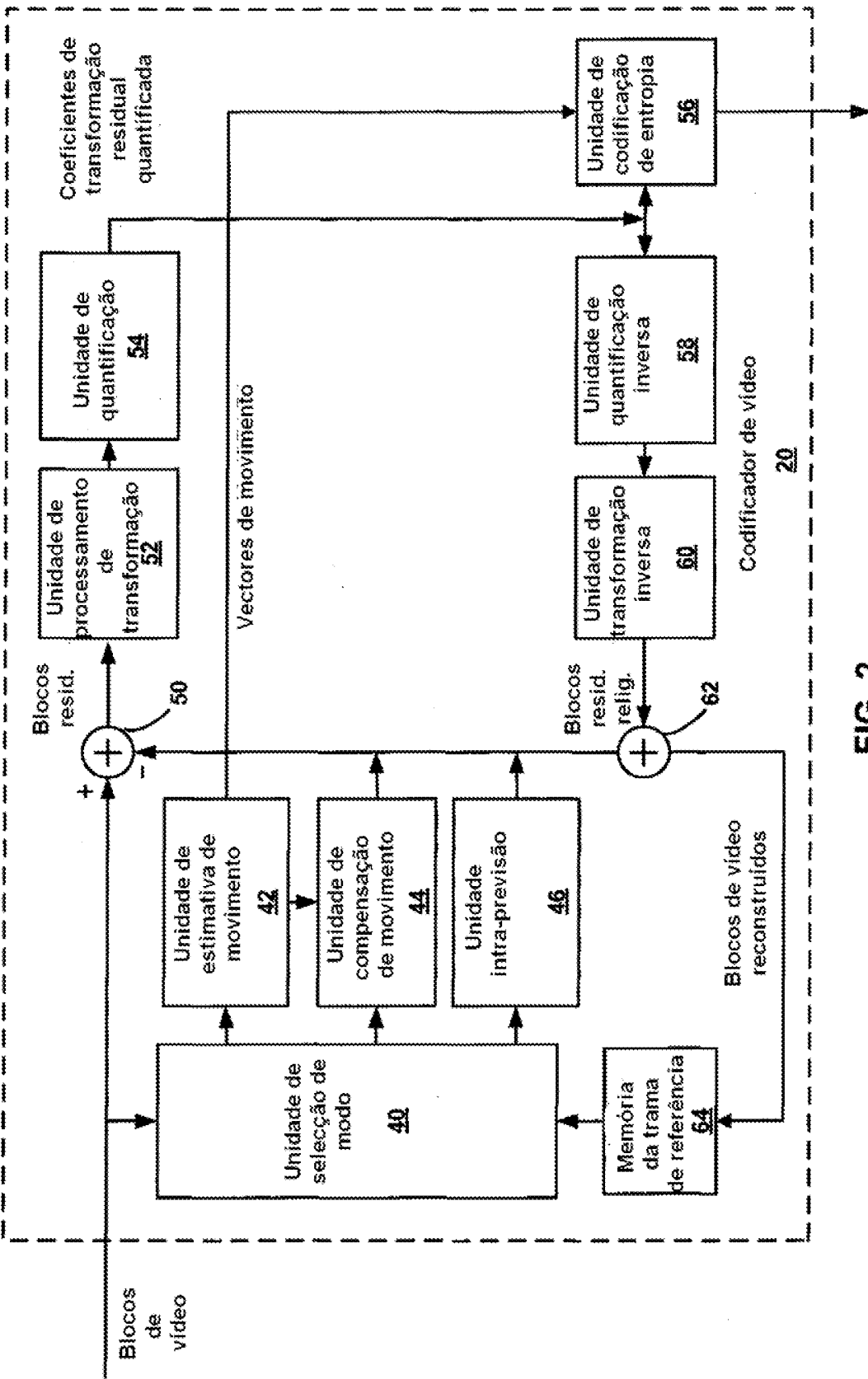


FIG. 2

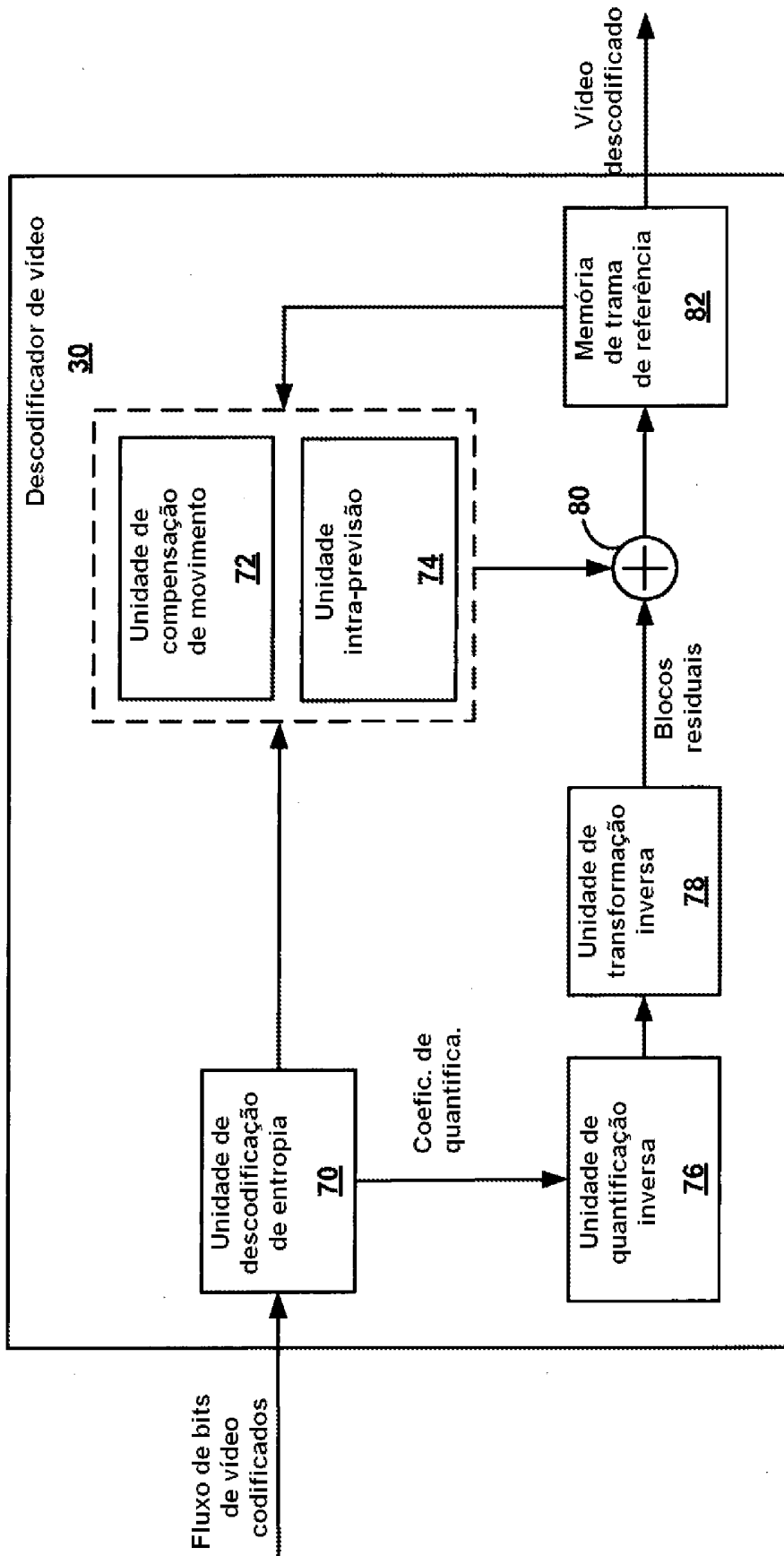


FIG. 3

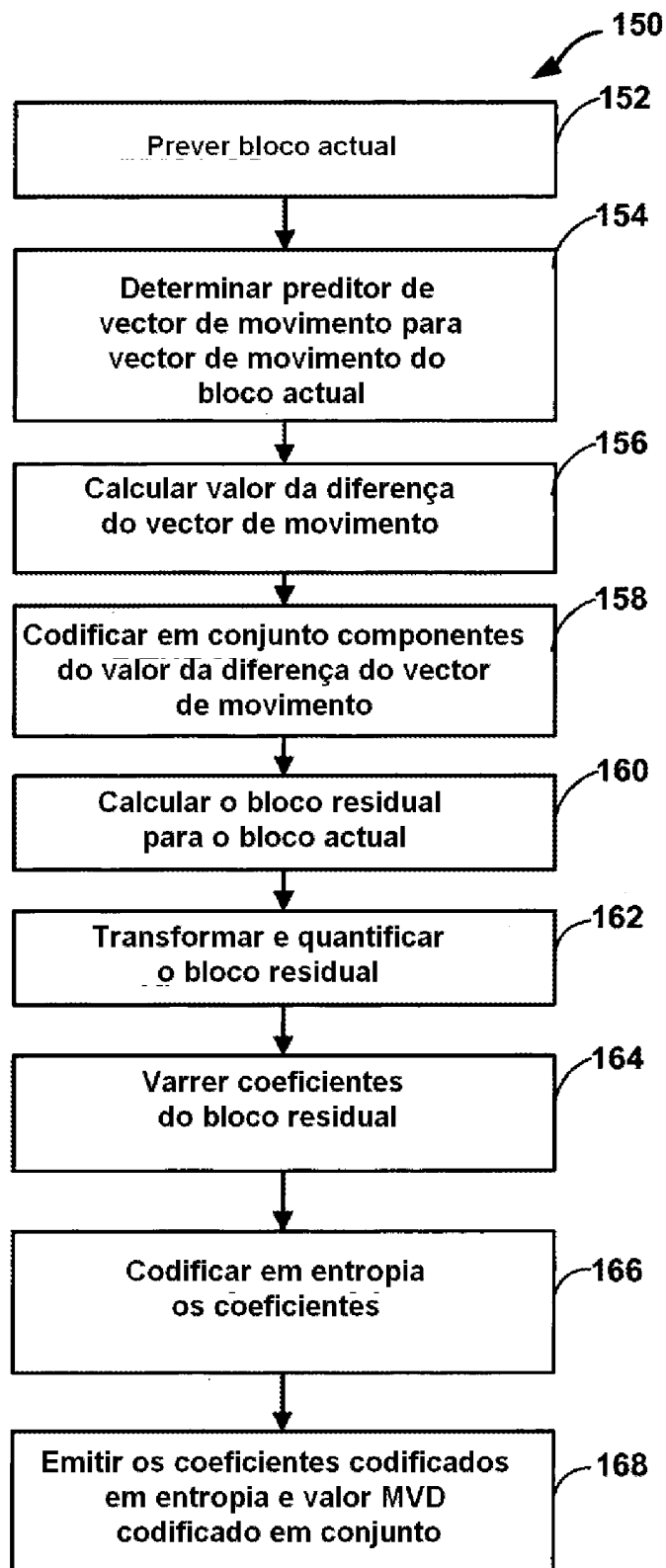


FIG. 4

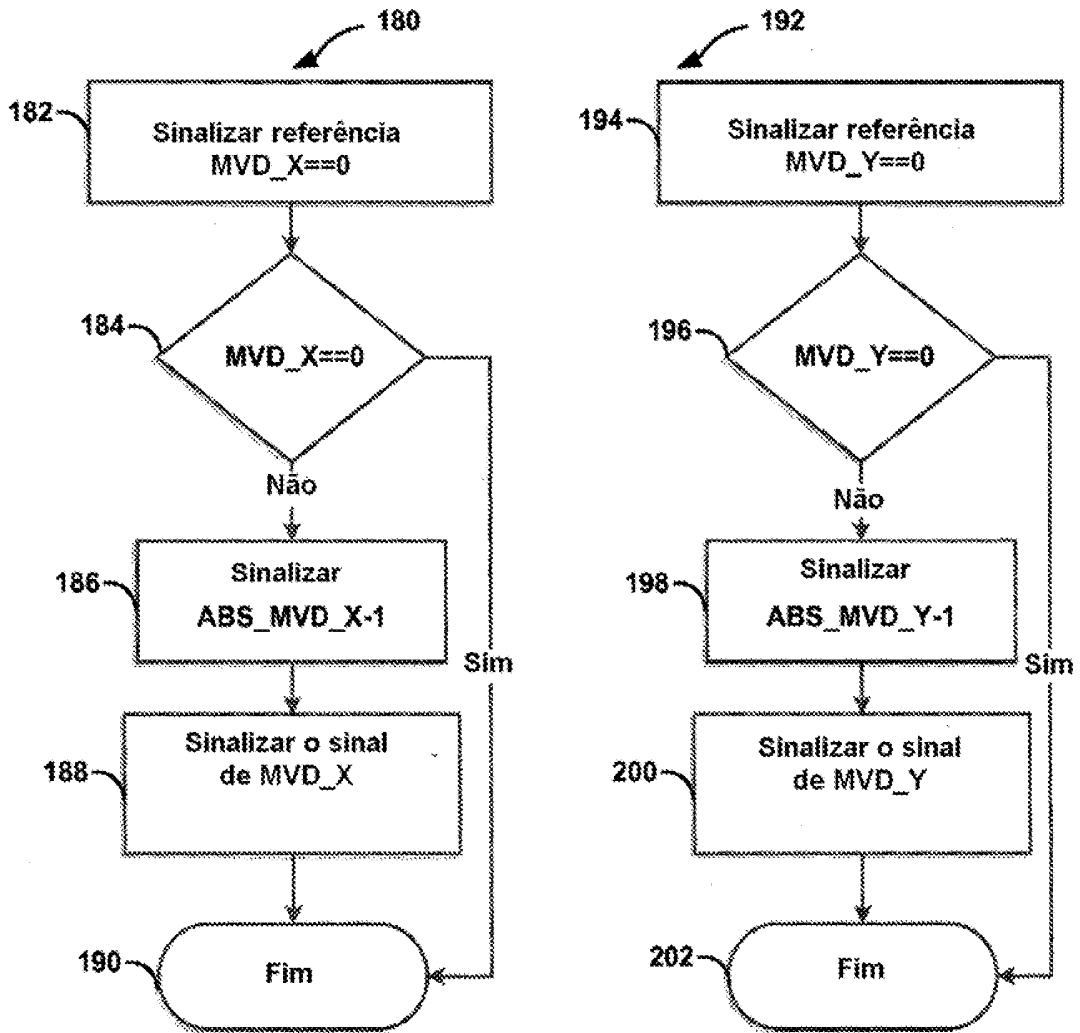


FIG. 5

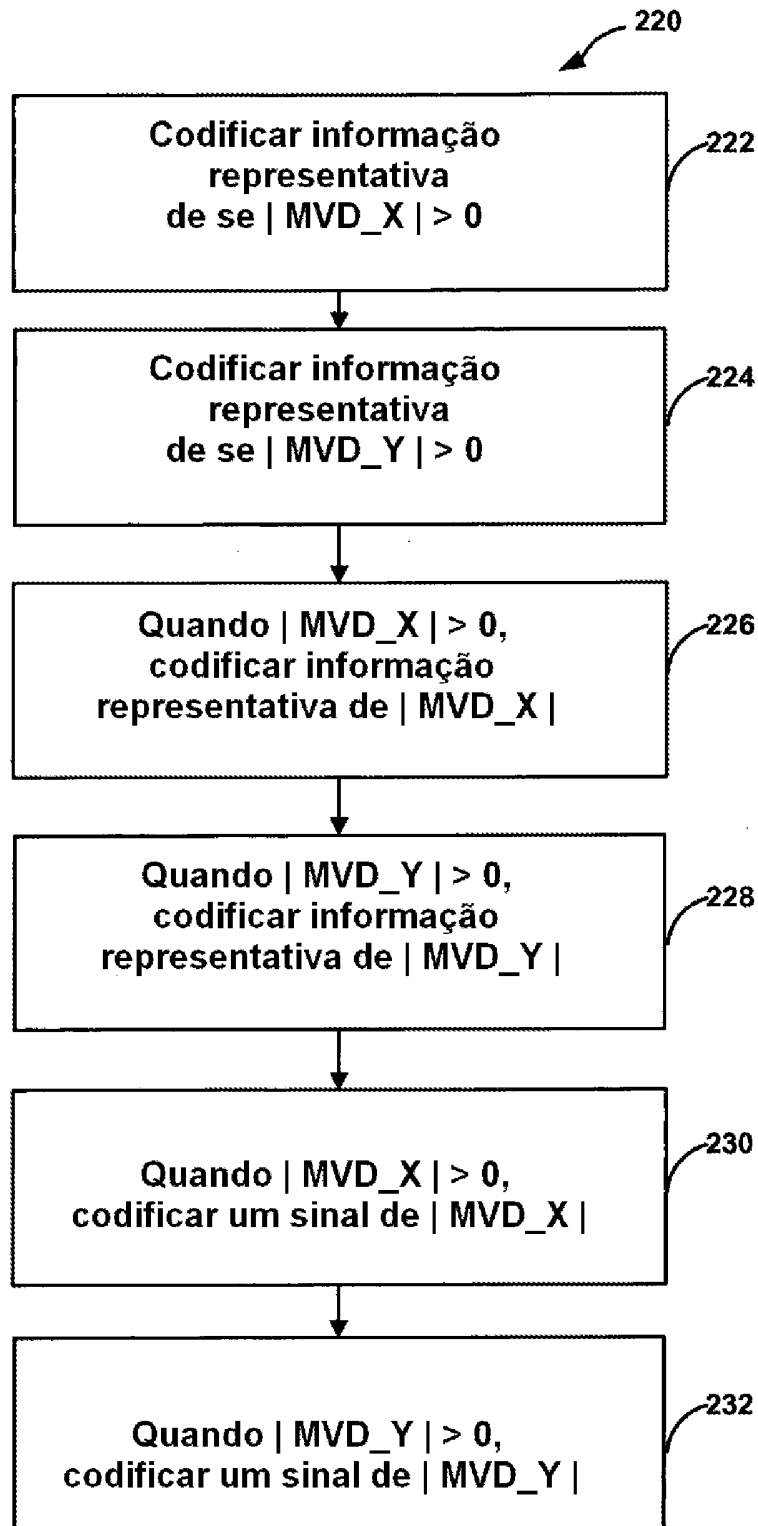


FIG. 6

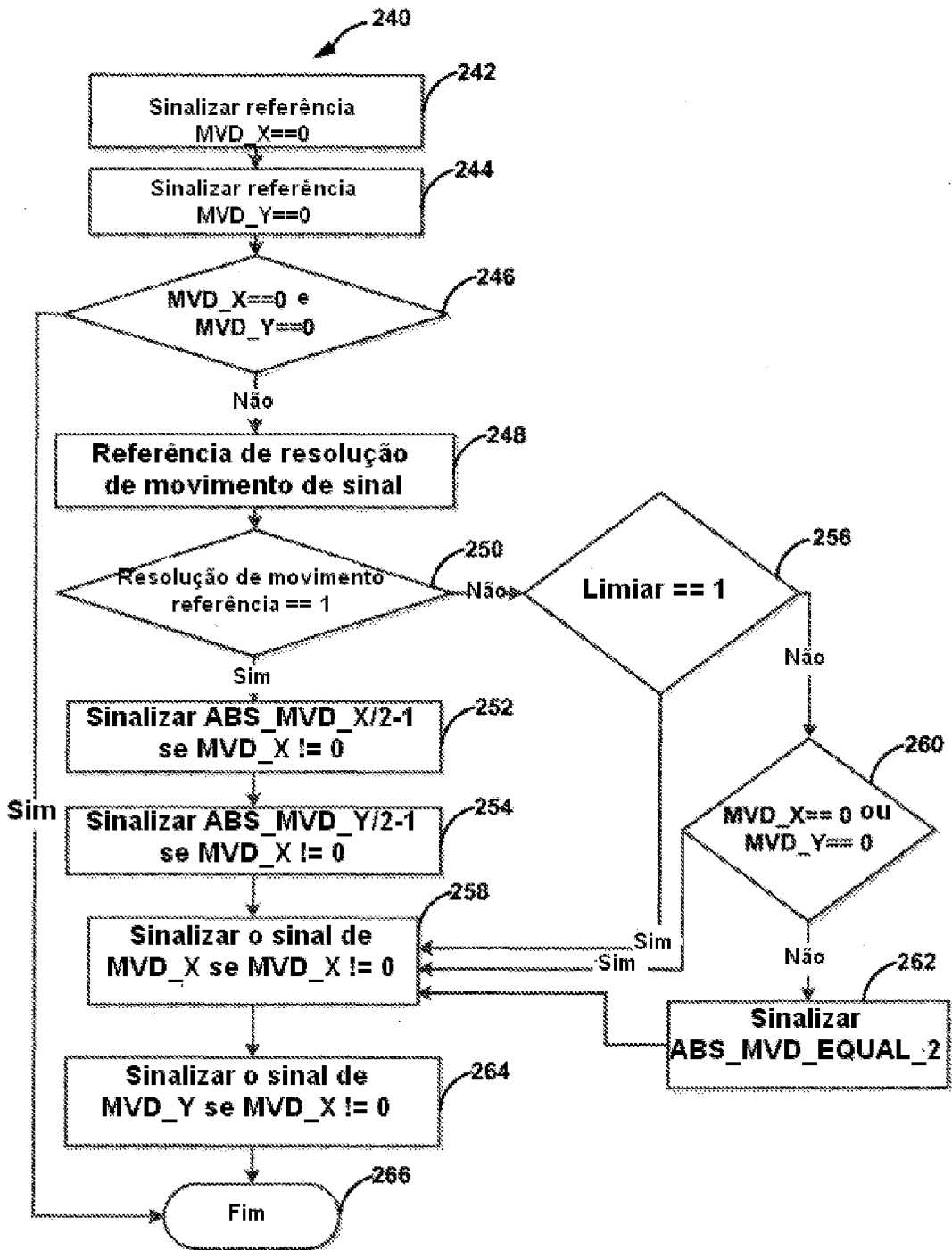


FIG. 7

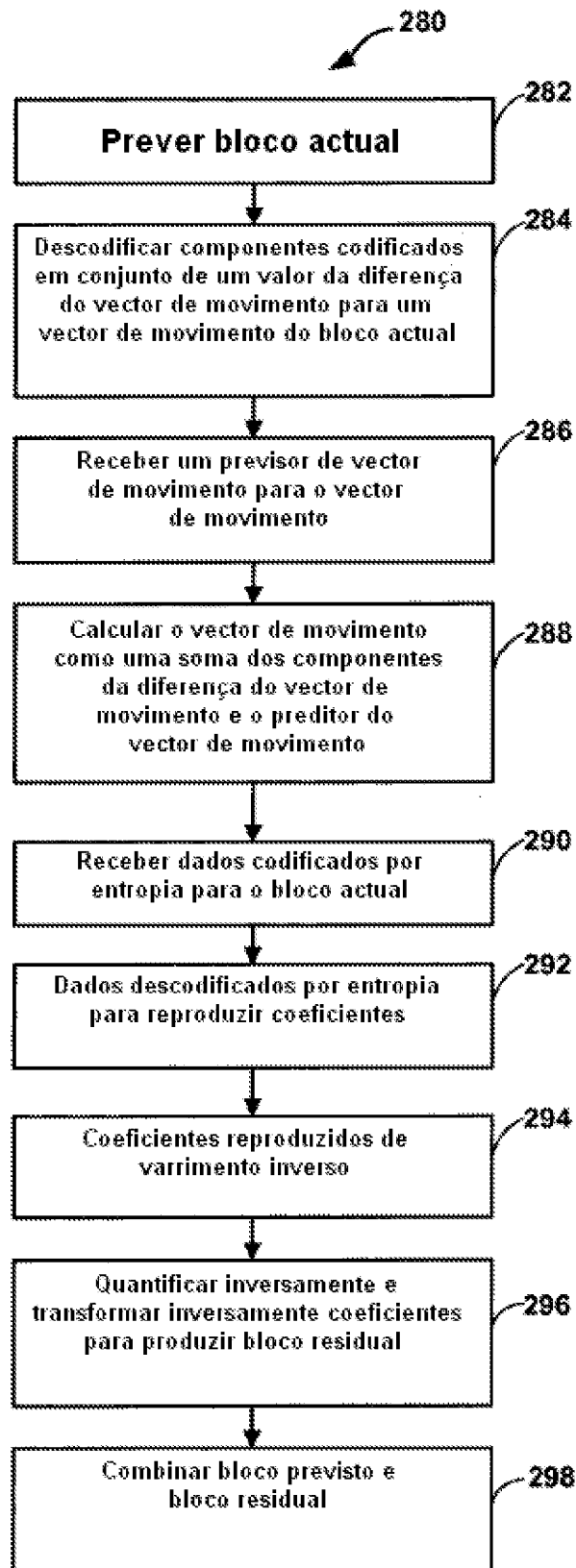


FIG. 8

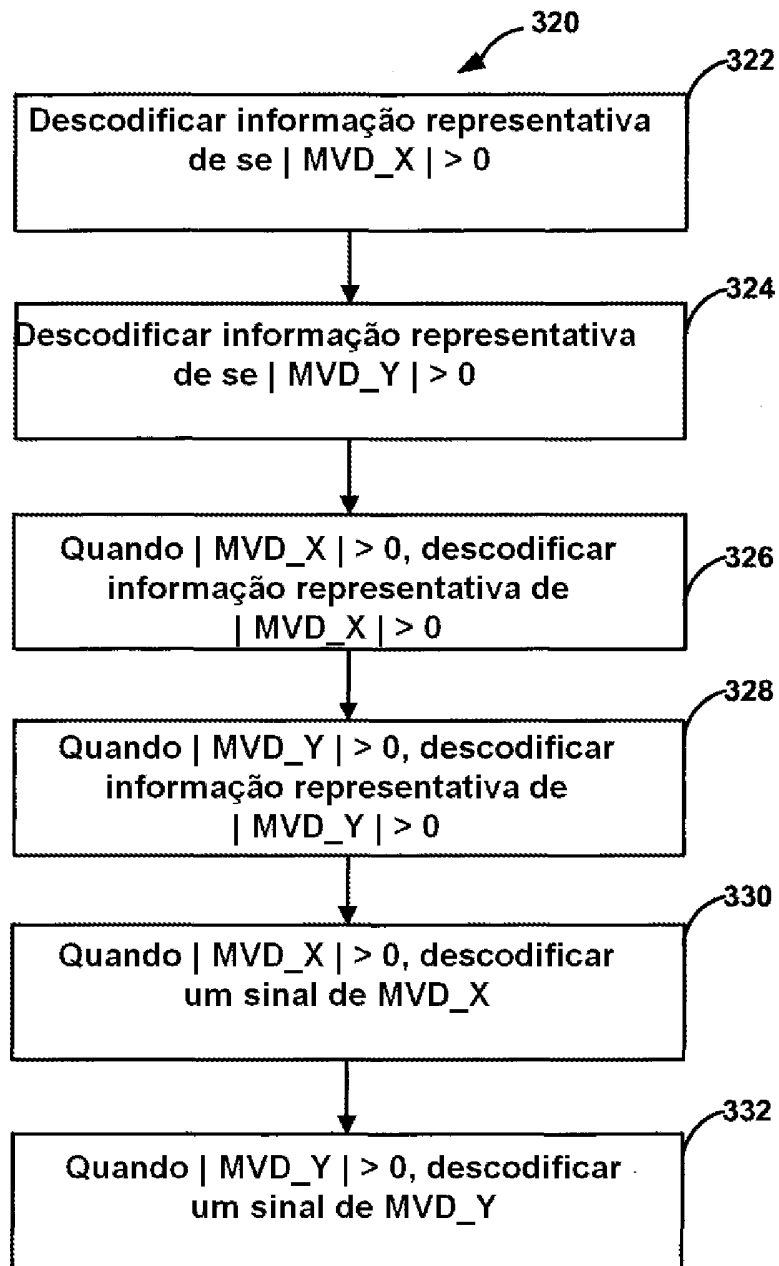


FIG. 9

**REFERÊNCIAS CITADAS NA DESCRIÇÃO**

*Esta lista de referências citadas pelo requerente é apenas para conveniência do leitor. A mesma não faz parte do documento da patente europeia. Ainda que tenha sido tomado o devido cuidado ao compilar as referências, podem não estar excluídos erros ou omissões e o IEP declina quaisquer responsabilidades a esse respeito.*

**Documentos de patentes citadas na Descrição**

- US 2005038837 A

**Literatura que não é de patentes citada na Descrição**

- MARPE et al. Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. *IEEE Transactions on Circuits and Systems*, 01 July 2003, vol. 13 (7), 620-636
- CE12: Adaptive Motion Vector Resolution from Qualcomm. 4. JCT-VC Meeting; 95. MPEG Meeting, 16 January 2011