



**República Federativa do Brasil**

Ministério do Desenvolvimento, Indústria,  
Comércio e Serviços

Instituto Nacional da Propriedade Industrial

**(11) BR 112015024766-0 B1**

**(22) Data do Depósito:** 25/03/2014

**(45) Data de Concessão:** 28/03/2023

**(54) Título:** DESABILITAÇÃO DA OCULTAÇÃO DE DADOS DE SINAL NA CODIFICAÇÃO DE VÍDEO

**(51) Int.Cl.:** H04N 19/70; H04N 19/467; H04N 19/593.

**(30) Prioridade Unionista:** 05/04/2013 US 61/809,199; 05/04/2013 US 61/809,203; 05/07/2013 US 61/843,144; 08/04/2013 US 61/809,811; 08/04/2013 US 61/809,870; (...).

**(73) Titular(es):** QUALCOMM INCORPORATED.

**(72) Inventor(es):** MARTA KARCZEWICZ; JOEL SOLE ROJALS; RAJAN LAXMAN JOSHI.

**(86) Pedido PCT:** PCT US2014031734 de 25/03/2014

**(87) Publicação PCT:** WO 2014/160714 de 02/10/2014

**(85) Data do Início da Fase Nacional:** 25/09/2015

**(57) Resumo:** DESABILITAÇÃO DA OCULTAÇÃO DE DADOS DE SINAL NA CODIFICAÇÃO DE VÍDEO Um codificador de vídeo pode determinar que a ocultação de dados de sinal é desabilitada para um bloco atual se o bloco atual é gerado usando a codificação com perda sem a aplicação de uma transformada aos dados residuais e o bloco atual é intra previsto usando um modo de intra-previsão no qual uma técnica de modulação de código de pulso diferencial residual (DPCM) é usada.

**DESABILITAÇÃO DA OCULTAÇÃO DE DADOS DE SINAL NA  
CODIFICAÇÃO DE VÍDEO**

**[0001]** Este pedido reivindica o benefício do Pedido de Patente Provisório US. Nº 61/805.094, depositado em 25 de março de 2013, Pedido de Patente Provisório US. Nº 61/809.199, depositado em 5 de abril de 2013, Pedido de Patente Provisório US. Nº 61/809,203, depositado em 5 de abril de 2013, Pedido de Patente Provisório US. Nº 61/809,811, depositado em 8 de abril de 2013, Pedido de Patente Provisório US. Nº 61/809,870, depositado em 8 de abril de 2013, Pedido de Patente Provisório US. Nº 61/810,179, depositado em 9 de abril de 2013, Pedido de Patente Provisório US. Nº 61/810,218, depositado em 9 de abril de 2013, e Pedido de Patente Provisório US. Nº 61/843,144, depositado em 5 de julho de 2013, o conteúdo total de cada um dos quais é incorporado aqui a título de referência.

**CAMPO TÉCNICO**

**[0002]** Esta revelação refere-se a codificação e compressão de vídeo.

**FUNDAMENTO**

**[0003]** As capacidades de vídeo digital podem ser incorporadas em uma ampla faixa de dispositivos, incluindo televisores digitais, sistemas de transmissão direta digitais, sistemas de transmissão remotos, assistentes digitais pessoais (PDAs), computadores laptop ou desktop, câmeras digitais, dispositivos de gravação digitais, reprodutores de mídia digitais, dispositivos de vídeo game, consoles de vídeo game, telefones celulares ou de rádio por satélite, dispositivos de teleconferência por vídeo, e similares. Dispositivos de vídeo digital implementam técnicas de compressão de vídeo, como aquelas definidas nos padrões definidos por MPEG-2, MPEG-4, ITU-T

H.263, ITU-T H.264/MPEG-4, Parte 10, Codificação de vídeo avançada (AVC), o padrão de Codificação de Vídeo de Alta Eficiência (HEVC), e extensões de tais padrões, para transmitir, receber e armazenar a informação de vídeo digital de modo mais eficiente.

**[0004]** As técnicas de compressão de vídeo realizam a previsão espacial (intra-imagem) e/ou previsão temporal (inter-imagem) para reduzir ou remover a redundância inerente às sequências de vídeo. Para a codificação de vídeo baseada em bloco, uma fatia de vídeo pode ser dividida em blocos de vídeo, que também podem ser referidos como treeblocks, unidades de codificação (CUs) e/ou nós de codificação. Os blocos de vídeo em uma fatia intra-codificada (I) de uma imagem são codificados usando a previsão espacial com relação às amostras de referência nos blocos vizinhos na mesma imagem. Os blocos de vídeo em uma fatia inter-codificada (P ou B) de uma imagem pode usar a previsão espacial com relação às amostras de referência nos blocos vizinhos na mesma imagem ou previsão temporal com relação às amostras de referência nas outras imagens de referência. As imagens podem ser referidas como quadros, e as imagens de referência podem ser referidas como quadros de referência.

#### SUMÁRIO

**[0005]** Em geral, um codificador de vídeo pode gerar um bitstream que inclui uma sequência de bits que forma uma representação codificada dos dados de vídeo. Como parte da geração do bitstream, o codificador de vídeo pode determinar que a ocultação de dados de sinal é desabilitada para um bloco atual se o bloco atual é gerado usando a codificação com perda sem a aplicação de uma transformada aos dados residuais e o bloco atual é intra previsto usando um modo de intra-previsão no qual uma

técnica de modulação de código de pulso diferencial residual (DPCM) é usada. De modo similar, um decodificador de vídeo pode obter elementos de sintaxe a partir de um bitstream que inclui uma sequência de bits que forma uma representação codificada dos dados de vídeo. Como parte da obtenção dos elementos de sintaxe do bitstream, o decodificador de vídeo pode determinar que a ocultação de dados de sinal é desabilitada para um bloco atual se o bloco atual é gerado usando a codificação com perda sem a aplicação de uma transformada aos dados residuais e o bloco atual é intra previsto usando um modo de intra-previsão no qual uma técnica de DPCM residual é usada. O decodificador de vídeo pode reconstruir uma imagem dos dados de vídeo com base pelo menos em parte nos elementos de sintaxe obtidos a partir do bitstream.

**[0006]** Em um exemplo, esta revelação descreve um método de decodificação de dados de vídeo, o método compreendendo: obter elementos de sintaxe de um bitstream que inclui uma sequência de bits que forma uma representação codificada dos dados de vídeo, sendo que obter os elementos de sintaxe do bitstream compreende determinar que a ocultação de dados de sinal é desabilitada para um bloco atual se o bloco atual é gerado usando codificação com perda sem a aplicação de uma transformada aos dados residuais e o bloco atual é intra previsto usando um modo de intra-previsão no qual uma técnica de modulação de código de pulso diferencial residual (DPCM) é usada; e reconstruir uma imagem dos dados de vídeo com base pelo menos em parte nos elementos de sintaxe obtidos a partir do bitstream.

**[0007]** Em outro exemplo, esta revelação descreve um método de codificação de dados de vídeo, o método compreendendo: gerar um bitstream que inclui uma



sequência de bits que forma uma representação codificada dos dados de vídeo, sendo que gerar o bitstream compreende determinar que a ocultação de dados de sinal é desabilitada para um bloco atual se o bloco atual é gerado sem a aplicação de uma transformada aos dados residuais e o bloco atual é intra previsto usando um modo de intra-previsão no qual uma técnica de modulação de código de pulso diferencial residual (DPCM) é usada; e emitir o bitstream.

**[0008]** Em outro exemplo, esta revelação descreve um aparelho de codificação de vídeo compreendendo: uma memória que armazena dados; e um ou mais processadores configurados para determinar que a ocultação de dados de sinal é desabilitada para um bloco atual se o bloco atual é gerado usando codificação com perda sem a aplicação de uma transformada aos dados residuais e o bloco atual é intra previsto usando um modo de intra-previsão no qual uma técnica de modulação de código de pulso diferencial residual (DPCM) é usada.

**[0009]** Em outro exemplo, esta revelação descreve um aparelho de decodificação de vídeo compreendendo: meios para obter elementos de sintaxe de um bitstream que inclui uma sequência de bits que forma uma representação codificada dos dados de vídeo, sendo que obter os elementos de sintaxe do bitstream compreende determinar que a ocultação de dados de sinal é desabilitada para um bloco atual se o bloco atual é gerado usando codificação com perda sem a aplicação de uma transformada aos dados residuais e o bloco atual é intra previsto usando um modo de intra-previsão no qual uma técnica de modulação de código de pulso diferencial residual (DPCM) é usada; e meios para reconstruir uma imagem dos dados de vídeo com base pelo menos em parte nos elementos de sintaxe obtidos a partir do bitstream.

**[0010]** Em outro exemplo, esta revelação descreve um aparelho de codificação de vídeo compreendendo: meios para gerar um bitstream que inclui uma sequência de bits que forma uma representação codificada dos dados de vídeo, sendo que gerar o bitstream compreende determinar que a ocultação de dados de sinal é desabilitada para um bloco atual se o bloco atual é gerado usando codificação com perda sem a aplicação de uma transformada aos dados residuais e o bloco atual é intra previsto usando um modo de intra-previsão no qual uma técnica de modulação de código de pulso diferencial residual (DPCM) é usada; e meios para emitir o bitstream.

**[0011]** Em outro exemplo, esta revelação descreve um mídia de armazenamento de dados legíveis por computador não transitória tendo instruções armazenadas nela que, quando executadas, fazem com que um ou mais processadores: determinem que a ocultação de dados de sinal é desabilitada para um bloco atual se o bloco atual é gerado usando codificação com perda sem a aplicação de uma transformada aos dados residuais e o bloco atual é intra previsto usando um modo de intra-previsão no qual uma técnica de modulação de código de pulso diferencial residual (DPCM) é usada.

**[0012]** Os detalhes de um ou mais exemplos da revelação são apresentados nos desenhos anexos e a descrição abaixo. Outros recursos, objetos e vantagens serão evidentes a partir da descrição, desenhos e reivindicações.

#### BREVE DESCRIÇÃO DOS DESENHOS

**[0013]** FIG. 1 é um diagrama em bloco ilustrando um sistema de codificação de vídeo exemplar que pode usar as técnicas dessa revelação.

**[0014]** FIG. 2 é um diagrama conceitual que ilustra um bloco de tamanho M (altura) × N (largura).

**[0015]** FIG. 3 é um diagrama conceitual que ilustra as direções do modo de intra-previsão exemplar.

**[0016]** FIG. 4 é um diagrama conceitual que ilustra amostras exemplares que podem ser usadas para a previsão na codificação de vídeo.

**[0017]** FIG. 5A mostra uma direção da modulação do código de pulso diferencial residual (DPCM) para os modos quase verticais.

**[0018]** FIG. 5B mostra uma direção de DPCM residual para os modos quase horizontais.

**[0019]** FIG. 6 é um diagrama em bloco ilustrando um codificador de vídeo exemplar que pode implementar as técnicas desta revelação.

**[0020]** FIG. 7 é um diagrama em bloco ilustrando um decodificador de vídeo exemplar que pode implementar as técnicas desta revelação.

**[0021]** FIG. 8A é um fluxograma ilustrando uma operação exemplar de um codificador de vídeo, de acordo com uma ou mais técnicas desta revelação.

**[0022]** FIG. 8B é um fluxograma ilustrando uma operação exemplar de um codificador de vídeo, de acordo com uma ou mais técnicas desta revelação.

**[0023]** FIG. 9A é um fluxograma ilustrando uma operação exemplar de um decodificador de vídeo, de acordo com uma ou mais técnicas desta revelação.

**[0024]** FIG. 9B é um fluxograma ilustrando uma operação exemplar de um decodificador de vídeo, de acordo com uma ou mais técnicas desta revelação.

**[0025]** FIG. 10A é um fluxograma ilustrando uma operação do codificador de vídeo exemplar para ocultação de

dados de sinal, de acordo com uma ou mais técnicas desta revelação.

**[0026]** FIG. 10B é um fluxograma ilustrando uma operação do decodificador de vídeo exemplar para ocultação de dados de sinal, de acordo com uma ou mais técnicas desta revelação.

#### DESCRIÇÃO DETALHADA

**[0027]** Em geral, esta revelação descreve técnicas para melhorar a intra-previsão na Codificação de Vídeo de Alta Eficiência (HEVC) e outros padrões de codificação de vídeo. A intra-previsão é um processo de geração, baseado nos valores de amostra em uma imagem atual, um bloco preditivo para um bloco de vídeo da imagem atual. Assim, quando um bloco de vídeo de uma imagem atual é codificado usando intra-previsão, o codificador de vídeo não usa valores de amostra a partir de outras imagens para gerar ou de outro modo identificar o bloco preditivo para o bloco de vídeo.

**[0028]** Após gerar um bloco preditivo, um codificador de vídeo pode usar o bloco preditivo para determinar um bloco de amostras residuais (i.e., um bloco residual). As amostras residuais no bloco residual podem indicar a diferença entre as amostras no bloco preditivo e amostras originais correspondentes do bloco de vídeo. O codificador de vídeo pode gerar um bloco de coeficiente de transformada pela aplicação de uma transformada ao bloco residual. A transformada pode converter as amostras residuais de um domínio pixel a um domínio de transformada. O codificador de vídeo pode então quantificar os coeficientes de transformada no bloco de coeficiente de transformada para reduzir as profundidades de bit dos coeficientes de transformada. O codificador de vídeo pode codificar a entropia dos elementos de sintaxe que

representam os coeficientes de transformada quantificados e incluem os elementos de sintaxe codificados por entropia resultantes em um bitstream.

**[0029]** Um decodificador de vídeo pode realizar um inverso deste processo. Ou seja, o decodificador de vídeo pode decodificar por entropia elementos de sintaxe no bitstream para determinar coeficientes de transformada quantificados. O decodificador de vídeo pode então quantificar inversamente os coeficientes de transformada quantificados para determinar os coeficientes de transformada. Além disso, o decodificador de vídeo pode aplicar uma transformada inversa aos coeficientes de transformada para determinar o bloco residual. Além disso, o decodificador de vídeo pode determinar um bloco preditivo (ex., usando intra-previsão). O decodificador de vídeo pode usar amostras no bloco preditivo e amostras residuais correspondentes no bloco residual para reconstruir amostras do bloco de vídeo.

**[0030]** A aplicação da transformada e uso da quantificação causam perda de informação. Assim, as amostras de um bloco de vídeo reconstruído pelo decodificador de vídeo podem não ter o mesmo nível de precisão que as amostras originais do bloco de vídeo. Dessa forma, a aplicação da transformada e uso da quantificação podem ser uma forma de codificação "com perda". Em alguns casos, o codificador de vídeo pode codificar um bloco de vídeo usando a codificação sem perdas. Quando o codificador de vídeo codifica um bloco de vídeo usando a codificação sem perdas, o codificador de vídeo não aplica a transformada às amostras residuais e não quantifica as amostras residuais. Da mesma forma, o decodificador de vídeo não aplica a quantificação inversa ou a transformada inversa. Como um resultado, as amostras

do bloco de vídeo reconstruído pelo decodificador de vídeo podem ter o mesmo nível de precisão que as amostras originais do bloco de vídeo.

**[0031]** Em outros casos, o codificador de vídeo pode realizar um tipo de codificação com perda no qual o codificador de vídeo não aplica uma transformada às amostras residuais, mas quantifica as amostras residuais. Da mesma forma, o decodificador de vídeo pode aplicar a quantificação inversa às amostras residuais, mas não aplica uma transformada inversa às amostras residuais. Porque o codificador de vídeo ainda aplica a quantificação às amostras residuais, as amostras reconstruídas pelo decodificador de vídeo podem ter menos precisão do que as amostras originais, mas a perda de precisão pode potencialmente ser menor do que se a transformada tiver sido aplicada.

**[0032]** Como indicado acima, um codificador de vídeo (ex., um codificador de vídeo ou um decodificador de vídeo) pode usar a intra-previsão para gerar um bloco preditivo. Mais especificamente, o codificador de vídeo usa um modo de intra-previsão específico dentre uma pluralidade de modos de intra-previsão disponíveis para gerar o bloco preditivo. Na HEVC e outros padrões de codificação de vídeo, os modos de intra-previsão incluem uma pluralidade de modos de intra-previsão direcionais, um modo de intra-previsão plano, e um modo de intra-previsão DC. Em geral, quando o codificador de vídeo gera um bloco preditivo usando o modo de intra-previsão plano, as amostras do bloco preditivo podem ser determinadas com base em uma combinação de projeções lineares. Quando o codificador de vídeo gera um bloco preditivo usando o modo de intra-previsão DC, o codificador de vídeo pode determinar um valor de intra-previsão DC. O valor de

intra-previsão DC pode ser um valor médio das amostras adjacentes a uma borda esquerda e uma borda superior do bloco preditivo. O codificador de vídeo pode definir cada valor de amostra no bloco preditivo igual ao valor de intra-previsão DC.

**[0033]** Algumas técnicas desta revelação fornecem melhorias ao modo de intra-previsão DC quando um codificador de vídeo usa a codificação sem perdas. Na codificação sem perdas, um codificador de vídeo pode usar valores de amostras originais quando usando o modo de intra-previsão DC para determinar valores de amostras em um bloco preditivo. Na codificação com perda, um decodificador de vídeo não tem acesso aos valores de amostras originais quando usa a intra-previsão DC para determinar valores de amostras em um bloco preditivo. No entanto, na codificação sem perdas, o decodificador de vídeo tem acesso aos valores de amostra reconstruídos quando usa a intra-previsão DC para determinar valores no bloco preditivo. Na codificação sem perdas, os valores reconstruídos das amostras são os mesmos que os valores originais das amostras.

**[0034]** Como descrito acima, o codificador de vídeo pode gerar um bloco preditivo. Como parte da geração do bloco preditivo, o codificador de vídeo pode usar pelo menos uma de uma amostra reconstruída sem perdas à esquerda de uma amostra atual em uma fileira atual de um bloco preditivo e uma amostra reconstruída sem perdas para uma fileira do bloco preditivo acima da fileira atual para a previsão DC da amostra atual. Além disso, em alguns casos, isso pode permitir que o decodificador de vídeo canalize a determinação dos valores de amostra no bloco preditivo.

**[0035]** Além disso, como indicado acima, um codificador de vídeo pode realizar uma forma de codificação

com perda na qual a quantificação é usada, mas a transformada é pulada, a qual pode ser referida como codificação de pulo de transformada. De acordo com uma ou mais técnicas adicionais desta revelação, o codificador de vídeo pode aplicar uma forma de modulação de código de pulso diferencial residual (DPCM) para preparar as amostras residuais não-transformadas, mas quantificadas, para codificação. Esta forma de DPCM residual é descrita em detalhe em outro lugar nesta revelação. Em contraste com outras propostas para usar DPCM na intra-codificação com perda, esta forma de DPCM residual descrito nesta revelação pode aumentar a capacidade de transmissão do codificador de vídeo e/ou decodificador de vídeo.

**[0036]** Como indicado acima, um codificador de vídeo pode codificar a entropia elementos de sintaxe que representam coeficientes de transformada quantificados. Na codificação sem perdas, ou codificação com perda quando a transformada é pulada, alguns elementos de sintaxe podem ser usados para representar amostras residuais. Em HEVC e outros padrões de codificação de vídeo, os elementos de sintaxe que representam um coeficiente de transformada ou amostra residual pode incluir um elemento de sintaxe de sinal que indica se o coeficiente de transformada ou amostra residual é positivo ou negativo. Em alguns casos, pode ser desnecessário incluir elementos de sintaxe de sinal para indicar se um coeficiente de transformada ou amostra residual é positivo ou negativo. Ao invés, a informação que indica se um coeficiente de transformada ou amostra residual é positivo ou negativo pode ser incorporada nos valores de outros elementos de sintaxe para o coeficiente de transformada ou amostra residual. A incorporação de tal informação nos valores de outros elementos de sintaxe, ao invés da sinalização de elementos



de sintaxe de sinal pode ser referida como ocultação de dados de sinal.

**[0037]** No entanto, a ocultação de dados de sinal pode ser difícil de implementar para blocos que são codificados usando codificação com perda para a qual a transformada é pulada e um modo de intra-previsão plano, um modo de intra-previsão DC (ex., um modo de intra-previsão DC no qual as amostras reconstruídas correspondentes às amostras no bloco preditivo são usadas para determinar o valor das amostras preditivas no bloco preditivo), ou DPCM residual é usada. Além disso, na codificação de pulo de transformada, a ocultação de dados de sinal pode introduzir erros nos valores residuais que são compostos quando a DPCM residual é aplicada. Tais erros podem se propagar para amostras residuais subsequentes, resultando em uma degradação de desempenho. Então, de acordo com uma ou mais técnicas desta revelação, a ocultação de dados de sinal pode ser normativamente desabilitada para tais blocos mesmo se um ou mais elementos de sintaxe indicam que a ocultação de dados de sinal é desabilitada para tais blocos.

**[0038]** Por exemplo, em alguns exemplos, o decodificador de vídeo determina que a ocultação de dados de sinal é desabilitada para um bloco atual se o bloco atual é gerado usando a codificação com perda sem a aplicação de uma transformada aos dados residuais e o bloco atual é intra previsto usando um modo de intra-previsão no qual a DPCM residual é usada. Em tais exemplos, quando a ocultação de dados de sinal é desabilitada para o bloco atual, o decodificador de vídeo pode obter, a partir do bitstream, para cada respectivo valor significativo no bloco, um respectivo elemento de sintaxe indicando se o respectivo valor significativo é positivo ou negativo.

**[0039]** FIG. 1 é um diagrama em bloco ilustrando um sistema de codificação de vídeo exemplar 10 que pode usar as técnicas dessa revelação. Como descrito acima, o termo "codificador de vídeo" refere-se genericamente a ambos codificadores de vídeo e decodificadores de vídeo. Nesta revelação, os termos "codificação de vídeo" ou "codificação" podem se referir genericamente à codificação de vídeo ou decodificação de vídeo.

**[0040]** Como mostrado na FIG. 1, o sistema de codificação de vídeo 10 inclui um dispositivo fonte 12 e um dispositivo de destino 14. O dispositivo fonte 12 gera dados de vídeo codificados. Dessa forma, o dispositivo fonte 12 pode ser referido como um dispositivo de codificação de vídeo ou um aparelho de codificação de vídeo. O dispositivo de destino 14 pode decodificar os dados de vídeo codificados gerados pelo dispositivo fonte 12. Dessa forma, o dispositivo de destino 14 pode ser referido como um dispositivo de decodificação de vídeo ou um aparelho de decodificação de vídeo. O dispositivo fonte 12 e dispositivo de destino 14 podem ser exemplos de dispositivos de codificação de vídeo ou aparelhos de codificação de vídeo.

**[0041]** O dispositivo fonte 12 e dispositivo de destino 14 podem compreender uma ampla gama de dispositivos, incluindo computadores desktop, dispositivos de computação móvel, computadores notebook (ex., portátil), computadores tablet, decodificadores, aparelhos de telefone, tais como os chamados telefones "inteligentes", televisores, câmeras, dispositivos de exibição, reprodutores de mídia digitais, consoles de vídeo games, computadores de bordo, ou semelhantes.

**[0042]** O dispositivo de destino 14 pode receber dados de vídeo codificados do dispositivo fonte 12 através de um canal 16. O canal 16 pode compreender uma ou mais mídias ou dispositivos capazes de mover os dados de vídeo codificados do dispositivo fonte 12 para o dispositivo de destino 14. Em um exemplo, o canal 16 pode compreender um ou mais meio de comunicação que habilita o dispositivo fonte 12 a transmitir dados de vídeo codificados diretamente para o dispositivo de destino 14 em tempo real. Neste exemplo, o dispositivo fonte 12 pode modular os dados de vídeo codificados de acordo com um padrão de comunicação, como um protocolo de comunicação remoto, e pode transmitir os dados de vídeo modulados para o dispositivo de destino 14. Os um ou mais meios de comunicação podem incluir meios de comunicação sem fio e/ou com fio, como um espectro de radiofrequência (RF) ou uma ou mais linhas de transmissão físicas. Os um ou mais meios de comunicação podem fazer parte de uma rede baseada em pacote, como uma rede de área local, uma rede de área ampla, ou uma rede global (ex., a Internet). O canal 16 pode incluir vários tipos de dispositivos, como roteadores, comutadores, estações base, ou outro equipamento que facilita a comunicação do dispositivo fonte 12 para o dispositivo de destino 14.

**[0043]** Em outro exemplo, o canal 16 pode incluir uma mídia de armazenamento que armazena dados de vídeo codificados gerados pelo dispositivo fonte 12. Neste exemplo, o dispositivo de destino 14 pode acessar a mídia de armazenamento, ex., através de acesso com disco ou acesso com cartão. A mídia de armazenamento pode incluir uma variedade de mídias de armazenamento de dados acessadas localmente como discos Blu-ray, DVDs, CD-ROMs, memória

flash, ou outra mídia de armazenamento digital adequada para armazenar dados de vídeo codificados.

**[0044]** Em um exemplo adicional, o canal 16 pode incluir um servidor de arquivo ou outro dispositivo de armazenamento intermediário que armazena dados de vídeo codificados gerados pelo dispositivo fonte 12. Neste exemplo, o dispositivo de destino 14 pode acessar dados de vídeo codificados armazenados no servidor de arquivo ou outro dispositivo de armazenamento intermediário através de streaming ou download. O servidor de arquivo pode ser um tipo de servidor capaz de armazenar dados de vídeo codificados e transmitir os dados de vídeo codificados para o dispositivo de destino 14. Servidores de arquivo exemplares incluem servidores da web (ex., para um site da web), servidores de protocolo de transferência de arquivo (FTP), dispositivos de armazenamento ligados à rede (NAS), unidades de disco locais, e similares.

**[0045]** O dispositivo de destino 14 pode acessar os dados de vídeo codificados através de uma conexão de dados padrão, como uma conexão da Internet. Tipos de conexões de dados exemplares podem incluir canais sem fio (ex., conexões Wi-Fi), conexões com fio (ex., DSL, modem a cabo, etc.), ou combinações de ambas que são adequadas para acessar os dados de vídeo codificados armazenados em um servidor de arquivo. A transmissão de dados de vídeo codificados a partir do servidor de arquivo pode ser uma transmissão por streaming, uma transmissão por download, ou uma combinação de ambas.

**[0046]** As técnicas desta revelação não são limitadas às aplicações ou ajustes sem fio. As técnicas podem ser aplicadas à codificação de vídeo em suporte de uma variedade de aplicações multimídia, como transmissões de televisão pelo ar, transmissões de televisão por cabo,

transmissões de televisão por satélite, streaming de transmissões de vídeo, ex., através da Internet, codificação dos dados de vídeo para armazenamento em uma mídia de armazenamento de dados, decodificação dos dados de vídeo armazenados em uma mídia de armazenamento de dados, ou outras aplicações. Em alguns exemplos, o sistema de codificação de vídeo 10 pode ser configurado para suportar a transmissão de vídeo de uma via ou de duas vias para suportar aplicações como streaming de vídeo, reprodução de vídeo, transmissão de vídeo, e/ou telefonia por vídeo.

**[0047]** No exemplo da FIG. 1, o dispositivo fonte 12 inclui uma fonte de vídeo 18, um codificador de vídeo 20, e uma interface de saída 22. Em alguns exemplos, a interface de saída 22 pode incluir um modulador/demodulador (modem) e/ou um transmissor. A fonte de vídeo 18 pode incluir um dispositivo de captura de vídeo, ex., uma câmera de vídeo, um arquivo de vídeo contendo dados de vídeo anteriormente capturados, uma interface de feed de vídeo para receber dados de vídeo de um provedor de conteúdo de vídeo, e/ou um sistema de gráfico de computador para gerar dados de vídeo, ou uma combinação de tais fontes dos dados de vídeo.

**[0048]** O codificador de vídeo 20 pode codificar dados de vídeo da fonte de vídeo 18. Em alguns exemplos, o dispositivo fonte 12 transmite diretamente os dados de vídeo codificados para o dispositivo de destino 14 através da interface de saída 22. Em outros exemplos, os dados de vídeo codificados também podem ser armazenados em uma mídia de armazenamento ou um servidor de arquivo para acesso posterior pelo dispositivo de destino 14 para decodificação e/ou playback.

**[0049]** No exemplo da FIG. 1, o dispositivo de destino 14 inclui uma interface de entrada 28, um

decodificador de vídeo 30, e um dispositivo de exibição 32. Em alguns exemplos, a interface de entrada 28 inclui um receptor e/ou a modem. A interface de entrada 28 pode receber dados de vídeo codificados sobre o canal 16. O dispositivo de exibição 32 pode ser integrado com ou pode ser externo ao dispositivo de destino 14. Em geral, o dispositivo de exibição 32 exibe os dados de vídeo decodificados. O dispositivo de exibição 32 pode compreender uma variedade de dispositivos de exibição, como um monitor de cristal líquido (LCD), um monitor de plasma, um monitor de diodo emissor de luz orgânico (OLED, ou outro tipo de dispositivo de exibição.

**[0050]** FIG. 1 é meramente um exemplo e as técnicas desta revelação podem se aplicar aos ajustes de codificação de vídeo (ex., codificação de vídeo ou decodificação de vídeo) que não necessariamente incluem qualquer comunicação de dados entre o dispositivo de codificação de vídeo e o dispositivo de decodificação de vídeo. Em outros exemplos, os dados são recuperados de uma memória local, com fluxo contínuo através de uma rede, ou similares. Um dispositivo de codificação de vídeo pode codificar e armazenar dados para a memória, e/ou um dispositivo de decodificação de vídeo pode recuperar e decodificar dados a partir da memória. Em muitos exemplos, a codificação de vídeo e decodificação é realizada pelos dispositivos que não se comunicam um com o outro, mas simplesmente codifica dados para a memória e/ou recupera e decodifica dados a partir da memória.

**[0051]** O codificador de vídeo 20 e decodificador de vídeo 30 cada podem ser implementados como qualquer de uma variedade de conjunto de circuitos adequado, como um ou mais microprocessadores, processadores de sinal digital (DSPs), circuitos integrados de aplicação

específica (ASICs), arranjos de porta programáveis em campo (FPGAs), lógica discreta, hardware, ou qualquer combinação dos mesmos. Se as técnicas são implementadas parcialmente em software, um dispositivo pode armazenar instruções para o software em uma mídia de armazenamento legível por computador não transitória, adequada e pode executar as instruções em hardware usando um ou mais processadores para realizar as técnicas desta revelação. Qualquer um dos anteriores (incluindo hardware, software, uma combinação de hardware e software, etc.) pode ser considerado como um ou mais processadores. Cada um do codificador de vídeo 20 e decodificador de vídeo 30 pode ser incluído em um ou mais codificadores ou decodificadores, qualquer um dos quais pode ser integrado como parte de um codificador/decodificador combinado (CODEC) em um respectivo dispositivo.

**[0052]** Esta revelação pode geralmente se referir ao codificador de vídeo 20 "que sinaliza" determinada informação. O termo "que sinaliza" pode geralmente se referir à comunicação dos elementos de sintaxe e/ou outros dados usados para decodificar os dados de vídeo comprimidos. Tal comunicação pode ocorrer em tempo real ou quase real. Alternativamente, tal comunicação pode ocorrer ao longo de uma duração de tempo, como poderá ocorrer quando se armazenam elementos de sintaxe em uma mídia de armazenamento legível por computador em um bitstream codificado no momento da codificação, tal dispositivo de decodificação de vídeo pode então se recuperar em qualquer momento sendo armazenado nesta mídia.

**[0053]** Em alguns exemplos, o codificador de vídeo 20 e decodificador de vídeo 30 operam de acordo com um padrão de compressão de vídeo, como o padrão de

Codificação de Vídeo de Alta Eficiência (HEVC). Um esboço do padrão HEVC, referido como "HEVC Working Draft 6," é descrito em Bross et al., "High Efficiency Video Coding (HEVC) text specification draft 6," Equipe Colaborativa Conjunta em Codificação de vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 7th Meeting: Geneva, Switzerland, Novembro, 2011, o conteúdo total do qual é incorporado aqui a título de referência. A partir de 5 de abril 2013, HEVC Working Draft 6 pode ser baixado em [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/8\\_San%20Jose/wg11/JCTVC-H1003-v22.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/8_San%20Jose/wg11/JCTVC-H1003-v22.zip). Outro esboço do padrão HEVC, referido como "HEVC Working Draft 9," é descrito em Bross et al., "High Efficiency Video Coding (HEVC) text specification draft 9," Equipe Colaborativa Conjunta em Codificação de vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 11th Meeting: Shanghai, China, Outubro, 2012, o conteúdo total do qual é incorporado aqui a título de referência. A partir de 24 de março de 2014, HEVC Working Draft 9 pode ser baixado em [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/11\\_Shanghai/wg11/JCTVC-K1003-v8.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v8.zip). Outro rascunho de HEVC, referido como "HEVC Working Draft 10," é descrito em Bross et al., "High Efficiency Video Coding (HEVC) text specification draft 10 (para FDIS & Consent)," Equipe Colaborativa Conjunta em Codificação de vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 12th Meeting: Geneva, CH, 14 a 23 de Janeiro de 2013, o conteúdo total do qual é incorporado aqui a título de referência. Desde 24 de março de 2014, HEVC Working Draft 10 está disponível em [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/12\\_Geneva/wg11/JCTVC-L1003-v20.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v20.zip). As técnicas desta revelação, no entanto, não são limitadas a qualquer padrão ou técnica de codificação particular.



**[0054]** Além disso, uma especificação de extensão de faixa está sendo desenvolvida para HEVC. A especificação de extensão de faixa fornece modos de amostragem alternativos, como amostragem croma 4:0:0, 4:2:0, 4:2:2, e 4:4:4. Flynn et al., "High Efficiency Video Coding (HEVC) Range Extensions text specification: Draft 3," Equipe Colaborativa Conjunta em Codificação de vídeo (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 13th Meeting: Incheon, KR, 18-26 de Abril de 2013, (doravante, "JCTVC-M1005\_v2") é um esboço do especificação de extensão de faixa for HEVC. A partir de 24 de Março de 2014, JCTVC-M1005\_v2 está disponível em [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/13\\_Incheon/pending/JCTVC-M1005-v2.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/13_Incheon/pending/JCTVC-M1005-v2.zip). O conteúdo total de JCTVC-M1005\_v2 é incorporado aqui a título de referência.

**[0055]** Como mencionado brevemente acima, o codificador de vídeo 20 codifica dados de vídeo. Os dados de vídeo podem compreender uma ou mais imagens. Cada uma das imagens é uma imagem estática que faz parte de um vídeo. Quando o codificador de vídeo 20 codifica os dados de vídeo, o codificador de vídeo 20 pode gerar um bitstream. O bitstream pode incluir uma sequência de bits que forma uma representação codificada dos dados de vídeo. O bitstream pode incluir imagens codificadas e dados associados. Uma imagem codificada é a representação codificada de uma imagem. Os dados associados podem incluir conjuntos de parâmetro de sequência (SPSs), conjuntos de parâmetro de imagem (PPSs), e outras estruturas de sintaxe. Um SPS pode conter parâmetros aplicáveis a zero ou mais sequências de imagens. Um PPS pode conter parâmetros aplicáveis a zero ou mais imagens.

**[0056]** Uma imagem pode incluir três matrizes de amostra, denotadas  $S_L$ ,  $S_{Cb}$  e  $S_{Cr}$ .  $S_L$  é uma matriz

bidimensional (i.e., um bloco) de amostras luma. As amostras luma também podem ser referidas aqui como amostras "Y".  $S_{Cb}$  é uma matriz bidimensional de Cb amostras de crominância.  $S_{Cr}$  é uma matriz bidimensional de amostras de crominância Cr. As amostras de crominância também podem ser referidas aqui como amostras "croma". As amostras de crominância Cb podem ser referidas aqui como "amostras U." As amostras de crominância Cr podem ser referidas aqui como "amostras V."

**[0057]** Em alguns exemplos, o codificador de vídeo 20 pode sub-amostrar as matrizes croma de uma imagem (i.e.,  $S_{Cb}$  e  $S_{Cr}$ ). Por exemplo, o codificador de vídeo 20 pode usar um formato de vídeo YUV 4:2:0, um formato de vídeo YUV 4:2:2, ou um formato de vídeo 4:4:4. No formato de vídeo YUV 4:2:0, o codificador de vídeo 20 pode sub-amostrar as matrizes croma de modo que as matrizes croma são  $\frac{1}{2}$  da altura e  $\frac{1}{2}$  da largura da matriz luma. No formato de vídeo YUV 4:2:2, o codificador de vídeo 20 pode sub-amostrar as matrizes croma de modo que as matrizes croma sejam  $\frac{1}{2}$  da largura e tenham a mesma altura que a matriz luma. No formato de vídeo YUV 4:4:4, o codificador de vídeo 20 não sub-amostra as matrizes croma.

**[0058]** Para gerar uma representação codificada de uma imagem, o codificador de vídeo 20 pode gerar um conjunto de unidades de árvore de codificação (CTUs). Cada uma das CTUs pode ser um bloco de árvore de codificação de amostras luma, dois correspondendo aos blocos de árvore de codificação das amostras croma, e estruturas de sintaxe usadas para codificar as amostras dos blocos de árvore de codificação. Um bloco de árvore de codificação pode ser um bloco de amostras  $N \times N$ . Uma CTU também pode ser referida como um "bloco de árvore" ou uma "unidade de codificação maior" (LCU). As CTUs da HEVC podem ser amplamente

análogas aos macroblocos de outros padrões, como H.264/AVC. No entanto, uma CTU não é necessariamente limitada a um tamanho particular e pode incluir uma ou mais unidades de codificação (CUs).

**[0059]** Como parte da codificação de uma imagem, o codificador de vídeo 20 pode gerar representações codificadas de cada fatia da imagem (i.e., fatias codificadas). Para gerar uma fatia codificada, o codificador de vídeo 20 pode codificar uma série de CTUs. Esta revelação pode se referir a uma representação codificada de uma CTU como uma CTU codificada. Em alguns exemplos, cada uma das fatias inclui um número inteiro de CTUs codificadas.

**[0060]** Para gerar uma CTU codificada, o codificador de vídeo 20 pode recursivamente realizar a partição de quad-tree nos blocos de árvore de codificação de uma CTU para dividir os blocos de árvore de codificação em blocos de codificação, consequentemente o nome "unidades de árvore de codificação." Um bloco de codificação é um bloco de amostras  $N \times N$ . Uma CU pode ser um bloco de codificação de amostras luma e dois correspondendo aos blocos de codificação das amostras croma de uma imagem que tem uma matriz de amostra luma, uma matriz de amostra Cb e uma matriz de amostra Cr, e estruturas de sintaxe usadas para codificar as amostras dos blocos de codificação. Nas imagens monocromáticas, ou imagens codificadas usando planos de cor separados, uma CU pode compreender um bloco de codificação único de amostras e estruturas de sintaxe usadas para codificar os blocos de codificação. O codificador de vídeo 20 pode dividir um bloco de codificação de uma CU em um ou mais blocos de previsão. Um bloco de previsão pode ser um bloco de amostras retangular (i.e., quadrado ou não-quadrado) no qual a mesma previsão é

aplicada. Uma unidade de previsão (PU) de uma CU pode ser um bloco de previsão de amostras luma, dois correspondendo aos blocos de previsão das amostras croma de uma imagem, e estruturas de sintaxe usadas para prever as amostras do bloco de previsão. O codificador de vídeo 20 pode gerar blocos luma, Cb e Cr preditivos para blocos de previsão luma, Cb e Cr de cada PU da CU. Nas imagens monocromáticas, ou imagens codificadas usando planos de cor separados, uma CU pode compreender um bloco de codificação único de amostras e estruturas de sintaxe usadas para codificar os blocos de codificação.

**[0061]** O codificador de vídeo 20 pode usar a intra-previsão ou inter previsão para gerar os blocos preditivos para uma PU. Se o codificador de vídeo 20 usa a intra-previsão para gerar os blocos preditivos de uma PU, o codificador de vídeo 20 pode gerar os blocos preditivos da PU com base nas amostras decodificadas da imagem associada com a PU.

**[0062]** Se o codificador de vídeo 20 usa a inter previsão para gerar os blocos preditivos de uma PU, o codificador de vídeo 20 pode gerar os blocos preditivos da PU baseado nas amostras decodificadas de uma ou mais imagens que não a imagem associada com a PU. O codificador de vídeo 20 pode usar uni-previsão ou bi-previsão para gerar os blocos preditivos de uma PU. Quando o codificador de vídeo 20 usa a uni-previsão para gerar os blocos preditivos para uma PU, a PU pode ter um vetor de movimento único. Quando o codificador de vídeo 20 usa a uni-previsão para gerar os blocos preditivos para uma PU, a PU pode ter dois vetores de movimento.

**[0063]** Após o codificador de vídeo 20 gerar bloco preditivos (ex., blocos luma, Cb e Cr) para uma ou mais PUs de uma CU, o codificador de vídeo 20 pode gerar um

bloco residual para a CU. Por exemplo, o codificador de vídeo 20 pode gerar um bloco residual luma para a CU. Cada amostra no bloco residual luma da CU indica uma diferença entre uma amostra luma em um dos blocos luma preditivos da CU e a amostra correspondente no bloco de codificação luma original da CU. Além disso, o codificador de vídeo 20 pode gerar um bloco residual Cb para a CU. Cada amostra no bloco residual Cb da CU pode indicar uma diferença entre uma amostra Cb em um dos blocos Cb preditivos da CU e a amostra correspondente no bloco de codificação Cb original da CU. O codificador de vídeo 20 também pode gerar um bloco residual Cr para a CU. Cada amostra no bloco residual Cr da CU pode indicar uma diferença entre uma amostra Cr em um dos blocos Cr preditivos da CU e a amostra correspondente no bloco de codificação Cr original da CU.

**[0064]** Além disso, o codificador de vídeo 20 pode usar a partição de quad-tree para decompor os blocos residuais de uma CU nos blocos de transformada. Por exemplo, o codificador de vídeo 20 pode usar a partição de quad-tree para decompor bloco residuais luma, Cb, e Cr de uma CU em blocos de transformada luma, Cb, e Cr. Um bloco de transformada pode ser um bloco de amostras retangular no qual a mesma transformada é aplicada. Uma unidade de transformada (TU) de uma CU pode ser um bloco de transformada de amostras luma, dois correspondendo aos blocos de transformada das amostras croma, e estruturas de sintaxe usadas para transformar as amostras do bloco de transformada. Assim, cada TU de uma CU pode ser associada com um bloco de transformada luma, um bloco de transformada Cb, e um bloco de transformada Cr. O bloco de transformada luma associado com a TU pode ser um sub-bloco do bloco residual luma da CU. O bloco de transformada Cb pode ser um sub-bloco do bloco residual Cb da CU. O bloco de

transformada Cr pode ser um sub-bloco do bloco residual Cr da CU. Nas imagens monocrômicas, ou imagens codificadas usando planos de cor separados, uma TU pode compreender um bloco de transformada único e estruturas de sintaxe usadas para transformar as amostras do bloco de transformada. Um tamanho de TU pode ser o tamanho de um bloco de transformada de uma TU.

**[0065]** O codificador de vídeo 20 pode aplicar uma ou mais transformadas a um bloco de transformada de uma TU para gerar um bloco de coeficiente para a TU. Por exemplo, o codificador de vídeo 20 pode aplicar uma ou mais transformadas a um bloco de transformada luma de uma TU para gerar um bloco de coeficiente luma para a TU. Um bloco de coeficiente pode ser uma matriz bidimensional de coeficientes de transformada. Um coeficiente de transformada pode ser uma quantidade escalar. Além disso, o codificador de vídeo 20 pode aplicar uma ou mais transformadas a um bloco de transformada Cb de uma TU para gerar um bloco de coeficiente Cb para a TU. O codificador de vídeo 20 pode aplicar uma ou mais transformadas a um bloco de transformada Cr de uma TU para gerar um bloco de coeficiente Cr para a TU.

**[0066]** Após gerar um bloco de coeficiente (ex., um bloco de coeficiente luma, um bloco de coeficiente Cb ou um bloco de coeficiente Cr), o codificador de vídeo 20 pode quantificar o bloco de coeficiente. A quantificação geralmente refere-se a um processo no qual os coeficientes de transformada são quantificados para reduzir possivelmente a quantidade de dados usada para representar os coeficientes de transformada, proporcionando compressão adicional. Após o codificador de vídeo 20 quantificar um bloco de coeficiente, o codificador de vídeo 20 pode codificar a entropia dos elementos de sintaxe que indicam

os coeficientes de transformada quantificados. Por exemplo, o codificador de vídeo 20 pode realizar a Codificação Aritmética Binária Adaptativo ao Contexto (CABAC) nos elementos de sintaxe que indicam os coeficientes de transformada quantificados. O codificador de vídeo 20 pode produzir os elementos de sintaxe com entropia codificada em um bitstream.

**[0067]** O decodificador de vídeo 30 pode receber um bitstream gerado pelo codificador de vídeo 20. Além disso, o decodificador de vídeo 30 pode analisar o bitstream para decodificar elementos de sintaxe a partir do bitstream. O decodificador de vídeo 30 pode reconstruir as imagens dos dados de vídeo com base pelo menos em parte nos elementos de sintaxe decodificados a partir do bitstream. O processo para reconstruir os dados de vídeo pode ser geralmente recíproco ao processo realizado pelo codificador de vídeo 20. Por exemplo, o decodificador de vídeo 30 pode usar vetores de movimento das PUs para determinar os blocos preditivos para as PUs de uma CU atual. Além disso, o decodificador de vídeo 30 pode quantificar inversamente o bloco de coeficientes de transformada associado com as TUs da CU atual. O decodificador de vídeo 30 pode realizar transformadas inversas no bloco de coeficientes de transformada para reconstruir os blocos de transformada associada com as TUs da CU atual. O decodificador de vídeo 30 pode reconstruir os blocos de codificação da CU atual pela adição das amostras dos blocos preditivos para as PUs da CU atual para as amostras correspondentes de um bloco de transformada das TUs da CU atual. Pela reconstrução dos blocos de codificação para cada CU de uma imagem, o decodificador de vídeo 30 pode reconstruir a imagem.

**[0068]** Como indicado acima, um codificador de vídeo, como codificador de vídeo 20 ou decodificador de

vídeo 30, pode usar a intra-previsão para gerar um bloco preditivo para uma PU atual. Quando um codificador de vídeo usa a intra-previsão para gerar um bloco preditivo para uma PU atual, o codificador de vídeo pode determinar valores de amostras no bloco preditivo usando um conjunto de amostras de referência. Por exemplo, na intra-previsão da HEVC, as amostras já reconstruídas a partir dos blocos vizinhos do lado superior e esquerdo podem ser usadas para previsão. Estas amostras reconstruídas podem ser referidas como amostras de referência.

**[0069]** FIG. 2 ilustra amostras de referência de um bloco para a intra-previsão HEVC. Em outras palavras, a FIG. 2 é um diagrama conceitual que ilustra um bloco de tamanho  $M$  (altura)  $\times$   $N$  (largura). Na FIG. 2,  $M$  indica fileiras e  $N$  indica colunas. Além disso, na FIG. 2, as amostras de um bloco são denotadas por  $P_{i,j}$ , onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ . Nesta revelação, o termo "amostras" pode se referir aos valores de pixel originais de um componente de entrada (ex.,  $R$ ,  $G$ , ou  $B$  em uma imagem RGB,  $Y$ ,  $Cb$ , ou  $Cr$ , em imagens YCbCr, etc.) ou valores de amostra de um componente após a aplicação de uma transformada de cor aos componentes de entrada. No exemplo da FIG. 2, os pixels de referência são denotados por  $P_{-1,j}$ , onde  $-1 \leq j < 2N$  e  $P_{i,-1}$ , onde  $-1 \leq i < 2M$ .

**[0070]** Como mostrado no exemplo da FIG. 2, as amostras de referência podem incluir um conjunto de amostras de referência da esquerda da PU atual e um conjunto de amostra de referência acima da PU atual. Esta revelação pode se referir ao conjunto de amostras de referência acima da PU atual como o previsor superior. Esta revelação pode se referir ao conjunto de amostras de referência da esquerda da PU atual como o previsor da esquerda. Em outras palavras, na intra-previsão da HEVC,



as amostras já reconstruídas a partir dos blocos vizinhos do lado superior e esquerdo são usadas para previsão (o bloco vizinho "superior" também pode ser chamado de bloco vizinho "acima"). Estas amostras são referidas como amostras de referência. Em alguns exemplos, se os pixels de referência não estão disponíveis, um codificador de vídeo usando HEVC pode usar um processo de preenchimento específico para gerar amostras de referência faltantes.

**[0071]** Quando um codificador de vídeo usa a intra-previsão para gerar um bloco preditivo, o codificador de vídeo pode gerar o bloco preditivo de acordo com um modo de intra-previsão a partir de uma pluralidade de modos de intra-previsão disponíveis. Os modos de intra-previsão podem incluir uma pluralidade de modos de intra-previsão direcionais (i.e., angular). Por exemplo, em algumas versões de HEVC, há 33 modos de intra-previsão direcionais. Cada um dos modos de intra-previsão direcionais corresponde a uma direção diferente. A FIG. 3 é um diagrama conceitual que ilustra direções do modo de intra-previsão exemplar. Quando o codificador de vídeo gera um bloco preditivo de acordo com um modo de intra-previsão direcional, o codificador de vídeo pode, para cada respectiva amostra do bloco preditivo, atribuir à respectiva amostra um valor de uma amostra de referência (ou uma combinação ponderada de amostras de referência) que é alinhada com a respectiva amostra em uma direção correspondente ao modo de intra-previsão direcional. Quando um codificador de vídeo usa um modo de intra-previsão direcional (i.e., angular) para gerar um bloco preditivo para um bloco atual, o codificador de vídeo pode ser dito por realizar a intra-previsão angular.

**[0072]** Além disso, em algumas versões de HEVC, os modos de intra-previsão incluem um modo de intra-

previsão DC. Em tais versões de HEVC, quando o codificador de vídeo usa a intra-previsão DC para gerar um bloco preditivo, o codificador de vídeo pode determinar um valor médio das amostras de referência. O codificador de vídeo pode então determinar que cada amostra no bloco preditivo tem o valor médio determinado. Assim, quando o codificador de vídeo usa a intra-previsão DC para gerar um bloco preditivo, todas as amostras do bloco preditivo têm o mesmo valor. Por exemplo, assumem que um processo de preenchimento foi concluído de modo que todas as amostras de referência estejam disponíveis. Neste exemplo, com relação ao bloco 4×4 mostrado no exemplo da FIG. 2, a previsão DC pode ser formada como:

$$(\sum_{i=0}^3 P_{i,-1} + \sum_{j=0}^3 P_{-1,j} + 4) \gg 3, \quad (1)$$

onde  $\gg$  denota uma operação de deslocamento para a direita bit por bit.

**[0073]** Em algumas versões de HEVC, os modos de intra-previsão incluem um modo de intra-previsão plano. Quando um codificador de vídeo gera um bloco preditivo para uma PU usando o modo de intra-previsão plano, o codificador de vídeo pode determinar um conjunto de amostras vizinhas,  $p[x][y]$ , com  $x = -1$ ,  $y = -1..nTbS*2-1$  e  $x = 0..nTbS*2-1$ ,  $y = -1$ , onde  $nTbS$  é o tamanho da PU atual. Além disso,  $predSamples[x][y]$  podem denotar o valor de uma amostra na posição  $x$ ,  $y$  do bloco de previsão. O codificador de vídeo pode determinar as amostra partir do bloco preditivo como segue:

$$\begin{aligned} predSamples[x][y] = & ( ( nTbS - 1 - x ) * p[-1][y] \\ + & \\ & (x + 1) * p[nTbS][-1] + \end{aligned} \quad (2)$$

$$\begin{aligned} & (nTbs - 1 - y) * p[x][-1] + \\ & (y + 1) * p[-1][nTbS] + nTbS) >> ( \text{Log2}(nTbS) \\ & + 1) \end{aligned}$$

Em termos gerais, quando o modo de intra-previsão plano é usado, o valor de uma amostra do bloco preditivo é um média de duas interpolações lineares do valor. Na primeira interpolação linear, na medida em que os valores de  $x$  aumentam da esquerda para a direita através de uma fileira do bloco preditivo, um peso concedido a uma amostra de referência esquerda da fileira diminui enquanto um peso concedido a uma amostra de referência acima e direita de um canto superior direito do bloco preditivo aumenta. Na segunda interpolação linear, na medida em que os valores de  $y$  aumenta abaixo de uma coluna do bloco preditivo, um peso concedido a uma amostra de referência acima da coluna diminui enquanto um peso concedido a uma amostra abaixo e esquerda de um canto esquerdo inferior do bloco preditivo aumenta.

**[0074]** No exemplo da FIG. 2, a intra-previsão plana pode usar amostras  $P_{-1,j}$ , onde  $0 \leq j \leq (N - 1)$ , e  $P_{M,-1}$  para gerar uma previsão bi-linear na direção vertical. De modo similar, as amostras  $P_{i,-1}$ , onde  $0 \leq i \leq (M - 1)$ , e  $P_{1,N}$  podem ser usados para gerar uma previsão bi-linear na direção horizontal. Finalmente, neste exemplo, as previsões horizontais e verticais podem ter as médias calculadas (ou possivelmente combinadas com outra operação matemática). Por exemplo, deixe os valores previstos planos serem denotados por  $T_{i,j}$  e assuma que o bloco é quadrado, isto é,  $M = N$ . Neste exemplo,  $T_{i,j}$  pode ser determinado como:

$$\begin{aligned}
T_{i,j}^V &= (M - i) * P_{-1,j} + i * P_{M,-1}, \\
T_{i,j}^H &= (N - j) * P_{i,-1} + j * P_{-1,N} \text{ and} \\
T_{i,j} &= (T_{i,j}^V + T_{i,j}^H + N) \gg (\log_2 N + 1).
\end{aligned} \tag{3}$$

Neste exemplo,  $*$  denota produto,  $\gg$  denota um operação de deslocamento para a direita bit por bit e os sobrescritos H e V denotam previsões nas direções horizontais e verticais, respectivamente.

**[0075]** Em alguns casos, o codificador de vídeo 20 e decodificador de vídeo 30 implementam um modo de codificação sem perdas como descrito acima. Tipicamente, quando o codificador de vídeo 20 codifica um bloco, o codificador de vídeo 20 transforma (ex., usando uma transformada de cosseno discreta) e quantifica dados residuais (i.e., erro de previsão) para o bloco. Em outras palavras, o erro de previsão é transformado e quantificado. No entanto, quando o codificador de vídeo 20 codifica um bloco (ex., a CU) usando a codificação sem perdas, o codificador de vídeo 20 pode não aplicar uma transformada ou quantificação aos dados residuais para o bloco. Em outras palavras, no modo de codificação sem perdas (ex., para uma CU ou uma imagem inteira), as etapas de transformada e quantificação podem ser puladas. Ao invés, o codificador de vídeo 20 pode tratar os valores de amostra dos dados residuais da mesma maneira como coeficientes de transformada quantificados. Por exemplo, o codificador de vídeo 20 pode codificar a entropia de elementos de sintaxe que representam valores de amostra dos dados residuais e incluem os dados resultantes em um bitstream. Assim, os dados residuais não sofrem qualquer perda de informação devido a transformação ou quantificação.

**[0076]** De modo similar, em alguns casos onde o decodificador de vídeo 30 decodifica um bloco (ex., uma CU)

que foi codificado usando codificação sem perdas, o decodificador de vídeo 30 pode não aplicar a quantificação inversa ou transformadas inversas aos dados residuais para o bloco. Ao invés, o decodificador de vídeo 30 pode decodificar por entropia os elementos de sintaxe que representam valores de amostra dos dados residuais e então reconstruir valores de amostra do bloco com base pelo menos em parte nos valores de amostra dos dados residuais.

**[0077]** Algumas técnicas exemplares desta revelação referem-se à codificação sem perdas. Como descrito acima, ao invés de usar as amostras de referência dos blocos vizinhos para previsão, as amostras a partir de um bloco atual podem ser usadas para previsão melhorada. Por exemplo, algumas técnicas exemplares desta revelação descrevem modificações que podem ser aplicáveis a um modo de intra-previsão DC para codificação sem perdas no padrão HEVC. Além disso, algumas técnicas exemplares desta revelação descrevem modificações que podem ser aplicáveis a um modo de intra-previsão plano para codificação sem perdas no padrão HEVC. As técnicas desta revelação também podem ser aplicáveis a outros tipos de previsão, e também podem ser aplicáveis a outros padrões de codificação.

**[0078]** Na codificação sem perdas, ao invés de usar amostras de referência dos blocos vizinhos para previsão, as amostras a partir do bloco atual podem ser usadas para previsão melhorada. Por exemplo, técnicas para a intra-previsão angular para modos de codificação com perda assim como modos de codificação sem perdas quando a transformada é pulada são apresentados em Lan et al., "Intra and inter coding tools for screen contents," Equipe Colaborativa Conjunta em Codificação de vídeo (JCT-VC) de ITU-T SG 16 WP3 e ISO/IEC JTC1/SC29/WG11, 5<sup>th</sup> Meeting, Geneva, CH, 16-23 de março de 2011, documento JCTVC-E145

(doravante, "JCTVC-E145"). Além disso, para codificação sem perdas, as etapas de transformada e quantificação são puladas, assim pode ser possível melhorar mediante o processo para determinar as amostras de um bloco preditivo quando usando o modo intra DC.

**[0079]** De acordo com uma ou mais técnicas desta revelação, para cálculo do bloco preditivo, pode-se assumir que as amostras estão sendo processadas em uma varredura de rastreo ao longo das fileiras. No entanto, em outros exemplos, as mesmas técnicas podem ser extraídas para a varredura com rastreo ao longo das colunas (ou mesmo para varreduras diagonais ou em ziguezague, mas perdendo algum potencial de paralelização das técnicas). Quando nenhuma transformada ou quantificação é aplicada ao erro de previsão no modo sem perdas, pode-se supor que após a decodificação de entropia do erro de previsão (i.e., residual), a amostra original pode ser reconstruída sem perdas. Consequentemente, no contexto da codificação sem perdas, o termo "amostra original" ou "valor de amostra original" pode se referir ou aos valores de amostra originais reais ou valores de amostra reconstruídos (i.e., amostras não-residuais). Devido a varredura de rastreo ao longo das fileiras, todas as amostras das fileiras anteriores assim como todas as amostras à esquerda da amostra atual a partir da fileira atual estão disponíveis para previsão DC. Uma ou mais técnicas desta revelação tiram proveito disso para melhorar a intra-previsão usando o modo de intra-previsão DC .

**[0080]** Esta revelação descreve alguns modos de intra-previsão exemplares. Os modos de intra-previsão desta revelação podem substituir o modo de intra-previsão plano ou pode ser compreendido como modos de intra-previsão planos. Assim, os modos de intra-previsão desta revelação

podem substituir o modo plano atual na HEVC para codificação sem perdas. Os detalhes descritos nos exemplos desta revelação podem ser combinados com um ou mais detalhes de outros exemplos desta revelação. Ou seja, os detalhes podem ser combinados em qualquer de uma ampla variedade de diferentes maneiras de conseguir ainda outros exemplos.

**[0081]** De acordo com some exemplos desta revelação onde um codificador de vídeo (ex., o codificador de vídeo 20 ou decodificador de vídeo 30) gera um bloco preditivo usando um modo de intra-previsão DC, o codificador de vídeo pode processar amostras de um bloco preditivo em uma ordem de varredura de rastreamento. Quando o codificador de vídeo processa uma amostra do bloco preditivo, o codificador de vídeo pode usar uma vizinhança causal da amostra do bloco preditivo para formar um valor de previsão DC para a amostra. Em geral, uma vizinhança causal de uma amostra em um bloco preditivo é um conjunto de amostras reconstruídas (ex., amostras não-residuais, não-preditivas) que corresponde às amostras no bloco preditivo que já foram determinadas. Por exemplos, quando usa-se uma ordem de varredura de rastreamento ao longo das fileiras que começam a partir da amostra superior esquerda do bloco preditivo, a vizinhança causal de uma amostra no bloco preditivo pode incluir amostras reconstruídas que correspondem aos locais acima e à esquerda da amostra. Em tal exemplo, quando um codificador de vídeo gera um bloco preditivo usando um modo de intra-previsão DC, o codificador de vídeo calcula o valor de previsão DC,  $DC_{i,j}$ , para uma amostra atual  $P_{i,j}$ , onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ , como:

$$DC_{i,j} = (P_{i,j-1} + P_{i-1,j} + 1) \gg 1. \quad (4)$$

ou de modo similar,

$$DC_{i,j} = (P_{i,j-1} + P_{i-1,j}) \gg 1. \quad (5)$$

Assim, nas equações (4) e (5), para cada respectiva amostra em um bloco preditivo, o codificador de vídeo 20 pode determinar um valor de intra-previsão DC (i.e.,  $DC_{i,k}$ ) para a respectiva amostra como uma média de um valor reconstruído de uma amostra acima da respectiva amostra (i.e.,  $P_{i,j-1}$ ) e um valor reconstruído de uma amostra à esquerda da respectiva amostra (i.e.,  $P_{i-1,j}$ ). Porque o codificador de vídeo 20 está usando a codificação sem perdas, o valor reconstruído da amostra à esquerda da respectiva amostra e o valor reconstruído da amostra acima da respectiva amostra são os mesmos que os valores originais da amostra à esquerda da respectiva amostra e o valor original da amostra acima da respectiva amostra. De modo similar, nas equações (4) e (5), para cada respectiva amostra em um bloco preditivo, o decodificador de vídeo 30 pode determinar um valor de intra-previsão DC (i.e.,  $DC_{i,k}$ ) para a respectiva amostra como uma média de um valor reconstruído de uma amostra acima do respectiva amostra (i.e.,  $P_{i,j-1}$ ) e um valor reconstruído de uma amostra à esquerda da respectiva amostra (i.e.,  $P_{i-1,j}$ ). Porque o decodificador de vídeo 30 está usando a codificação sem perdas, o valor reconstruído da amostra acima da respectiva amostra e o valor reconstruído da amostra à esquerda da respectiva amostra são os mesmos que o valor original da amostra acima da respectiva amostra e o valor original da amostra à esquerda da respectiva amostra.

**[0082]** Devido a varredura de rastreamento e o processo de preenchimento para amostras de referência, as



amostras reconstruídas  $P_{i,j-1}$  e  $P_{i-1,j}$  podem estar sempre disponíveis. Em outras palavras, uma amostra acima da amostra atual e a amostra de referência à esquerda da amostra atual podem estar sempre disponíveis. Além disso, devido a natureza não linear da operação de deslocamento para a direita usada no cálculo do valor de previsão DC  $DC_{i,j}$ , pode ser difícil para o decodificador de vídeo 30 processar várias amostras em paralelo. Por exemplo,  $DC_{i,j+1}$  pode ser representado como

$$DC_{i,j+1} = (P_{i,j} + P_{i-1,j+1} + 1) \gg 1, \text{ or}$$

$$DC_{i,j+1} = ((R_{i,j} + ((P_{i,j-1} + P_{i-1,j} + 1) \gg 1) + P_{i-1,j+1} + 1) \gg 1,$$

Nas equações acima,  $R_{i,j}$  é a previsão residual para a amostra no local  $(i,j)$ . Porque o deslocamento de bit para a direita é processo não linear, podem não haver maneira de calcular  $DC_{i,j+1}$  antes de terminar o cálculo de  $DC_{i,j}$ . Por exemplo, pode ser difícil para o decodificador de vídeo 30 processar as várias amostras do bloco preditivo em paralelo.

**[0083]** Quando o decodificador de vídeo 30 decodifica a representação codificada de um bloco atual que é codificado usando a codificação sem perdas, o decodificador de vídeo 30 pode obter, a partir de um bitstream, elementos de sintaxe que indicam os valores de amostra residuais do bloco atual. Assim, quando o bloco atual é codificado usando a codificação sem perdas, o decodificador de vídeo 30 não precisa aplicar a quantificação inversa ou uma transformada inversa para determinar os valores de amostra residuais do bloco atual. Obter os elementos de sintaxe do bitstream pode envolver decodificar a entropia dos elementos de sintaxe. Dessa forma, quando o bloco atual é codificado usando codificação sem perda ou com perda, pode-se supor que as previsões

residuais (i.e., o erro de previsão) do bloco atual já foram decodificadas por entropia.

**[0084]** Assim, quando um bloco atual é codificado usando a codificação sem perda ou com perda, as previsões residuais estão disponíveis para uso na determinação dos valores de amostra reconstruídos que o decodificador de vídeo 30 usa na determinação dos valores de previsão DC para as amostras partir de um bloco preditivo para o bloco atual. Assumindo que as previsões residuais já foram decodificadas por entropia, pode ser possível canalizar o processamento das amostras em diferentes fileiras com um atraso de uma amostra. Assim, de acordo com uma ou mais técnicas desta revelação, o decodificador de vídeo 30 pode iniciar o processamento de uma segunda fileira de amostras após uma amostra a partir da primeira fileira ter sido reconstruída. Desta maneira, o decodificador de vídeo 30 pode processar várias fileiras de amostras em paralelo. Consequentemente, como parte da geração de um bloco preditivo, um codificador de vídeo pode canalizar o processamento de amostras em diferentes fileiras do bloco preditivo, sendo que um atraso de ciclo para a previsão DC existe entre as fileiras do bloco preditivo.

**[0085]** Em outro exemplo, um codificador de vídeo (ex., o codificador de vídeo 20 ou decodificador de vídeo 30) pode calcular um valor de previsão DC  $DC_{i,j}$ , onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ , como:

$$DC_{i,j} = (P_{i,j-1} + P_{i-1,j} - P_{i-1,j-1}). \quad (6)$$

Assim, neste exemplo, para cada respectiva amostra de um bloco preditivo, o codificador de vídeo 20 pode determinar um valor de previsão DC (i.e.  $DC_{i,j}$ ) para a

respectiva amostra como uma soma de um valor reconstruído de uma amostra acima da respectiva amostra (i.e.,  $P_{i,j-1}$ ) e um valor reconstruído de uma amostra à esquerda da respectiva amostra (i.e.,  $P_{i-1,j}$ ), menos um valor original de uma amostra imediatamente acima e à esquerda da respectiva amostra (i.e.,  $P_{i-1,j-1}$ ). De modo similar, para cada respectiva amostra de um bloco preditivo, o decodificador de vídeo 30 pode determinar um valor de previsão DC (i.e.  $DC_{i,j}$ ) para a respectiva amostra como uma soma de um valor reconstruído de uma amostra acima da respectiva amostra (i.e.,  $P_{i,j-1}$ ) e um valor reconstruído de uma amostra à esquerda da respectiva amostra (i.e.,  $P_{i-1,j}$ ), menos um valor reconstruído de uma amostra imediatamente acima e à esquerda da respectiva amostra (i.e.,  $P_{i-1,j-1}$ ). Porque a codificação sem perdas está sendo usada, o valor reconstruído das amostras são os mesmo que os valores originais das amostras.

**[0086]** Neste exemplo, pode ser mais simples processar vários valores de amostra em uma fileira usando lógica adicional. Por exemplo, um codificador de vídeo pode calcular  $DC_{i,j+1}$  para uma amostra atual  $P_{i,j}$  de um bloco preditivo sem esperar por  $P_{i,j}$  como segue.  $DC_{i,j+1}$  pode ser expresso como:

$$DC_{i,j+1} = ((P_{i,j-1} + P_{i-1,j} - P_{i-1,j-1}) + r_{i,j} + P_{i-1,j+1} - P_{i-1,j}) \quad (7)$$

Equação (7) pode ser reescrita como segue:

$$DC_{i,j+1} = \sum_{n=0}^j r_{i,j-n} + P_{i-1,j+1} + P_{i-1} + P_{i-1,-1}. \quad (8)$$

Na equações (7) e (8),  $r_{i,j}$  é o erro de previsão residual para a amostra  $P_{i,j}$ . Assim, neste exemplo, o

cálculo de um valor de intra-previsão DC para uma amostra específica no local  $(i, j+1)$  (i.e.,  $DC_{i,j+1}$ ) não é dependente dos valores reconstruídos das amostras à esquerda da amostra específica. Ao invés, o cálculo do valor de intra-previsão DC  $DC_{i,j+1}$  para uma amostra específica pode depender do valor reconstruído para a amostra diretamente acima, assim como os valores residuais para todas as amostras à esquerda da amostra atual, e amostras de referência na mesma fileira e a fileira acima. Isso pode permitir que o decodificador de vídeo 30 (assim como codificador de vídeo 20) calcule os valores de previsão DC para todas as amostras em uma fileira de um bloco em paralelo assumindo que os residuais para a fileira inteira já foram decodificados.

**[0087]** Porque a codificação sem perdas está sendo usada, o valor reconstruído das amostras (i.e.,  $P_{i,j-1}$ ,  $P_{i-1,j}$ , e  $P_{i-1,j-1}$ ) são os mesmo que os valores originais das amostras. Como descrito em outro lugar nesta revelação, esta técnica pode ser aplicada à codificação com perda. Nesse caso, para manter a paralelização, é necessário usar o valor reconstruído não fixado à amostra à esquerda para previsão DC. As amostras reconstruídas a partir da fileira acima podem ser fixadas ou não fixadas. Por exemplo, para uma sequência de vídeo de 8 bits, as amostras reconstruídas são fixadas para o intervalo  $[0, 255]$ .

**[0088]** Em outro exemplo, as amostras superiores da esquerda, superiores à esquerda e superiores à direita são usadas na determinação dos valores de intra-previsão DC. Neste exemplo, um codificador de vídeo pode calcular um valor de previsão DC  $DC_{i,j}$  para uma amostra atual  $P_{i,j}$  de um bloco preditivo, onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ , como:

$$DC_{i,j} = (P_{i,j-1} + P_{i-1,j} + P_{i-1,j-1} + P_{i-1,j+1} + 2) \gg 2, \quad (9)$$

ou de modo similar,

$$DC_{i,j} = (P_{i,j-1} + P_{i-1,j} + P_{i-1,j-1} + P_{i-1,j+1}) \gg 2. \quad (10)$$

Assim, nos exemplos das equações (9) e (10), para cada respectiva amostra de um bloco preditivo, um valor de intra-previsão DC para a respectiva amostra (i.e.,  $DC_{i,j}$ ) é uma média da amostra reconstruída acima da respectiva amostra (i.e.,  $P_{i,j-1}$ ), a amostra reconstruída à esquerda da respectiva amostra (i.e.,  $P_{i-1,j}$ ), a amostra reconstruída acima e à esquerda da respectiva amostra (i.e.,  $P_{i-1,j-1}$ ), e a amostra reconstruída acima e à direita da respectiva amostra (i.e.,  $P_{i-1,j+1}$ ). Porque a codificação sem perdas está sendo usada, os valores reconstruídos das amostras (i.e.,  $P_{i,j-1}$ ,  $P_{i-1,j}$ ,  $P_{i-1,j-1}$ , e  $P_{i-1,j+1}$ ) são os mesmo que os valores originais das amostras.

**[0089]** Nos exemplos usando as equações (9) ou (10) para determinar valores de intra-previsão DC, para a amostras na última coluna ( $j = (N - 1)$ ,  $i > 0$ ), a amostra superior direita não está disponível. Para superar isso, um codificador de vídeo pode assumir que as amostra superiores e superiores à direita (i.e.,  $P_{i-1,j}$  e  $P_{i-1,j+1}$ , respectivamente) tem o mesmo valor. Em outro exemplo onde um codificador de vídeo usa as equações (9) ou (10) para determinar os valores de intra-previsão DC, o codificador de vídeo pode modificar a previsão DC para usar somente as amostras disponíveis. Em alguns exemplos, uma amostra pode estar indisponível se a amostra não está dentro dos limites de uma Fatia ou imagem atual, ou não foi codificada ainda.

**[0090]** Além disso, de acordo com outro exemplo desta revelação, um codificador de vídeo pode realizar a intra-previsão DC em um tamanho de bloco menor do que o tamanho de TU. Por exemplo, independentemente do tamanho da TU, um codificador de vídeo pode realizar a intra-previsão DC nos blocos 2×2. O codificador de vídeo pode processar os blocos 2×2 de um bloco preditivo em uma ordem de varredura de rastreo. Neste exemplo, para as amostras  $P_{2i,2j}$ ,  $P_{2i,2j+1}$ ,  $P_{2i+1,2j}$ , e  $P_{2i+1,2j+1}$ , o codificador de vídeo calcula o valor de intra-previsão DCs como:

$$(P_{2i-1,2j} + P_{2i-1,2j+1} + P_{2i,2j-1} + P_{2i+1,2j-1} + 2) \gg 2, \quad (11)$$

ou de modo similar,

$$(P_{2i-1,2j} + P_{2i-1,2j+1} + P_{2i,2j-1} + P_{2i+1,2j-1}) \gg 2. \quad (12)$$

Neste exemplo,  $0 \leq i \leq ((M / 2) - 1)$  e  $0 \leq j \leq ((N / 2) - 1)$ , onde M é a altura do bloco e N é a largura do bloco. Além disso, neste exemplo, assume-se que M e N são ambos pares. Neste exemplo, o codificador de vídeo pode processar quatro amostras em paralelo. Neste exemplo, o codificador de vídeo pode ser capaz de determinar os valores de intra-previsão DC de cada uma das quatro amostras de um bloco 2×2 em paralelo. Em exemplos similares, o codificador de vídeo pode usar os blocos 4×4 ou blocos 8×8 ao invés de blocos 2×2.

**[0091]** De acordo com outro exemplo desta revelação, a correlação entre os residuais após a realização da previsão DC normal é explorada. Por exemplo, vamos que  $r_{i,j}$ , sendo que  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ , seja os residuais de previsão após a realização da intra-previsão DC como especificado em HEVC (ex., HEVC Working

Draft 10). Por exemplo,  $r_{i,j}$  pode ser um valor residual de previsão após a realização da intra-previsão DC como descrito na equação (1) acima, para um bloco  $4 \times 4$ . Neste exemplo, um codificador de vídeo pode então gerar valores intermediários  $s_{i,j}$ , onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ . O codificador de vídeo pode gerar os valores intermediários  $s_{i,j}$  as:

$$s_{i,j} = r_{i,2j}, \quad s_{i,(j+(N/2))} = r_{i,2j} - r_{i,2j+1} \quad (13)$$

Na equação (13), acima,  $0 \leq i \leq (M - 1), 0 \leq j \leq ((N/2) - 1)$ .

**[0092]** O codificador de vídeo pode então gerar valores residuais modificados  $t_{i,j}$ , onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ , como segue:

$$t_{i,j} = s_{2i,j}, \quad t_{(i+(M/2)),j} = s_{2i,j} - s_{2i+1,j} \quad (14)$$

Na equação (14),  $0 \leq i \leq ((M/2) - 1)$  e  $0 \leq j \leq (N - 1)$ .

**[0093]** O codificador de vídeo pode codificar a entropia dos residuais modificados,  $t_{i,j}$ , como descrito no HEVC regular (ex., HEVC Working Draft 10). No lado do decodificador (ex., no decodificador de vídeo 30), este processo é revertido. Por exemplo, o decodificador de vídeo 30 pode determinar,

$$s_{2i,j} = t_{i,j}, \quad s_{2i+1,j} = t_{i,j} - t_{(i+(M/2)),j}, \quad (15)$$

onde  $0 \leq i \leq ((M / 2) - 1)$  e  $0 \leq j \leq (N - 1)$ .  
Decodificador de vídeo 30 também pode determinar,

$$r_{i,2j} = s_{i,j}, \quad r_{i,2j+1} = s_{i,j} - s_{i,(j+(N/2)),j}, \quad (16)$$

onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq ((N / 2) - 1)$ .  
Este exemplo assume que M e N são ambos pares.

**[0094]** Em outro exemplo desta revelação, um previsor potencialmente melhor pode ser usado ao invés de tomar a diferença simples. Por exemplo, um codificador de vídeo pode determinar  $s_{i,j}$  como segue:

$$s_{i,j} = r_{i,2j+1}, \quad s_{i,(j+(N/2))} = P_{i,2j} - ((P_{i,2j-1} + P_{i,2j+1} + 1) \gg 1) \quad (17)$$

$$t_{i,j} = s_{2i+1,j}, \quad t_{(i+(M/2)),j} = s_{2i,j} - ((s_{2i+1,j} + s_{2i+1,j} + 1) \gg 1) \quad (18)$$

Na equação (17), acima,  $0 \leq i < M, 0 \leq j < (N/2)$ . Na equação (18), acima,  $0 \leq i < M/2, 0 \leq j < N$ .

**[0095]** As técnicas descritas em vários outros exemplos desta revelação podem ser aplicados para melhorar o modo de intra-previsão DC na codificação com perda quando a transformada é pulada. Em outras palavras, vários outros exemplos desta revelação podem ser aplicados para melhorar o modo de intra-previsão DC quando um codificador de vídeo não aplica uma transformada às amostras residuais de um bloco de transformada, mas quantifica as amostras residuais de um bloco transformado. Por exemplo, no exemplo descrito no parágrafo [0081] acima, uma vizinhança causal é usada para calcular o valor de previsão DC para a amostra atual. Neste exemplo, ao invés de usar valores de amostra originais para calcular o valor de previsão DC como é feito no caso da codificação sem perdas, valores de amostra



reconstruídos (quantificados) na vizinhança causal podem ser usados. Porque a aplicação da transformada é pulada, os valores reconstruídos na vizinhança causal estão disponíveis. Deve-se observar que para manter os benefícios da paralelização, a operação de fixação não é aplicada aos valores reconstruídos a partir da fileira atual até o processamento para a fileira inteira ser concluído. Para a fileira acima, tantos os valores reconstruídos fixados ou não fixados podem ser usados.

**[0096]** De modo similar, como descrito no parágrafo [0090] acima, a TU é dividida em blocos menores (ex., blocos 2×2) e um valor de previsão DC é calculado para cada bloco menor. Ao invés de usar valores de amostra originais para calcular o valor de previsão DC como é feito no caso da codificação sem perdas, os valores de amostra reconstruídos (quantificados) podem ser usados no caso da codificação com perda onde a transformada é pulada.

**[0097]** Técnicas de codificação com perda são descritas acima nesta revelação. De acordo com algumas dessas técnicas, deixe que  $P_{i,j}$ , onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ , denote valores de amostra originais. Além disso, deixe que  $Q(P_{i,j})$  denote uma versão quantificada de  $P_{i,j}$ . Então, de acordo com um exemplo adicional desta revelação que usa a codificação com perda, um codificador de vídeo pode calcular o valor de previsão DC  $DC_{i,j}$  como:

$$DC_{i,j} = (Q(P_{i,j-1}) + Q(P_{i-1,j}) - Q(P_{i-1,j-1})) \quad (\text{Equ. DC1})$$

Note que a equação DC1 é similar à equação (6), acima, exceto pelo fato de que os valores de amostra (i.e.,  $P_{i,j-1}$ ,  $P_{i-1,j}$ , e  $P_{i-1,j-1}$ ) são quantificados e depois desquantificados. Esta revelação pode se referir a tais amostras como amostras quantificadas ou amostras

desquantificadas das amostras originais. Assim, na equação DC1, para cada respectiva amostra de um bloco preditivo, o codificador de vídeo pode calcular um valor de previsão DC para a respectiva amostra como uma soma de uma versão quantificada da amostra original acima da respectiva amostra (i.e.,  $Q(P_{i,j-1})$ ) e uma versão quantificada da amostra original à esquerda da respectiva amostra (i.e.,  $Q(P_{i-1,j})$ ), menos uma versão quantificada da amostra original acima e à esquerda da respectiva amostra (i.e.,  $Q(P_{i-1,j-1})$ ). Na equação DC1, os termos da forma  $Q(P_{i,j})$  são amostras reconstruídas. O codificador de vídeo pode então calcular a previsão residual como  $r_{i,j} = P_{i,j} - DC_{i,j}$ . Em outras palavras, o valor residual  $r_{i,j}$  é igual ao valor de amostra  $P_{i,j}$  menos o valor de intra-previsão DC correspondente  $DC_{i,j}$ . O residual reconstruído após a quantificação e desquantificação é denotado por  $Q(r_{i,j})$ .

**[0098]** O exemplo descrito no parágrafo anterior pode ter algumas propriedades de capacidade de transmissão desejáveis no lado do decodificador. Por exemplo, pode ser possível para o decodificador de vídeo 30 calcular os valores de amostra reconstruídos para todas as amostras em uma fileira (ou coluna) de um bloco simultaneamente. Por exemplo, o decodificador de vídeo 30 pode obter os valores de amostra reconstruídos como:

$$Q(P_{i,j}) = Q(r_{i,j}) + DC_{i,j}, \text{ or}$$

$$Q(P_{i,j}) = (\sum_{n=0}^j Q(r_{i,j-n})) + Q(P_{i-1,j}) + Q(P_{i-1}) + Q(P_{i-1,-1}) \text{ (Equ. DC2)}$$

Na equação DC2,  $Q(P_{i-1,j})$  denota uma amostra reconstruída que pode ser adequadamente fixada. Por exemplo, para fixar adequadamente um valor reconstruído com

uma profundidade de bit de entrada de 8, os valores de  $Q(P_{i-1,j})$  são fixados entre 0 e 255.

**[0099]** Além disso, em alguns exemplos, pode ser possível usar versões não fixadas de  $Q(P_{i-1,j})$ . Os outros valores,  $Q(P_{i-1,-1})$  e  $Q(P_{i,-1})$  pertencem aos blocos anteriormente reconstruídos e já são fixados. Em tais exemplos, a amostra reconstruída  $Q(P_{i-1,j})$  na equação DC2 é não fixada mas pode ser fixada adequadamente sem afetar a capacidade de transmissão. A previsão especificada na equação DC1 só é aproximado se o decodificador de vídeo usa a equação DC2 para reconstrução. A previsão na equação DC1 é somente aproximada porque ao invés de  $Q(P_{i,j-1})$ , que é uma versão fixada, uma versão não fixada é usada. Em tal caso, a versão não fixada de  $Q(P_{i,j-1})$  pode ser usada no lado do codificador assim como para gerar a previsão DC para evitar um deslocamento entre o codificador e o decodificador. É possível usar a versão fixada, mas então as amostras podem ter de ser reconstruídas uma por uma, afetando assim a capacidade de transmissão. Isto é porque, nesse caso, a equação DC1 pode ter de ser usada para reconstrução. Isso significa que a reconstrução de uma amostra pode depender da conclusão da reconstrução de uma amostra à esquerda da amostra. Foi descrito nesta revelação como uma fileira de amostras pode ser reconstruída em paralelo. Um processo similar pode ser seguido para reconstruir todas as amostras em uma coluna em paralelo. Se menos paralelismo é desejado, o termo somatório pode ser quebrado em pedaços menores, reduzindo potencialmente assim a capacidade de transmissão mas reduzindo o número médio das operações adicionais necessário para reconstruir uma amostra.

**[00100]** Assim, em pelo menos alguns dos exemplos de codificação com perda descritos acima, um codificador de vídeo pode gerar um bloco preditivo. Como parte da geração do bloco preditivo, o codificador de vídeo pode usar pelo menos uma de uma primeira amostra reconstruída (ex.,  $Q(P_{i-1,j})$ ) e uma segunda amostra reconstruída (ex.,  $Q(P_{i,j-1})$ ) para previsão DC de uma amostra atual do bloco preditivo. A primeira amostra reconstruída pode corresponder a uma amostra à esquerda da amostra atual na fileira atual do bloco preditivo. A segunda amostra reconstruída pode corresponder a uma amostra em uma fileira do bloco preditivo acima da fileira atual. O codificador de vídeo pode reconstruir um bloco de codificação que foi codificado usando a codificação com perda pela adição das amostras do bloco preditivo às amostras residuais correspondentes.

**[00101]** Além disso, em pelo menos alguns dos exemplos de codificação com perda descritos acima, o codificador de vídeo pode determinar um valor reconstruído  $Q(P_{i,j-1})$  correspondente a uma amostra acima da amostra atual. Além disso, o codificador de vídeo pode determinar um valor reconstruído  $Q(P_{i-1,j})$  correspondente a uma amostra à esquerda da amostra atual. O codificador de vídeo também pode determinar um valor reconstruído  $Q(P_{i-1,j-1})$  correspondente a uma amostra à esquerda e acima da amostra atual. O codificador de vídeo pode calcular um valor de previsão DC  $DC_{i,j}$  para a amostra atual  $P_{i,j}$  como:

$$DC_{i,j} = (Q(P_{i,j-1}) + Q(P_{i-1,j}) - Q(P_{i-1,j-1})).$$

**[00102]** Em pelo menos alguns dos exemplos de codificação com perda descritos acima, o codificador de vídeo pode, para cada respectivo valor reconstruído dentre  $Q(P_{i,j-1})$ ,  $Q(P_{i-1,j})$ , e  $Q(P_{i-1,j-1})$ , determinar o respectivo valor reconstruído em uma das seguintes maneiras. Primeiro, o codificador de vídeo pode determinar o respectivo valor reconstruído como um valor residual desquantificado para uma dada amostra (ex.,  $Q(r_{i,j})$ ) mais um valor de previsão DC para a amostra correspondente (ex.,  $DC_{i,j}$ ). Segundo, como descrito na equação DC2, o codificador de vídeo pode determinar o respectivo valor reconstruído como uma soma dos valores residuais desquantificados para as amostras acima da dada amostra (ex.,  $\sum_{n=0}^j Q(r_{i,j-n})$ ), mais um valor reconstruído correspondente a uma amostra à esquerda da dada amostra (ex.,  $Q(P_{i-1,j})$ ), mais um valor reconstruído correspondente a uma amostra de referência acima da amostra mais superior da coluna do bloco preditivo contendo a dada amostra (ex.,  $Q(P_{i,-1})$ ), mais um valor reconstruído correspondente a uma amostra de referência acima da amostra mais superior da coluna do bloco preditivo à esquerda da dada amostra (ex.,  $Q(P_{i-1,-1})$ ). Neste exemplo, a dada amostra corresponde ao respectivo valor reconstruído. Em alguns desses exemplos, o codificador de vídeo pode fixar o valor reconstruído correspondente a amostra à esquerda da dada amostra.

**[00103]** Outro exemplo desta revelação propõe uma modificação para o processo de previsão para o modo plano. Olhando para as equações para o modo plano (ex., equação (2)), a previsão vertical pode ser mais precisa se as amostras originais a partir da última fileira são usadas para realizar a previsão vertical ao invés de usar  $P_{M,-1}$ . De modo similar, a previsão horizontal pode ser mais precisa se as amostras originais a partir das última coluna

são usadas para realizar a previsão horizontal ao invés de usar  $P_{-1,N}$ . O uso das amostras originais no modo intra-plano é uma ideia básica atrás de um ou mais exemplos desta revelação.

**[00104]** Por exemplo, um codificador de vídeo (ex., o codificador de vídeo 20) pode usar a previsão plana de HEVC (ex., previsão plana como descrito em HEVC Working Draft 10) para a última fileira e coluna, ou seja,  $T_{i,j}$ , onde  $i = (M - 1)$  ou  $j = (N - 1)$ . O codificador de vídeo pode subtrair os valores de previsão a partir dos valores de amostra originais para gerar residuais para a última fileira e coluna. Ou seja, o codificador de vídeo pode determinar  $r_{i,j}$ , onde  $i = (M - 1)$  ou  $j = (N - 1)$ . O codificador de vídeo pode então gerar os valores de previsão  $T_{i,j}$ , onde  $0 \leq i \leq (M - 2)$  e  $0 \leq j \leq (N - 2)$ , como segue:

$$\begin{aligned} T_{i,j}^V &= (M - i) * P_{-1,j} + i * P_{M-1,j}, \\ T_{i,j}^H &= (N - j) * P_{i,-1} + j * P_{i,N-1} \text{ and} \\ T_{i,j} &= (T_{i,j}^V + T_{i,j}^H + N) \gg (\log_2 N + 1). \end{aligned} \tag{19}$$

Este exemplo assume que  $M = N$ . No entanto, a extensão deste conceito aos blocos retangulares é direta. O codificador de vídeo pode gerar os residuais restantes,  $r_{i,j}$ , onde  $0 \leq i \leq (M - 2)$  e  $0 \leq j \leq (N - 2)$ , pela subtração dos valores de previsão a partir dos valores de amostra originais. O codificador de vídeo pode codificar a entropia de todo o bloco de residuais,  $r_{i,j}$ , onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ , como na HEVC (ex., HEVC Working Draft 10).

**[00105]** Neste exemplo, no lado do decodificador, um decodificador de vídeo (ex., o decodificador de vídeo 30) pode decodificar por entropia

todo o bloco de residuais de previsão para gerar valores residuais  $r_{i,j}$ , onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ . O decodificador de vídeo pode então realizar a previsão plana de HEVC para as amostras na última fileira e coluna do bloco. Ou seja, o codificador de vídeo pode determinar  $T_{i,j}$ , onde  $i = (M - 1)$  ou  $j = (N - 1)$ , como especificado na HEVC (ex., HEVC Working Draft 10). Por exemplo, o codificador de vídeo pode determinar  $T_{i,j}$  usando a equação (3), acima. Além disso, neste exemplo, o decodificador de vídeo adiciona valores residuais para a última fileira e coluna aos valores de previsão acima para reconstruir os valores de amostra originais para a última fileira e coluna. Após isso, o decodificador de vídeo gera os valores de previsão  $T_{i,j}$ , onde  $0 \leq i \leq (M - 2)$  e  $0 \leq j \leq (N - 2)$ , exatamente como no lado do codificador acima. O decodificador de vídeo adiciona valores residuais  $r_{i,j}$ , onde  $0 \leq i \leq (M - 2)$  e  $0 \leq j \leq (N - 2)$ , aos valores de previsão para reconstruir os valores de amostra para as posições de amostra restantes no bloco.

**[00106]** Em outro exemplo desta revelação, um codificador de vídeo (como codificador de vídeo 20) usa a previsão plana de HEVC (ex., previsão planar como descrito em HEVC Working Draft 10) para o elemento na última fileira e coluna, ou seja,  $T_{i,j}$ , onde  $i = (M - 1)$  e  $j = (N - 1)$ . Neste exemplo, o codificador de vídeo subtrai um valor de previsão para uma posição a partir do valor de amostra original para a posição para gerar um valor residual para aquela posição. O codificador de vídeo pode então prever os elementos da última fileira e coluna bilinearmente como:

$$\begin{aligned} T_{M-1,j} &= ((N - j) * P_{M-1,-1} + j * P_{M-1,N-1}) \gg (\log_2 N), \\ T_{i,N-1} &= ((M - i) * P_{-1,N-1} + i * P_{M-1,N-1}) \gg (\log_2 M). \end{aligned} \quad (20)$$

Além disso, neste exemplo, o codificador de vídeo gera os valores de previsão  $T_{i,j}$ , onde  $0 \leq i \leq (M - 2)$  e  $0 \leq j \leq (N - 2)$ , como segue:

$$\begin{aligned} T_{i,j}^V &= (M - i) * P_{-1,j} + i * P_{M-1,j}, \\ T_{i,j}^H &= (N - j) * P_{i,-1} + j * P_{i,N-1} \text{ e} \\ T_{i,j} &= (T_{i,j}^V + T_{i,j}^H + N) \gg (\log_2 N + 1). \end{aligned} \quad (21)$$

Neste exemplo, o codificador de vídeo gera valores residuais  $r_{i,j}$ , onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$  pela subtração dos valores de previsão a partir dos valores de amostra originais. Além disso, neste exemplo, o codificador de vídeo codifica a entropia de todos os blocos de residuais,  $r_{i,j}$ , onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$  como na HEVC (ex., HEVC Working Draft 10).

**[0107]** Outro exemplo desta revelação fornece um refinamento nos exemplos anteriores. Neste exemplo, a distribuição de resíduo devido às técnicas de previsão desta revelação tende a ser a reversa de uma comum na compressão de vídeo quando a transformada é empregada. Comumente, o resíduo tem valores mais altos em frequências mais baixas, e valores esperados mais baixos em frequências mais altas. Para o resíduo que vem a partir dos exemplos acima, a última fileira e coluna tendem a ter valores maiores. Uma abordagem para melhorar o desempenho enquanto tira proveito do método de codificação de entropia projetado para um resíduo de transformada é girar o resíduo que vem a partir da previsão nos exemplos fornecidos acima. Ou seja, o resíduo é girado 180 graus, assim a parte superior esquerda se torna a inferior direita, e vice versa. Então, este resíduo girado é codificado por entropia. Correspondentemente, no decodificador, o resíduo é obtido e então girado 180 graus.



**[0108]** Em outros exemplos desta revelação, o processo de previsão plano é modificado como segue. Para a primeira fileira e coluna, a previsão é realizada como no caso do modo plano de HEVC (ex., modo plano como descrito em HEVC Working Draft 10) para gerar  $T_{i,j}$  e  $r_{i,j}$ ,  $i = 0$  ou  $j = 0$ . Neste exemplo, ao invés de usar o método de HEVC (ex., o método descrito em HEVC Working Draft 10, equação (3), etc.) para gerar a previsão para a primeira fileira e coluna, outros métodos são usados. Por exemplo,  $P_{0,0}$  pode ser previsto como  $(P_{-1,0} + P_{0,-1} + 1) \gg 1$ . As amostras restantes na primeira coluna podem ser previstas usando a amostra da esquerda na mesma fileira. De modo similar, as amostras restantes na primeira fileira podem ser previstas usando a amostra acima na mesma coluna. Para as posições restantes, a previsão plana,  $T_{i,j}$ , onde  $1 \leq i \leq (M - 1)$  e  $1 \leq j \leq (N - 1)$ , é gerado como segue:

$$\begin{aligned} T_{i,j}^V &= P_{i-1,j} + w_v * (P_{i-1,j} - P_{i-2,j}), \\ T_{i,j}^H &= P_{i,j-1} + w_h * (P_{i,j-1} - P_{i,j-2}) \text{ e} \\ T_{i,j} &= (T_{i,j}^V + T_{i,j}^H + 1) \gg 1. \end{aligned} \quad (22)$$

Na equação (22),  $w_v$  e  $w_h$  são pesos. Em alguns exemplos, um valor de 0,5 é usado para ambos  $w_v$  e  $w_h$  desde que o valor de 0,5 possa ser implementado como um deslocamento de bit. Os residuais restantes (i.e.,  $r_{i,j}$ , onde  $1 \leq i \leq (M - 1)$  e  $1 \leq j \leq (N - 1)$ ), são gerados pela subtração dos valores de previsão a partir dos valores de amostra originais. Todo o bloco de residuais (i.e.,  $r_{i,j}$ , onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ ), pode ser codificado por entropia como na HEVC (ex., HEVC Working Draft 10).

**[0109]** Neste exemplo, no lado do decodificador, um decodificador de vídeo (ex., o

decodificador de vídeo 30) pode decodificar por entropia todo o bloco de residuais de previsão para gerar valores residuais (i.e.,  $r_{i,j}$ , onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ ). Então, o decodificador de vídeo realiza a previsão plana para as amostras na primeira fileira e coluna (i.e.,  $T_{i,j}$ , onde  $i = 0$  ou  $j = 0$ ) como especificado na HEVC (ex., HEVC Working Draft 10) ou como descrito acima, por qualquer outro método usado pelo codificador de vídeo. Além disso, neste exemplo, o decodificador de vídeo adiciona os valores residuais para a primeira fileira e coluna aos valores de previsão acima para reconstruir os valores de amostra originais para a primeira fileira e coluna. Subsequentemente, o decodificador de vídeo gera os valores de previsão (i.e.,  $T_{i,j}$ , onde  $1 \leq i \leq (M - 1)$  e  $1 \leq j \leq (N - 1)$ ) exatamente como no lado do codificador acima. Neste exemplo, o decodificador de vídeo adiciona valores residuais (i.e.,  $r_{i,j}$ , onde  $1 \leq i \leq (M - 1)$  e  $1 \leq j \leq (N - 1)$ ) aos valores de previsão para reconstruir o valores de amostra para as posições restantes no bloco.

**[0110]** Os parágrafos acima descrevem uma maneira diferente de realizar a previsão plana para a codificação sem perdas. Em um exemplo adicional desta revelação, a coluna mais à direita e fileira inferior são previstas usando um procedimento de previsão plana como descrito na HEVC (ex., HEVC Working Draft 10). Neste exemplo, os valores de amostra originais para a coluna mais à direita e a fileira inferior são então usados para realizar a previsão plana ou angular para as amostras restantes do bloco. No caso da codificação com perda quando a transformada é pulada, ao invés de usar os valores de amostra originais para a coluna mais à direita e a fileira inferior, os valores de amostra reconstruídos

(quantificados) podem ser usados para realizar a previsão plana ou angular nas amostras restantes.

**[0111]** Na HEVC (ex., HEVC Working Draft 10), os valores de previsão DC para a primeira fileira e coluna são filtrados (filtragem da previsão DC). De modo similar, para os modos de intra-previsão horizontais e verticais, a primeira fileira e coluna dos valores de previsão, respectivamente, são filtrados (filtragem de gradiente). Quando os métodos descritos acima são aplicados à codificação com perda para DC, os modos de intra-previsão horizontais ou verticais, e quando a transformada é pulada, a filtragem da previsão DC ou filtragem de gradiente podem ser pulada.

**[0112]** Em outro exemplo desta revelação, ao invés de alterar o modo de previsão plano, o mesmo conceito é aplicado aos modos de intra-previsão angulares. Para cada modo angular, a última fileira e coluna são previstas como especificado na HEVC (ex., HEVC Working Draft 10, equação (3), etc.). Então, os valores de amostra originais para a última fileira e coluna são usados para realizar a intra-previsão além das amostras de referência. FIG. 4 mostra as amostras que são usadas para a previsão. As posições sombreadas na FIG. 4 são as posições usadas como amostras de referência para realizar a previsão. Se, para uma direção de previsão específica e uma posição específica, as amostras de referência  $P_{-1,j}$ , onde  $N \leq j \leq 2N$  (ou sua interpolação bilinear), teriam sido usadas como o valor de previsão, então os valores de amostra originais a partir da coluna da direita (i.e.,  $P_{i,N-1}$ ,  $0 \leq i \leq M - 1$ ) são usados ao invés. Se a posição onde o ângulo de previsão cruza a coluna mais à direita é uma fração, interpolação bilinear ou qualquer outro método de interpolação adequado pode ser usado. Como um exemplo,

considere um bloco  $4 \times 4$ . Para o modo de intra-previsão 34, para a amostra  $(2, 2)$ , a amostra de referência correspondente seria  $P_{-1,6}$ . Assim, para o modo 34, a previsão de HEVC seria  $T_{2,2} = P_{-1,5}$ . Ao invés, neste exemplo,  $T_{2,2} = P_{1,3}$ .

**[0113]** De modo similar, se para uma direção de previsão específica e uma posição específica, as amostras de referência  $P_{i,-1}$ ,  $M \leq i \leq 2M$  (ou sua interpolação bilinear) teriam sido usadas como valores de previsão, então os valores de amostra originais a partir da fileira inferior (i.e.,  $P_{M-1,j}$ ,  $0 \leq j \leq N - 1$ ) são usados ao invés. Se a posição onde o ângulo de previsão cruza a fileira inferior é uma fração, a interpolação bilinear ou outra técnica de interpolação adequada pode ser usada. Como um exemplo, considere um bloco  $4 \times 4$ . Neste exemplo, para o modo de intra-previsão 2, para amostra  $(2, 1)$ , a amostra de referência correspondente é  $P_{4,-1}$ . Assim, a previsão da HEVC para o modo 2 seria  $T_{2,1} = P_{4,-1}$ . Ao invés, neste exemplo,  $T_{2,1} = P_{3,0}$ .

**[0114]** Em alguns exemplos adicionais desta revelação, o codificador de vídeo 20 e decodificador de vídeo 30 podem realizar a codificação que usa um ou mais valores de amostra originais dentro de um bloco de dados de vídeo para realizar a previsão de outros valores de amostra dentro do bloco. Os valores de amostra originais podem corresponder a uma última fileira e uma última coluna do bloco (ex., a fileira inferior e a coluna mais à direita do bloco). Como outro exemplo, os valores de amostra originais podem corresponder a uma primeira fileira e uma primeira coluna do bloco (ex., a fileira superior e a colina mais à esquerda). A FIG. 4 ilustra um exemplo de valores de amostra usados para realizar a previsão de outros valores de amostra. Neste exemplo, o codificador de

vídeo pode realizar um modo de codificação sem perdas, e o modo de codificação sem perdas pode compreender um modo de codificação plano, um modo de intra-codificação angular, ou outro modo. Em alguns exemplos, as técnicas podem adicionalmente incluir uma operação de rotação no conjunto de valores residuais gerado pela previsão, seguido pela codificação de entropia com relação ao conjunto girado dos valores residuais.

**[0115]** Métodos de modulação de código de pulso diferencial (DPCM) são propostos em Lee et al., "AHG7: Residual DPCM for HEVC sem perdas coding" Equipe Colaborativa Conjunta em Codificação de vídeo (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 12<sup>th</sup> Meeting, Geneva, CH, 14-23 de janeiro de 2013, documento n°. JCTVC-L0117 (doravante, "JCTVC-L0117"). JCTVC-L0117 propôs uma melhoria à intra-previsão para os modos horizontais (ex., modo de intra-previsão 10) e verticais (ex., modo de intra-previsão 26) em HEVC para a codificação sem perdas. Esta melhoria foi denotada pela DPCM residual. Em JCTVC-L0117, a DPCM residual é aplicada quando uma CU está sendo codificada sem perdas. A ideia básica da DPCM residual é usar o pixel da fileira superior para prever o pixel atual para o modo vertical e para usar o pixel da coluna esquerda para prever o pixel atual para o modo vertical.

**[0116]** A DPCM residual, como descrito em JCTVC-L0117, pode ser descrita como segue. Considere um bloco de tamanho  $M$  (fileiras)  $\times$   $N$  (colunas). Deixe  $r_{i,j}$ , onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ , ser a previsão residual após a realização da intra-previsão como especificado em HEVC Working Draft 10. Isto é mostrado nas FIGS. 5A e 5B. O bloco poderia representar qualquer componente (ex., luma, croma, R, G, B, etc.). No método proposto em JCTVC-L0117, a DPCM residual é aplicada às

amostras residuais, de modo que a matriz modificado  $M \times N$   $\tilde{R}$  com elementos  $\tilde{r}_{i,j}$  é obtido como segue quando o modo de intra-previsão é vertical:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j} - r_{(i-1),j}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases} \quad (23)$$

Quando o modo de intra-previsão é horizontal,  $\tilde{r}_{i,j}$  é obtido como segue:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j=0 \\ r_{i,j} - r_{i,(j-1)}, & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad (24)$$

As amostras residuais modificadas de  $\tilde{R}$  são sinalizadas para o decodificador de vídeo, ao invés das amostras residuais originais  $R$ . Isto pode ser equivalente ao uso de  $P_{i-1,j}$  como a previsão para  $P_{i,j}$  para um modo de previsão vertical e usar  $P_{i,j-1}$  como uma previsão para  $P_{i,j}$  para o modo de previsão horizontal.

**[0117]** Além disso, no método proposto em JCTVC-L0117, no lado do decodificador, quando o modo de intra-previsão é um modo vertical, as amostras residuais originais podem ser reconstruídas após as amostras residuais modificadas serem analisadas como segue:

$$r_{i,j} = \sum_{k=0}^i \tilde{r}_{k,j}, \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1), \quad (25)$$

Quando o modo de intra-previsão é um modo horizontal, as amostras residuais originais podem ser

reconstruídas após as amostras residuais modificadas serem analisadas como segue:

$$r_{i,j} = \sum_{k=0}^j \tilde{r}_{i,k}, \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1). \quad (26)$$

**[0118]** Além disso, esta revelação discute os métodos para estender as técnicas de DPCM residuais propostas em JCTVC-L0117 para os modos de intra-previsão horizontais e verticais para outros modos de intra-previsão angulares para codificação sem perdas. Porque a codificação é sem perdas, as amostras vizinhas originais (na ordem de codificação causal) assim como os residuais de previsão correspondentes estão disponíveis para previsão (porque transformada e quantificação são puladas).

**[0119]** Em um primeiro exemplo, a técnica de DPCM residual pode ser estendida para outros modos de intra-previsão angulares. A FIG. 3 mostra as direções de intra-previsão para diferentes modos de previsão angulares (de 2 a 34). Considere agora um modo entre 22 e 30. Para cada um destes modos, a direção de previsão pode ser considerada como próxima à vertical (quase vertical). Os números 22 e 30 são apenas exemplos. Outras faixas (ex., modos de intra-previsão entre 24 e 28) podem ser escolhidas também. Considere agora que o residual  $r_{i,j}$ , onde  $0 \leq i \leq (M-1)$  e  $0 \leq j \leq (N-1)$ , é calculado como especificado na HEVC (ex., HEVC Working Draft 10). Ou seja, a previsão é realizada de acordo com o modo angular e é subtraída a partir dos valores de amostra originais para conseguir o residual  $r_{i,j}$ . A DPCM residual pode ser aplicada às amostras residuais exatamente como em JCTVC-L0117 para obter a matriz  $M \times N$  modificado  $\tilde{R}$  com elementos  $\tilde{r}_{i,j}$  como segue:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j} - r_{(i-1),j}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases} \quad (27)$$

**[0120]** De modo similar, neste primeiro exemplo, considere um modo de intra-previsão direcional entre 6 e 14. Para cada um destes modos de intra-previsão, a direção de previsão pode ser considerada como próxima à horizontal (i.e., quase horizontal). Os números 6 e 14 são apenas exemplos. Em outros exemplos, outras faixas (ex., modos de intra-previsão entre 8 e 12) podem ser usados. Agora, considere que o residual  $r_{i,j}$ , onde  $0 \leq i \leq (M-1)$  e  $0 \leq j \leq (N-1)$  é calculado como especificado na HEVC (ex., HEVC Working Draft 10). Ou seja, a previsão é realizada de acordo com o modo angular e a previsão é subtraída a partir dos valores de amostra originais para determinar o residual  $r_{i,j}$ . A DPCM residual é aplicada às amostras residuais exatamente como em JCTVC-L0117 para obter a matriz  $M \times N$  modificado  $\tilde{R}$  com elementos  $\tilde{r}_{i,j}$  como segue:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j=0 \\ r_{i,j} - r_{i,(j-1)}, & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad (28)$$

Como no caso da JCTVC-L0117, as amostras residuais modificadas de  $\tilde{R}$  são sinalizadas para o decodificador.

**[0121]** Além disso, neste primeiro exemplo, no lado do decodificador, as amostras residuais originais podem ser reconstruídas após as amostras residuais modificadas serem analisadas como segue. Quando o modo de intra-previsão é quase vertical (ex., modos 22 a 30,



inclusive), as amostras residuais originais podem ser reconstruídas como:

$$r_{i,j} = \sum_{k=0}^i \tilde{r}_{k,j}, \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1), \quad (29)$$

Quando o modo de intra-previsão é quase horizontal (ex., modos 6 a 14, inclusive), as amostras residuais originais podem ser reconstruídas como:

$$r_{i,j} = \sum_{k=0}^j \tilde{r}_{i,k}, \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1). \quad (30)$$

Uma vez que o residual  $r_{i,j}$  foi calculado, o residual  $r_{i,j}$  é adicionado à previsão realizada de acordo com o modo angular para obter as amostras originais. Quando se obtém a previsão para os modos horizontais e verticais pode ser possível habilitar ou desabilitar a adição de um termo gradiente à previsão para a primeira coluna (para o modo vertical) ou a primeira fileira (para o modo horizontal).

**[0122]** Em um segundo exemplo onde as técnicas de DPCM residuais são estendidas aos modos de intra-previsão angulares para codificação sem perdas, para o modo de intra-previsão 18 ou modo de intra-previsão com direções de previsão próximas à direção de previsão inferior-esquerda diagonal do modo de intra-previsão 18 (ver FIG. 3), o residual pode ser modificado da maneira a seguir. Primeiro, o residual  $r_{i,j}$ , onde  $0 \leq i \leq (M-1)$  e  $0 \leq j \leq (N-1)$ , pode ser calculado como especificado na HEVC Working Draft 10. Ou seja, a previsão pode ser realizada de acordo com o modo angular e a previsão pode ser subtraída a partir dos valores de amostra originais para ter o residual  $r_{i,j}$ .

Então, uma matriz  $M \times N$  modificado  $\tilde{R}$  com elementos  $\tilde{r}_{i,j}$  pode ser determinado como segue:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j=0 \\ r_{i,j} - r_{i-1,j-1}, & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} . \quad (31)$$

**[0123]** Neste segundo exemplo, as amostras residuais modificadas de  $\tilde{R}$  são sinalizadas para o decodificador (ex., o decodificador de vídeo 30). No lado do decodificador, o decodificador de vídeo pode reconstruir as amostras residuais originais após o decodificador de vídeo analisar as amostras residuais modificadas como segue. Quando a direção de intra-previsão está próxima à direção de previsão inferior-esquerda diagonal, o decodificador de vídeo pode reconstruir as amostras residuais originais como:

$$r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 1 \leq i \leq (M-1), j=0 \\ \tilde{r}_{i,j} + r_{i-1,j-1}, & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} . \quad (32)$$

**[0124]** Além disso, neste segundo exemplo, quando o residual  $r_{i,j}$  tiver sido calculado,  $r_{i,j}$  pode ser adicionado à previsão realizada de acordo com o modo angular para obter as amostras originais. No cálculo o residual  $r_{i,j}$ ,  $r_{i-1,j-1}$  pode precisar estar disponível. Isto pode sempre ser verdadeiro se os residuais verdadeiros são calculados fileira por fileira ou coluna por coluna. Assim, pode ser possível calcular um residual verdadeiro

$r_{i,j}$  para todas as amostras em uma fileira (ou coluna) em paralelo.

**[0125]** Em um terceiro exemplo onde as técnicas de DPCM residuais são estendidas para os modos de intra-previsão angulares para codificação sem perdas, para o modo de intra-previsão 34 ou modos de intra-previsão com direções de previsão próximas à direção de previsão inferior esquerda diagonal do modo de intra-previsão 34 (ver FIG. 3), o residual pode ser modificado da maneira a seguir. Primeiro, o residual  $r_{i,j}$ , onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ , pode ser calculado como especificado na HEVC (ex., HEVC Working Draft 10). Ou seja, a previsão pode ser realizada de acordo com o modo angular e a previsão pode ser subtraída a partir dos valores de amostra originais para determinar o residual  $r_{i,j}$ . Então, uma matriz  $M \times N$  modificado  $\tilde{R}$  com elementos  $\tilde{r}_{i,j}$  pode ser determinado como segue:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j=(N-1) \\ r_{i,j} - r_{i-1,j+1}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-2) \end{cases} \quad (33)$$

**[0126]** Além disso, no terceiro exemplo, as amostras residuais modificadas de  $\tilde{R}$  podem ser sinalizadas para um decodificador de vídeo (ex., o decodificador de vídeo 30). No lado do decodificador, o decodificador de vídeo pode reconstruir as amostras residuais originais após o decodificador de vídeo analisar as amostras residuais modificadas como segue. Quando a direção de intra-previsão é próxima à direção de previsão inferior esquerda diagonal

(modo 34), o decodificador de vídeo pode reconstruir as amostras residuais originais usando a seguinte equação:

$$r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 1 \leq i \leq (M-1), j=(N-1) \\ \tilde{r}_{i,j} + r_{i-1,j+1}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-2) \end{cases} \quad (34)$$

**[0127]** No terceiro exemplo, quando o residual  $r_{i,j}$  foi calculado,  $r_{i,j}$  pode ser adicionado à previsão realizada de acordo com o modo angular para obter as amostras originais. No cálculo o residual  $r_{i,j}$ ,  $r_{i-1,j+1}$  pode precisar estar disponível. Isto pode sempre ser verdadeiro se os residuais verdadeiros são calculados fileira por fileira. Assim, pode ser possível calcular um residual verdadeiro  $r_{i,j}$  para todas as amostras em uma fileira em paralelo.

**[0128]** Em um quarto exemplo onde as técnicas de DPCM residuais são estendidas para modos de intra-previsão angulares para codificação sem perdas, para o modo de intra-previsão 2 ou modos de intra-previsão com direções de previsão próximas à direção de previsão vertical diagonal do modo 2 (ver FIG. 3), o residual pode ser modificado da maneira a seguir. Primeiro, o residual  $r_{i,j}$ , onde  $0 \leq i \leq (M-1)$  e  $0 \leq j \leq (N-1)$  pode ser calculado como especificado na HEVC (ex., HEVC Working Draft 10). Ou seja, a previsão pode ser realizada de acordo com o modo angular e a previsão pode ser subtraída a partir dos valores de amostra originais para determinar o residual  $r_{i,j}$ . Então, uma matriz  $M \times N$  modificado  $\tilde{R}$  com elementos  $\tilde{r}_{i,j}$  pode ser determinado como segue:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=(M-1), 0 \leq j \leq (N-1) \\ r_{i,j}, & 0 \leq i \leq (M-2), j=0 \\ r_{i,j} - r_{i+1,j-1}, & 0 \leq i \leq (M-2), 1 \leq j \leq (N-1) \end{cases} . \quad (35)$$

**[0129]** Neste quarto exemplo, as amostras residuais modificadas de  $\tilde{R}$  podem ser sinalizadas para um decodificador de vídeo (ex., o decodificador de vídeo 30). No lado do decodificador, o decodificador de vídeo pode reconstruir as amostras residuais originais após o decodificador de vídeo analisar as amostras residuais modificadas como segue. Quando a direção de intra-previsão é próxima à direção de previsão vertical diagonal (modo 2), o decodificador de vídeo pode reconstruir as amostras residuais originais de acordo com a seguinte equação:

$$r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i=(M-1), 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 0 \leq i \leq (M-2), j=0 \\ \tilde{r}_{i,j} + r_{i+1,j-1}, & 0 \leq i \leq (M-2), 1 \leq j \leq (N-1) \end{cases} . \quad (36)$$

**[0130]** Além disso, neste quarto exemplo, quando o residual  $r_{i,j}$  foi calculado, o residual  $r_{i,j}$  pode ser adicionado à previsão realizada de acordo com o modo angular para obter as amostras originais. No cálculo o residual  $r_{i,j}$ ,  $r_{i+1,j-1}$  pode precisar estar disponível. Isto pode sempre ser verdadeiro se os residuais verdadeiros são calculados coluna por coluna. Assim, pode ser possível calcular um residual verdadeiro  $r_{i,j}$  para todas as amostras em uma coluna em paralelo. Os primeiro, segundo, terceiro e quarto exemplos onde as técnicas de DPCM residuais são estendidas para modos de intra-previsão angulares para codificação sem perdas podem ser empregados simultaneamente

visto que a faixa de modos de previsão para cada modalidade não se sobrepõe.

**[0131]** Como descrito em um quinto exemplo onde as técnicas de DPCM residuais são estendidas para modos de intra-previsão angulares para a codificação sem perdas, no método de DPCM residual proposto em JCTVC-L0117, para o modo vertical, o residual para a primeira fileira não é modificado. Uma observação similar é verdadeira para o modo horizontal. Nesse caso, o residual para uma primeira coluna não é modificado. Uma observação similar é verdadeira para os modos quase verticais e quase horizontais no caso de um primeiro exemplo onde as técnicas de DPCM residuais são estendidas para modos de intra-previsão angulares para codificação sem perdas. O quinto exemplo onde técnicas de DPCM residuais são estendidas para modos de intra-previsão angulares para codificação sem perdas propõe estender o conceito de DPCM residual para a primeira fileira para um modo de intra-previsão vertical ou quase vertical e para uma primeira coluna para um modo de intra-previsão horizontal ou quase horizontal. Considere um modo vertical ou quase vertical. Neste caso, o primeiro, o residual  $r_{i,j}$ , onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$  pode ser calculado como especificado na HEVC (ex., HEVC Working Draft 10). Ou seja, a previsão pode ser realizada de acordo com o modo angular e a previsão pode ser subtraída a partir dos valores de amostra originais para determinar o residual  $r_{i,j}$ . Então, uma matriz  $M \times N$  modificado  $\tilde{R}$  com elementos  $\tilde{r}_{i,j}$  pode ser determinado como segue:

$$\tilde{r}_{i,j} = r_{i,j} - r_{(i-1),j}, \quad 0 \leq i \leq (M - 1), 0 \leq j \leq (N - 1). \quad (37)$$

Na equação (37),  $r_{-1,j}$  refere-se ao residual a partir do bloco superior. Se o bloco superior não está disponível ou se o bloco superior pertence a uma LCU diferente, pode não ser possível realizar a DPCM residual na primeira fileira.

**[0132]** Neste quinto exemplo, as amostras residuais modificadas de  $\tilde{R}$  podem ser sinalizadas para o decodificador de vídeo. No lado do decodificador, o decodificador de vídeo pode reconstruir as amostras residuais originais após o decodificador de vídeo analisar as amostras residuais modificadas como segue. Quando o modo de intra-previsão é vertical ou quase-vertical, as amostras residuais originais podem ser reconstruídas como segue:

$$r_{i,j} = r_{-1,j} + \sum_{k=0}^i \tilde{r}_{k,j}, \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1). \quad (38)$$

Uma estratégia similar pode ser usada para os modos horizontais ou quase horizontais onde:

$$\tilde{r}_{i,j} = r_{i,j} - r_{i,(j-1)}, \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1), \quad \text{e} \quad (39)$$

$$r_{i,j} = r_{i,-1} + \sum_{k=0}^j \tilde{r}_{i,k}, \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1). \quad (40)$$

Esta abordagem pode ser estendida de um modo similar a um segundo, terceiro, e quarto exemplos onde as técnicas de DPCM residuais são estendidas para modos de intra-previsão angulares para codificação sem perdas como descritos acima.

**[0133]** Em um sexto exemplo onde as técnicas de DPCM residuais são estendidas para modos de intra-previsão angulares para codificação sem perdas, o método de DPCM

residual proposto em JCTVC-L0117 também pode ser aplicado ao modo intra DC (ex., modo 1 em HEVC) e ao modo plano (ex., modo 0 em HEVC). Por exemplo, a previsão de resíduo vertical (ou horizontal) pode ser aplicada após a intra-previsão DC ser feita. Além disso, a previsão de resíduo vertical e horizontal podem ambas ser aplicadas: primeiro aplicar a DCPM vertical (horizontal) e depois aplicar a DPCM horizontal (vertical) para a saída de uma primeira DCPM.

**[0134]** Um sétimo exemplo onde as técnicas de DPCM residuais são estendidas para modos de intra-previsão angulares para codificação sem perdas é similar ao sexto exemplo onde as técnicas de DPCM residuais são estendidas para modos de intra-previsão angulares para codificação sem perdas pelo fato de que duas ou mais DPCMs podem ser aplicadas no resíduo. Por exemplo, as duas DPCMs diagonais descritas nos terceiro e quarto exemplos onde as técnicas de DPCM residuais são estendidas para modos de intra-previsão angulares para codificação sem perdas para o modo plano, e então a DPCM horizontal e vertical.

**[0135]** Várias técnicas descritas nesta revelação também podem ser estendidas para intra codificação com perda quando a transformada é pulada. Por exemplo, na HEVC (ex., HEVC Working Draft 10), a transformada pode ser pulada para os blocos 4×4. Nesta revelação, o termo "bloco de pulo de transformada" se aplica a um bloco para o qual a aplicação da transformada é pulada. No entanto, as técnicas propostas descritas acima onde as técnicas de DPCM residuais são estendidas para modos de intra-previsão angulares para codificação sem perdas podem ser usadas para tamanhos de bloco mais altos se a transformada é pulada. Por exemplo, JCTVC-E145 discutiu uma maneira de estender essas técnicas para a



intra codificação com perda quando a transformada é pulada. Em geral, a ideia proposta em JCTVC-E145 é que ao invés de usar as amostras de referência para intra-previsão como mostrado na FIG. 2, os valores de amostra reconstruídos a partir de vizinhos causais podem ser usados para realizar a intra-previsão. No entanto, isto pode ser dispendioso computacionalmente conforme o processo de previsão tem de ser repetido para cada amostra. Além disso, não é claro como isto poderia funcionar para DC e modos de intra-previsão planos.

**[0136]** Uma ou mais técnicas desta revelação são relacionadas à DPCM residual para codificação com perda. Por exemplo, em alguns exemplos desta revelação, os modos de intra-previsão quase horizontais e quase verticais para um bloco para o qual a transformada é pulada são considerados. Nesta revelação, um modo quase vertical pode ser definido como um modo de intra-previsão onde a direção de previsão é quase vertical. Exemplos de modos quase verticais podem ser todos os modos de intra-previsão entre 22 e 30 como mostrado na FIG. 3. De modo similar, um modo quase horizontal pode ser definido como um modo de intra-previsão onde a direção de previsão é quase horizontal. Exemplos de modos de intra-previsão quase horizontais podem ser todos os modos de intra-previsão entre 6 e 14 como mostrado na FIG. 3. Além disso, considere um bloco de tamanho  $M$  (fileiras)  $\times$   $N$  (colunas). Deixe  $r_{i,j}$ , onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ , ser a previsão residual após a realização da intra-previsão como especificado na HEVC (ex., HEVC Working Draft 10). Isto é mostrado nas FIGS. 5A e 5B. O bloco pode representar qualquer componente (ex., luma, croma, R, G, B, etc.).

**[0137]** Especificamente, a FIG. 5A mostra uma direção de DPCM residual para os modos quase verticais. A

FIG. 5B mostra uma direção de DPCM residual para os modos quase horizontais. Cada respectivo quadrado nas FIG. 5A e 5B corresponde a uma respectiva amostra residual  $r_{i,j}$ . As matrizes verticais em cada coluna da FIG. 5A mostram a direção da DPCM residual para os modos quase verticais. As matrizes horizontais em cada fileira da FIG. 5B mostram a direção da DPCM residual para os modos quase horizontais. Uma direção de DPCM é uma direção (ex., horizontal ou vertical) junto com um codificador de vídeo que processa as amostras quando aplica-se a DPCM residual.

**[0138]** Além disso, considere um modo quase vertical e deixe  $Q(r_{i,j})$ , onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ , denotar a versão quantificada do residual  $r_{i,j}$ . O bloco de valores residuais pode ser denotado como  $R$ . Em outras palavras, residual  $r_{i,j}$  sofreu quantificação e quantificação inversa. A DPCM residual pode ser aplicadas às amostras residuais para obter uma matriz  $M \times N \tilde{R}$ , como segue. A matriz  $M \times N$  modificado  $\tilde{R}$  com elementos  $\tilde{r}_{i,j}$  pode ser obtido como segue quando o modo de intra-previsão é vertical:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N-1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases} \quad (41)$$

Quando o modo de intra-previsão é horizontal,  $\tilde{r}_{i,j}$  podem ser obtido como:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j = 0 \\ r_{i,j} - Q(r_{i,(j-1)}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad (42)$$

**[0139]** Assim, no exemplo acima, quando o modo de intra-previsão é vertical ou quase vertical, os valores residuais modificados na coluna mais à esquerda de  $\tilde{R}$  são iguais aos valores residuais correspondentes na coluna mais à esquerda de  $R$ . Para cada respectivo valor residual (i.e.,  $r_{i,j}$ ) em cada coluna de  $R$  para a direita da coluna mais à esquerda de  $R$ , o codificador de vídeo 20 pode definir um valor residual modificado correspondente  $\tilde{r}_{i,j}$  igual ao respectivo valor residual (i.e.,  $r_{i,j}$ ) menos um valor residual reconstruído correspondente a um valor residual que ocorre imediatamente à esquerda da respectivo valor residual (i.e.,  $Q(r_{(i-1),j})$ ). De modo similar, quando o modo de intra-previsão é horizontal ou quase horizontal, os valores residuais modificados na fileira superior de  $\tilde{R}$  são iguais aos valores residuais correspondentes na fileira superior de  $R$ . No entanto, para cada respectivo valor residual (i.e.,  $r_{i,j}$ ) em cada fileira de  $R$  abaixo da fileira superior de  $R$ , o codificador de vídeo 20 pode definir o valor residual modificado correspondente (i.e.,  $\tilde{r}_{i,j}$ ) igual ao respectivo valor residual (i.e.,  $r_{i,j}$ ) menos um valor residual reconstruído correspondente a um valor residual que ocorre imediatamente acima do respectivo valor residual (i.e.,  $Q(r_{i,(j-1)})$ ).

**[0140]** Além disso, neste exemplo, a amostra residual modificada,  $\tilde{r}_{i,j}$  é quantificada para produzir uma versão quantificada da amostra residual modificada,  $Q(\tilde{r}_{i,j})$ . Quando o modo de intra-previsão é vertical, a amostra residual reconstruída  $Q(r_{i,j})$  é calculada como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases} \quad (43)$$

Isto também podem ser escrito como:

$$Q(r_{i,j}) = \sum_{k=0}^i Q(\tilde{r}_{k,j}), \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1) \cdot$$

De modo similar, quando o modo é horizontal, a amostra residual reconstruída  $Q(r_{i,j})$  é calculada como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad (44)$$

Isto também pode ser escrito como:

$$Q(r_{i,j}) = \sum_{k=0}^j Q(\tilde{r}_{i,k}), \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1) \cdot$$

**[0141]** Assim, quando o modo de intra-previsão é vertical ou quase vertical, os valores residuais reconstruídos correspondentes a um valores residuais em uma coluna mais à esquerda de  $R$  são iguais às amostras desquantificadas correspondentes de valores residuais modificados na coluna mais à esquerda de  $\tilde{R}$ . No entanto, para cada respectivo valor residual em cada coluna de  $R$  para a direita da coluna mais a esquerda de  $R$ , o codificador de vídeo 20 define o valor residual reconstruído correspondente igual a soma de uma versão quantificada do valor residual modificado correspondente em  $\tilde{R}$  (i.e.,  $Q(\tilde{r}_{i,j})$ ) e o valor residual reconstruído correspondente ao valor residual que ocorre imediatamente à esquerda do respectivo valor residual (i.e.,  $Q(r_{i-1,j})$ ). De modo similar, quando o modo de intra-previsão é horizontal ou quase horizontal, os valores residuais reconstruídos

correspondentes a valores residuais em uma fileira superior de  $R$  são iguais às amostras desquantificadas correspondentes de valores residuais modificados em uma fileira superior de  $\tilde{R}$ . No entanto, para cada respectivo valor residual de  $R$  em cada fileira de  $R$  abaixo da fileira superior de  $R$ , o codificador de vídeo 20 pode definir o valor residual reconstruído correspondente igual a uma soma da versão quantificada do valor residual modificado correspondente no  $\tilde{R}$  (i.e.,  $Q(\tilde{r}_{i,j})$ ) e um valor residual reconstruído correspondente ao valor residual em  $R$  que ocorre imediatamente acima do respectivo valor residual (i.e.,  $Q(r_{i,j-1})$ ).

**[0142]** No lado do decodificador, os cálculos (43), e (44) são repetidos para produzir a amostra residual reconstruída  $Q(r_{i,j})$ , onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ . O decodificador de vídeo 30 pode adicionar valores reconstruídos de amostra residuais aos valores originais de previsão para produzir valores de amostra reconstruídos. Por exemplo, o decodificador de vídeo 30 pode adicionar a amostra residual reconstruída  $Q(r_{i,j})$  à amostra correspondente de um bloco preditivo para reconstruir valores de amostra de um bloco atual (ex., uma CU atual). O processo de determinação dos valores reconstruídos residuais a partir dos valores residuais que são codificados usando DPCM pode ser referido aqui como "RDPCM inversa". Normalmente, ambos codificador de vídeo 20 e decodificador de vídeo 30 fixam as amostras reconstruídas para uma profundidade de bit adequada. No entanto, neste caso, a operação de fixação é realizada após o processo de RDPCM inverso para reconstruir os residuais quantificados.

**[0143]** Os exemplos acima podem ser estendidos de uma maneira direta para as direções diagonais também.

Por exemplo, em um exemplo, quando a direção de intra-previsão é próxima a uma direção descendente para a direita diagonal (ex., modo de intra-previsão 18),  $\tilde{r}_{i,j}$  pode ser determinada como:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j=0 \\ r_{i,j} - Q(r_{i-1,j-1}), & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad (45)$$

A amostra residual reconstruída  $Q(r_{i,j})$  pode ser determinada como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1) \\ Q(\tilde{r}_{i,j}), & 1 \leq i \leq (M-1), j=0 \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j-1}), & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} . \quad (46)$$

Outros modos quase diagonais (ex., modos quase para baixo à esquerda (modo 34) e quase para cima à direita (modo 2)) podem ser tratados de modo similar.

**[0144]** Além disso, em alguns exemplos, ao invés de aplicar DPCM aos residuais reconstruídos, a DPCM pode ser aplicada ao deslocamento das versões com bits dos residuais reconstruídos. Por exemplo, para os blocos de pulso de transformada, o decodificador de vídeo 30 pode realizar, para cada amostra, uma operação de desquantificação, um deslocamento para a esquerda por 7, e depois um deslocamento para a direita (ex., após a adição de um deslocamento) por (20 - profundidade de bit). Neste contexto, a profundidade de bit pode se referir ao número de bits usado para representar um valor de amostra ou um coeficiente de transformada. Em uma operação de

desquantificação, há um deslocamento para a direita por `bdShift` após a adição de um deslocamento.

**[0145]** Por exemplo, a seção 8.6.2 de JCTVC-M1005\_v2 (i.e., um esboço da especificação de extensão de faixa para HEVC) descreve um processo de dimensionamento e transformação que determina uma matriz das amostras residuais baseado em uma matriz de coeficiente de níveis de transformada de um bloco de transformada de uma TU. Na seção 8.6.2 de JCTVC-M1005\_v2, a matriz de coeficiente de níveis de transformada pode conter valores de amostra residuais se a transformada e quantificação não é aplicada ao bloco transformado. Uma porção da seção 8.6.2 de JCTVC-M1005\_v2 é reproduzida abaixo.

A matriz de amostras residuais  $(nTbS) \times (nTbS)$  `r` é derivada como segue:

-Se `cu_transquant_bypass_flag` é igual a 1, a matriz  $(nTbS) \times (nTbS)$  `r` é definida igual a matriz de coeficientes de transformada  $(nTbS) \times (nTbS)$  `TransCoeffLevel[ xTbY ][ yTbY ][ cIdx ]`.

-Caso contrário, as seguintes etapas ordenadas se aplicam:

1. O processo de dimensionamento para os coeficientes de transformada como especificado na sub-cláusula 8.6.3 é invocado com um local de bloco transformado  $(xTbY, yTbY)$ , o tamanho de um bloco transformado `nTbS`, a variável do componente de cor `cIdx`, e o parâmetro de quantificação `qP` como entradas, e a saída é uma matriz  $(nTbS) \times (nTbS)$  de coeficientes de transformada dimensionados `d`.

2. A matriz  $(nTbS) \times (nTbS)$  de amostras residuais `r` é derivada como segue:

-Se `transform_skip_flag[ xTbY ][ yTbY ][ cIdx ]` é igual a 1, os valores da matriz residual da amostra `r[ x ][ y ]` com `x = 0..nTbS - 1`, `y = 0..nTbS - 1` são derivados como segue:

$$r[ x ][ y ] = ( d[ x ][ y ] \ll 7 ) \quad (8-267)$$

-Caso contrário  
(`transform_skip_flag[ xTbY ][ yTbY ][ cIdx ]` é igual a 0), um processo de transformação para coeficientes de transformada dimensionados como especificado na sub-cláusula 8.6.4 é invocado com um local de bloco transformado `( xTbY, yTbY )`, o tamanho de um bloco transformado `nTbS`, a variável do componente de cor `cIdx`, e a matriz `(nTbS)x(nTbS)` de coeficientes de transformada dimensionados `d` como entradas, e a saída é uma matriz `(nTbS)x(nTbS)` de amostras residuais `r`.

3. O `bdShift` variável é derivado como segue:

$$\text{bdShift} = ( \text{cIdx} == 0 ) ? 20 - \text{BitDepth}_y : 20 - \text{BitDepth}_c \quad (8-268)$$

4. Os valores de amostra residuais `r[ x ][ y ]` com `x = 0..nTbS - 1`, `y = 0..nTbS - 1` são modificados como segue:

$$r[ x ][ y ] = ( r[ x ][ y ] + ( 1 \ll ( \text{bdShift} - 1 ) ) ) \gg \text{bdShift} \quad (8-269)$$

**[0146]** Como mostrado acima, a seção 8.6.2 de JCTVC-M1005\_v2 especifica que a seção 8.6.3 de JCTVC-M1005\_v2 é invocada se a transformada e quantificação são aplicadas a um bloco transformado. A seção 8.6.3 de JCTVC-M1005\_v2 descreve um processo de dimensionamento para coeficientes de transformada. Seção 8.6.3 de JCTVC-M1005\_v2 é reproduzida abaixo.



As entradas para esse processo são:

- um local luma ( xTbY, yTbY ) que especifica a amostra superior à esquerda do bloco de transformada luma atual em relação a uma amostra superior à esquerda luma da imagem atual,
- uma variável nTbS que especifica o tamanho do bloco de transformada atual,
- uma variável cIdx que especifica o componente de cor do bloco atual,
- uma variável qP que especifica o parâmetro de quantificação.

A saída deste processo é a matriz (nTbS)x(nTbS) d de coeficientes de transformada dimensionados com elementos d[ x ][ y ].

A variável bdShift é derivada como segue:

- Se cIdx é igual a 0,  

$$\text{bdShift} = \text{BitDepth}_Y + \text{Log2}( \text{nTbS} ) - 5$$
(8-270)

- Caso contrário,  

$$\text{bdShift} = \text{BitDepth}_C + \text{Log2}( \text{nTbS} ) - 5$$
(8-271)

A lista de levelScale[ ] é especificada como levelScale[ k ] = { 40, 45, 51, 57, 64, 72 } com k = 0..5.

Para a derivação dos coeficientes de transformada dimensionados d[ x ][ y ] com x = 0..nTbS - 1, y = 0..nTbS - 1, o seguinte se aplica:

- O fator de dimensionamento m[ x ][ y ] é derivado como segue:
  - Se scaling\_list\_enabled\_flag é igual a 0,  

$$m[ x ][ y ] = 16$$
(8-272)
  - Caso contrário (scaling\_list\_enabled\_flag é igual a 1),

$$m[x][y] = \text{ScalingFactor}[ \text{sizeId} ][ \text{matrixId} ][x][y] \quad (8-273)$$

Onde sizeId é especificado na Tabela 7-3 para o tamanho da matriz de quantificação igual a (nTbS)x(nTbS) e matrixId é especificado na Tabela 7-4 para sizeId, CuPredMode[ xTbY ][ yTbY ], e cIdx, respectivamente.

- O coeficiente de transformada dimensionado d[ x ][ y ] é derivado como segue:

$$d[x][y] = \text{Clip3}( -32768, 32767, ( ( \text{TransCoeffLevel}[ xTbY ][ yTbY ][ cIdx ][x][y] * m[x][y] * \text{levelScale}[ qP\%6 ] \ll (qP / 6) ) + ( 1 \ll ( \text{bdShift} - 1 ) ) ) \gg \text{bdShift} ) \quad (8-274)$$

**[0147]** Como indicado na seção 8.6.3 de JCTVC-M1005\_v2, o decodificador de vídeo 30 pode determinar um coeficiente de transformada dimensionado, d[x][y], usando a equação 8-274 de JCTVC-M1005\_v2. Na equação 8-274, a função Clip3(...) é definida como:

$$\text{Clip3}( x, y, z ) = \begin{cases} x & ; z < x \\ y & ; z > y \\ z & ; \text{otherwise} \end{cases}.$$

**[0148]** De acordo com uma ou mais técnicas desta revelação, ao invés de aplicar a DPCM inversa aos residuais reconstruídos, a DPCM inversa pode ser aplicada aos valores de amostra desquantificados antes de aplicar o deslocamento para a esquerda. De modo similar, ao invés de uma transformada inversa, um deslocamento para a esquerda por 7 pode ser aplicado após a desquantificação. Por exemplo, uma matriz de amostras residuais r[x][y] com x =

$0..nTbS-1$ ,  $y = 0..nTbS-1$  pode ser derivada como mostrado na equação 8-267 de JCTVC-M1005\_v2.

**[0149]** Em alguns exemplos, a DPCM inversa é aplicada às amostras após a desquantificação ou após um deslocamento para a esquerda por 7. De modo similar, o deslocamento para a direita por  $(20 - \text{profundidade de bit})$  após a adição de um deslocamento é aplicada bem no final do processo para reconstruir os residuais. Por exemplo, as amostras residuais  $r[x][y]$ , com  $x = 0..nTbS-1$  e  $y = 0..nTbS-1$ , podem ser modificadas como mostrado na equação 8-269 de JCTVC-M1005\_v2.

Nesse caso, a DPCM inversa pode ser aplicada após um deslocamento para a direita por uma quantidade menor que  $(20 - \text{profundidade de bit})$  para manter a melhor precisão, antes de aplicar o deslocamento restante para a direita de modo que o deslocamento total para a direita equivale a  $(20 - \text{profundidade de bit})$ .

**[0150]** Em alguns casos, o codificador de vídeo 20 não aplica uma transformada às amostras residuais de um bloco de transformada, mas quantifica as amostras residuais de um bloco transformado. Em outras palavras, o codificador de vídeo 20 pode aplicar uma forma de codificação com perda na qual a transformada é pulada. Em tais casos, o decodificador de vídeo 30 pode determinar, a partir dos elementos de sintaxe em um bitstream, amostras quantificadas residuais de um bloco transformado e desquantificar as amostras quantificadas residuais de um bloco transformado para reconstruir as amostras residuais de um bloco transformado sem aplicar uma transformada inversa a um bloco transformado.

**[0151]** Alternativamente, a filtragem de gradiente pode ser modificada como descrito acima usando valores de amostra reconstruídos (quantificados) ao invés

de valores de amostra originais. Na HEVC (ex., HEVC Working Draft 10), quando obtém-se a previsão para os modos horizontais e verticais, um termo gradiente é adicionado à previsão para a primeira coluna (para o modo vertical) ou a primeira fileira (para o modo horizontal). Isto pode ser referido como filtragem de gradiente. A adição do termo gradiente poderia ser habilitada ou desabilitada quando usa-se a DPCM residual. Esta revelação propõe melhorias para a filtragem de gradiente para os modos de intra-previsão horizontais e verticais quando a previsão residual é usada.

**[0152]** Por exemplo, considere que a DPCM residual como proposta em JCTVC-L0117 está sendo usada. Em HEVC Working Draft 10, para componentes luma e para tamanhos de bloco menores que 32, para o modo de intra-previsão vertical, a previsão para a amostra  $P_{i,0}, 0 \leq i \leq M-1$  é dada por:

$$\text{Fixar } \left( P_{-1,0} + \left( (P_{i,-1} - P_{-1,-1}) \gg 1 \right) \right),$$

Na equação acima,  $\gg$  representa um deslocamento para a direita de bit por bit e a operação de fixação fixa os valores de previsão à faixa de valores de amostra. Porque na DPCM residual, para o modo de intra-previsão vertical, a previsão para a amostra  $P_{i,j}$  é  $P_{i-1,j}$ , o termo gradiente para as amostras na primeira coluna é modificado como  $\left( (P_{i,-1} - P_{-1,-1}) \gg 1 \right)$ . Dessa forma, a previsão para a amostra  $P_{i,0}, 0 \leq i \leq M-1$  é:

$$\text{Fixar } \left( P_{i-1,0} + \left( (P_{i,-1} - P_{-1,-1}) \gg 1 \right) \right).$$

**[0153]** Além disso, a filtragem de gradiente pode ser estendida para outras colunas para o modo de intra-previsão vertical como segue. Porque as amostras originais da esquerda e esquerda superiores estão

disponíveis no modo sem perdas, o termo gradiente podem ser adicionado à previsão para qualquer coluna no modo de intra-previsão vertical. Assim, para o modo de intra-previsão vertical, a previsão para a amostra  $P_{i,j}, 0 \leq i \leq M-1, 0 \leq j \leq N-1$  pode ser modificada para:

$$\text{Fixar} \left( P_{i-1,j} + \left( (P_{i,j-1} - P_{i-1,j-1}) \gg 1 \right) \right).$$

A filtragem modificada de gradiente para o modo de intra-previsão vertical pode ser aplicada à qualquer componente e qualquer tamanho de bloco.

**[0154]** Este conceito pode ser estendido para o modo de intra-previsão horizontal também. Em HEVC Working Draft 10, para componentes luma e para tamanhos de bloco menores que 32, para o modo de intra-previsão horizontal, a previsão para a amostra  $P_{0,j}, 0 \leq j \leq N-1$  é dada por:

$$\text{Fixar} \left( P_{0,-1} + \left( (P_{-1,j} - P_{-1,-1}) \gg 1 \right) \right),$$

Na equação acima,  $\gg$  representa um deslocamento para a direita de bit por bit e a operação de fixação fixa os valores de previsão à faixa de valores de amostra. Porque na DPCM residual, para o modo de intra-previsão horizontal, a previsão para a amostra  $P_{i,j}$  é  $P_{i,j-1}$ , o termo gradiente para as amostras na primeira fileira pode ser modificado para  $\left( (P_{-1,j} - P_{-1,j-1}) \gg 1 \right)$ . A previsão para a amostra  $P_{0,j}, 0 \leq j \leq N-1$  pode ser determinada por:

$$\text{Fixar} \left( P_{0,j-1} + \left( (P_{-1,j} - P_{-1,j-1}) \gg 1 \right) \right).$$

**[0155]** A filtragem de gradiente pode ser estendida para outras colunas para o modo de intra-previsão horizontal como segue. Porque as amostras originais superiores e superiores à esquerda estão sempre disponíveis no modo sem perdas, o termo gradiente pode ser adicionado à previsão para qualquer fileira no modo de intra-previsão

horizontal. Assim, para o modo de intra-previsão horizontal, a previsão para a amostra  $P_{i,j}, 0 \leq i \leq M-1, 0 \leq j \leq N-1$  pode ser dada por:

$$\text{Fixar} \left( P_{i,j-1} + \left( (P_{i-1,j} - P_{i-1,j-1}) \gg 1 \right) \right).$$

**[0156]** Como descrito na seção 7.3.2.3 de JCTVC-M1005\_v2, um PPS pode incluir um elemento de sintaxe habilitado para a ocultação de dados de sinal (ex., `sign_data_hiding_enabled_flag`). Além disso, como mostrado na seção 7.3.8.11 de JCTVC-M1005\_v2, quando o elemento de sintaxe habilitado para a ocultação de dados de sinal tem um valor particular (ex., 1), os elementos de sintaxe (ex., `coeff_sign_flag` elementos de sintaxe) que indicam os sinais positivo/negativo dos coeficientes de transformada podem ser omitidos de uma TU. Assim, o codificador de vídeo 20 pode sinalizar, em um bitstream, uma indicação explícita (ex., uma `sign_data_hiding_enabled_flag`) de que a ocultação de dados de sinal é habilitada para uma imagem atual e consequentemente a um bloco atual dentro da imagem atual. Da mesma forma, o decodificador de vídeo 30 pode obter, a partir de um bitstream, uma indicação explícita (ex., uma `sign_data_hiding_enabled_flag`) de que a ocultação de dados de sinal é habilitada para uma imagem atual e consequentemente um bloco atual na imagem atual. Seção 7.3.8.11 de JCTVC-M1005\_v2 é reproduzida abaixo.

residual_codificação( x0, y0, log2TrafoSize, cIdx ) {	escritor
se( transform_skip_enabled_flag && !cu_transquant_bypass_flag && ( log2TrafoSize == 2 ) )	
transform_skip_flag[ x0 ][ y0 ][ cIdx ]	e(v)
last_sig_coeff_x_prefix	e(v)
last_sig_coeff_y_prefix	e(v)
se( last_sig_coeff_x_prefix > 3 )	
last_sig_coeff_x_suffix	e(v)
se( last_sig_coeff_y_prefix > 3 )	
last_sig_coeff_y_suffix	e(v)
lastScanPos = 16	
lastSubBlock = ( 1 << ( log2TrafoSize - 2 ) ) * ( 1 << ( log2TrafoSize - 2 ) ) - 1	
fazer {	
se( lastScanPos == 0 ) {	
lastScanPos = 16	
lastSubBlock--	
}	
lastScanPos--	
xS = ScanOrder[ log2TrafoSize - 2 ][ scanIdx ][ lastSubBlock ][ 0 ]	
yS = ScanOrder[ log2TrafoSize - 2 ][ scanIdx ][ lastSubBlock ][ 1 ]	

<code>xC = ( xS &lt;&lt; 2 ) + ScanOrder[ 2 ][ scanIdx ][ lastScanPos ][ 0 ]</code>	
<code>yC = ( yS &lt;&lt; 2 ) + ScanOrder[ 2 ][ scanIdx ][ lastScanPos ][ 1 ]</code>	
<code>} enquanto( ( xC != LastSignificantCoeffX )    ( yC != LastSignificantCoeffY ) )</code>	
<code>para( i = lastSubBlock; i &gt;= 0; i-- )</code>	
<code>{</code>	
<code>    xS = ScanOrder[ log2TrafoSize - 2 ][ scanIdx ][ i ][ 0 ]</code>	
<code>    yS = ScanOrder[ log2TrafoSize - 2 ][ scanIdx ][ i ][ 1 ]</code>	
<code>    inferSbDcSigCoeffFlag = 0</code>	
<code>    se( ( i &lt; lastSubBlock ) &amp;&amp; ( i &gt; 0 ) ) {</code>	
<code>        coded_sub_block_flag[ xS ][ yS ]</code>	<code>e(v)</code>
<code>        inferSbDcSigCoeffFlag = 1</code>	
<code>    }</code>	
<code>    para( n = ( i == lastSubBlock ) ? lastScanPos - 1 : 15; n &gt;= 0; n-- ) {</code>	
<code>        xC = ( xS &lt;&lt; 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 0 ]</code>	
<code>        yC = ( yS &lt;&lt; 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 1 ]</code>	
<code>        se( coded_sub_block_flag[ xS ][ yS ] &amp;&amp; ( n &gt; 0    !inferSbDcSigCoeffFlag ) ) {</code>	
<code>            sig_coeff_flag[ xC ][ yC ]</code>	<code>e(v)</code>
<code>        se( sig_coeff_flag[ xC ][ yC ] )</code>	
<code>        inferSbDcSigCoeffFlag = 0</code>	



}	
}	
firstSigScanPos = 16	
lastSigScanPos = -1	
numGreater1Flag = 0	
lastGreater1ScanPos = -1	
para( n = 15; n >= 0; n-- ) {	
xC = ( xS << 2 ) +	
ScanOrder[ 2 ][ scanIdx ][ n ][ 0 ]	
yC = ( yS << 2 ) +	
ScanOrder[ 2 ][ scanIdx ][ n ][ 1 ]	
se( sig_coeff_flag[ xC ][ yC ] ) {	
se( numGreater1Flag < 8 ) {	
coeff_abs_level_greater1_flag[ n ]	e(v)
numGreater1Flag++	
se(	
coeff_abs_level_greater1_flag[ n ]	&&
lastGreater1ScanPos == -1 )	
lastGreater1ScanPos = n	
}	
se( lastSigScanPos == -1 )	
lastSigScanPos = n	
firstSigScanPos = n	
}	
}	
signHidden = ( lastSigScanPos -	
firstSigScanPos > 3	&&
!cu_transquant_bypass_flag )	
se( lastGreater1ScanPos != -1 )	

coeff_abs_level_greater2_flag[ lastGreater1ScanPos ]	e(v)
para( n = 15; n >= 0; n-- ) {	
xC = ( xS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 0 ]	
yC = ( yS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 1 ]	
se( sig_coeff_flag[ xC ][ yC ] && ( !sign_data_hiding_enabled_flag    !signHidden    ( n != firstSigScanPos ) ) )	
coeff_sign_flag[ n ]	e(v)
}	
numSigCoeff = 0	
sumAbsLevel = 0	
para( n = 15; n >= 0; n-- ) {	
xC = ( xS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 0 ]	
yC = ( yS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 1 ]	
se( sig_coeff_flag[ xC ][ yC ] ) {	
baseLevel = 1 + coeff_abs_level_greater1_flag[ n ] + coeff_abs_level_greater2_flag[ n ]	
se( baseLevel == ( ( numSigCoeff < 8 ) ( ( n == lastGreater1ScanPos ) ? 3 : 2 ) : 1 ) )	
coeff_abs_level_remaining[ n ]	e(v)

TransCoeffLevel[ x0 ][ y0 ][ cIdx ][ xC ][ yC ] = ( coeff_abs_level_remaining[ n ] + baseLevel ) * ( 1 - 2 * coeff_sign_flag[ n ] )	
se( sign_data_hiding_enabled_flag && signHidden ) {	
sumAbsLevel += ( coeff_abs_level_remaining[ n ] + baseLevel )	
se( ( n == firstSigScanPos ) && ( ( sumAbsLevel % 2 ) == 1 ) )	
TransCoeffLevel[ x0 ][ y0 ][ cIdx ][ xC ][ yC ] =  -TransCoeffLevel[ x0 ][ y0 ][ cIdx ][ xC ][ yC ]	
}	
numSigCoeff++	
}	
}	
}	
}	

**[0157]** Como especificado na seção 7.4.3.3 de JCTVC-M1005\_v2, sign\_data\_hiding\_enabled\_flag igual a 0 especifica que a ocultação de bit de sinal (i.e., ocultação de dados de sinal) é desabilitada. A sign\_data\_hiding\_enabled\_flag igual a 1 especifica que a ocultação de bit de sinal é habilitada. Esta revelação pode usar os termos “ocultação de bit de sinal” e “ocultação de dados de sinal” indistintamente. Além disso, um codificador de vídeo pode organizar elementos de sintaxe que representam coeficientes de transformada de acordo com uma ordem de varredura de sub-bloco. Na ordem de varredura de sub-bloco, o codificador de vídeo divide um bloco de

coeficiente de transformada em sub-blocos, cada um tendo 4 coeficientes de transformada de largura e 4 coeficientes de transformada de altura. O codificador de vídeo pode determinar uma variável de sinal escondido (i.e., `signHidden`) para cada um dos sub-blocos. Quando a diferença nas posições entre um primeiro coeficiente de transformada significativo (i.e., não-zero) em um sub-bloco 4x4 de um bloco de coeficiente de transformada e um último coeficiente de transformada significativo no sub-bloco 4x4 do bloco de coeficiente de transformada é maior que 3 (i.e., `lastSignScanPos - firstSigScanPos > 3`) e as etapas de transformada e quantificação não são ignoradas (i.e., pulada) para a CU atual (i.e., `!cu_transquant_bypass_flag`), o codificador de vídeo pode determinar que a variável de sinal escondido (i.e., `signHidden`) para o sub-bloco 4x4 é igual a 1. Se a transformada é pulada ou ignorada, os coeficientes de transformada são na verdade residuais de previsão.

**[0158]** Para cada respectivo coeficiente de transformada no sub-bloco 4x4, um elemento de sintaxe de sinal (ex., `coeff_sign_flag[n]`) para o respectivo coeficiente de transformada é sinalizado no bitstream se o respectivo coeficiente de transformada é significativo (i.e., non-zero) e se a `sign_data_hiding_enabled_flag` especifica que a ocultação de bit de sinal não é habilitada, a variável de sinal escondido para o sub-bloco 4x4 não é igual a 1, ou o respectivo coeficiente de transformada não é o primeiro coeficiente de transformada significativo no sub-bloco 4x4. Por exemplo, o sinal de um coeficiente de transformada significativo não é sinalizado se a `sign_data_hiding_enabled_flag` indica que a ocultação de bit de sinal é habilitada, a diferença na posição entre um primeiro coeficiente significativo do sub-

bloco 4x4 e o último coeficiente significativo do sub-bloco 4x4 é maior que 3, o sub-bloco 4x4 não foi gerado usando a codificação de desvio de transformada e quantificação (ex., codificação sem perdas), e o respectivo coeficiente de transformada é o primeiro coeficiente de transformada significativo do sub-bloco 4x4 .

**[0159]** Como especificado na seção 7.4.3.3 de JCTVC-M1005\_v2, `coeff_sign_flag[n]` especifica o sinal de um nível de coeficiente de transformada para a posição de varredura `n` como segue. Se `coeff_sign_flag[n]` é igual a 0, o nível de coeficiente de transformada correspondente tem um valor positivo. Caso contrário (`coeff_sign_flag[n]` é igual a 1), o nível de coeficiente de transformada correspondente tem um valor negativo. Quando `coeff_sign_flag[n]` não está presente, `coeff_sign_flag[n]` é deduzido como igual a 0.

**[0160]** Quando a ocultação de dados de sinal é habilitada, o codificador de vídeo 20 pode inserir dados que indicam o sinal de um coeficiente de transformada/amostra residual no valor do próprio coeficiente de transformada/valor de amostra residual. Por exemplo, o codificador de vídeo 20 pode modificar um ou mais bits que representam um coeficiente de transformada/amostra residual de modo que a informação de paridade pode ser usada pelo decodificador de vídeo 30 para determinar se um primeiro coeficiente significativo em um sub-bloco 4x4 é positivo ou negativo. Por exemplo, se a soma de valores absolutos de coeficientes de transformada em um sub-bloco 4x4 é par, o sinal é deduzido como positivo, caso contrário o sinal é deduzido como negativo. Tipicamente, o codificador de vídeo 20 pode alterar um bit menos significativo da representação feita do coeficiente de transformada/amostras residuais quantificados em um sub-

bloco 4×4. Consequentemente, um usuário pode não ser capaz de perceber qualquer perda de qualidade visual devido ao codificador de vídeo 20 que modifica um bit da representação do coeficiente de transformada/amostra residual para indicar o sinal do coeficiente de transformada/amostra residual.

**[0161]** Ao usar as técnicas descritas desta revelação para codificação com perda, pode ser difícil para um codificador de vídeo implementar a ocultação de dados de sinal para os blocos para os quais as técnicas descritas são aplicadas. Estes são intra-blocos para os quais a transformada é pulada e os modos são planos e/ou DC e/ou os modos para o qual a DPCM residual é aplicada. Os erros podem ser introduzidos pelo codificador de vídeo 20 que altera os valores de bits na representação do coeficiente de transformada/amostra residual para indicar o sinal do coeficiente de transformada/amostra residual. Além disso, no uso destas técnicas, o erro em uma amostra residual modificada pode se propagar para as amostras residuais subsequentes. Tais erros podem ser compostos quando a DPCM residual é aplicada porque a DPCM residual pode depender do decodificador de vídeo 30 adicionar junto vários coeficientes de transformada/amostras residuais para determinar os coeficientes de transformada/amostras residuais. Assim, a ocultação de dados de sinal pode na verdade resultar na degradação do desempenho. Em tais casos, a ocultação de dados de sinal pode ser normativamente desabilitada. Isso significa que mesmo que seja indicado em um bitstream (ou como uma escolha padrão) que a ocultação de dados de sinal está sendo usada, a ocultação de dados de sinal é desabilitada para determinados blocos.

**[0162]** Em um exemplo, um codificador de vídeo pode desabilitar a ocultação de dados de sinal para um bloco se o codificador de vídeo não aplica uma transformada ao bloco, se o bloco é intra previsto usando um modo de intra-previsão plano ou um modo de intra-previsão DC, ou se o bloco é intra previsto usando um modo de intra-previsão para a DPCM residual ser aplicada. Por exemplo, em alguns exemplos, a ocultação de dados de sinal pode ser desabilitada quando a transformada é pulada e o bloco é intra-codificado e o intra modo é a intra-previsão plana. Além disso, em alguns exemplos, a ocultação de dados de sinal pode ser desabilitada quando a transformada é pulada e o bloco é intra-codificado e o intra modo é a intra-previsão DC. Além disso, em alguns exemplos, a ocultação de dados de sinal pode ser desabilitada quando a transformada é pulada e o bloco é intra-codificado e o intra modo é um modo para o qual a DPCM residual é aplicada.

**[0163]** Por exemplo, o codificador de vídeo 20 pode determinar que a ocultação de dados de sinal é desabilitada para um bloco atual se o bloco atual é gerado usando codificação com perda sem a aplicação de uma transformada (ex., uma transformada de cosseno discreta, transformada direcional, ou outra transformada) aos dados residuais e o bloco atual é intra previsto usando um modo de intra-previsão no qual a DPCM residual é usada. Quando a ocultação de dados de sinal é desabilitada para o bloco atual, o codificador de vídeo 20 pode incluir, no bitstream, para cada respectivo valor significativo no bloco atual, um respectivo elemento de sintaxe indicando se o respectivo valor significativo é positivo ou negativo. De modo similar, neste exemplo, o decodificador de vídeo 30 pode determinar que a ocultação de dados de sinal é

desabilitada para o bloco atual se o bloco atual é gerado usando a codificação com perda sem a aplicação de uma transformada (ex., uma transformada de cosseno discreta, transformada direcional, ou outra transformada) aos dados residuais e o bloco atual é intra previsto usando um modo de intra-previsão no qual a DPCM residual é usada. Em tais exemplos, quando a ocultação de dados de sinal é desabilitada para o bloco atual, o decodificador de vídeo 30 obtém, a partir do bitstream, para cada respectivo valor significativo no bloco, um respectivo elemento de sintaxe indicando se o respectivo valor significativo é positivo ou negativo.

**[0164]** Em outros exemplos, a ocultação de dados de sinal pode ser desabilitada para um bloco quando a transformada é pulada, o bloco é intra-codificado, ou para todos os blocos de pulo de transformada. Assim, neste exemplo, a ocultação de dados de sinal pode ser desabilitada independente de qual modo de intra-previsão é usado. Por exemplo, um codificador de vídeo pode determinar que a ocultação de dados de sinal é desabilitada para um bloco atual se o bloco atual é codificado sem a aplicação da transformada aos dados residuais do bloco atual, e o bloco atual é intra codificado usando um modo de intra-previsão DC ou um modo de intra-previsão plano.

**[0165]** Deve-se observar que esta revelação discutiu as técnicas para DPCM residual e modificações para DC e planas. Qualquer uma ou uma combinação destas técnicas podem ser usadas.

**[0166]** FIG. 6 é um diagrama em bloco ilustrando um codificador de vídeo exemplar 20 que pode implementar as técnicas desta revelação. A FIG. 6 é fornecida para fins de explicação e não deve ser considerada limitante das técnicas como amplamente



exemplificado e descrito nesta revelação. Para fins de explicação, esta revelação descreve o codificador de vídeo 20 no contexto da codificação HEVC. No entanto, as técnicas desta revelação podem ser aplicáveis a outros padrões ou métodos de codificação.

**[0167]** No exemplo da FIG. 6, o codificador de vídeo 20 inclui uma unidade de processamento de previsão 100, uma unidade de geração residual 102, uma unidade de processamento de transformada 104, uma unidade de quantificação 106, uma unidade de quantificação inversa 108, uma unidade de processamento de transformada inversa 110, uma unidade de reconstrução 112, uma unidade de filtro 114, um buffer de imagem codificada 116, e uma unidade de codificação de entropia 118. A unidade de processamento de previsão 100 inclui uma unidade de processamento de inter-previsão 120 e uma unidade de processamento de intra-previsão 126. A unidade de processamento de inter-previsão 120 inclui uma unidade de estimativa de movimento 122 e uma unidade de compensação de movimento 124. Em outros exemplos, o codificador de vídeo 20 pode incluir mais, menos, ou componentes funcionais diferentes.

**[0168]** Codificador de vídeo 20 pode receber dados de vídeo. O codificador de vídeo 20 pode codificar cada CTU em uma fatia de uma imagem dos dados de vídeo. Cada uma das CTUs pode ser associada com os blocos de árvore de codificação luma dimensionados igualmente (CTBs) e CTBs correspondentes da imagem. Como parte da codificação de uma CTU, a unidade de processamento de previsão 100 pode realizar a partição de quad-tree para dividir os CTBs da CTU em blocos progressivamente menores. O menor bloco pode ser bloco de codificação de CUs. Por exemplo, a unidade de processamento de previsão 100 pode dividir um CTB associado com a CTU em quatro sub-blocos

igualmente dimensionados, divisão de um ou mais de um dos sub-blocos em quatro sub-blocos dimensionados igualmente, e assim por diante.

**[0169]** O codificador de vídeo 20 pode codificar CUs de uma CTU para gerar representações codificadas das CUs (i.e., CUs codificadas). Como parte da codificação de uma CU, a unidade de processamento de previsão 100 pode dividir os blocos de codificação associados com a CU dentre uma ou mais PUs da CU. Assim, cada PU pode ser associada com um bloco de previsão luma e blocos de previsão croma correspondentes. O codificador de vídeo 20 e decodificador de vídeo 30 podem suportar PUs tendo vários tamanhos. Como indicado acima, o tamanho de uma CU pode se referir ao tamanho do bloco de codificação luma da CU e o tamanho de uma PU pode se referir ao tamanho de um bloco de previsão luma da PU. Assumindo que o tamanho de uma CU particular é  $2N \times 2N$ , o codificador de vídeo 20 e decodificador de vídeo 30 podem suportar tamanhos de PU de  $2N \times 2N$  ou  $N \times N$  para intra-previsão, e tamanhos de PU simétricos de  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$ ,  $N \times N$ , ou similar para inter previsão. O codificador de vídeo 20 e decodificador de vídeo 30 também podem suportar a divisão assimétrica para tamanhos de PU de  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$ , e  $nR \times 2N$  para inter-previsão.

**[0170]** A unidade de processamento de inter-previsão 120 pode gerar dados preditivos para uma PU pela realização da inter-previsão em cada PU de uma CU. Os dados preditivos para a PU podem incluir blocos de amostra preditivos (i.e., bloco preditivos) da PU e informação de movimento para a PU. A unidade de inter-previsão 121 pode realizar diferentes operações para uma PU de uma CU dependendo de se a PU está em uma fatia I, uma fatia P, ou uma fatia B. Em uma fatia I, todas as Pus são intra

previstas. Consequentemente, se a PU está em uma fatia I, a unidade de inter-previsão 121 não realiza a inter-previsão na PU. Assim, para blocos codificados no modo I, o bloco preditivo é formado usando a previsão espacial a partir dos blocos vizinhos codificados anteriormente dentro do mesmo quadro.

**[0171]** Se uma PU está em uma fatia P, a unidade de estimativa de movimento 122 pode buscar as imagens de referência em uma lista de imagens de referência (ex., "RefPicList0") para uma região de referência para a PU. A região de referência para a PU pode ser uma região, dentro de uma imagem de referência, que contém as amostras (ex., blocos de amostra) que correspondem mais proximamente aos blocos de amostra da PU. A unidade de estimativa de movimento 122 pode gerar um índice de referência que indica uma posição em RefPicList0 da imagem de referência contendo a região de referência para a PU. Além disso, a unidade de estimativa de movimento 122 pode gerar um vetor de movimento que indica um deslocamento espacial entre um bloco de codificação da PU e um local de referência associado com a região de referência. Por exemplo, o vetor de movimento pode ser um vetor bidimensional que fornece um deslocamento a partir das coordenadas na imagem de codificada atual para as coordenadas em uma imagem de referência. A unidade de estimativa de movimento 122 pode produzir o índice de referência e o vetor de movimento como a informação de movimento da PU. A unidade de compensação de movimento 124 pode gerar os blocos de amostra preditivos da PU baseado nas amostras reais ou interpoladas no local de referência indicado pelo vetor de movimento da PU.

**[0172]** Se uma PU está em uma fatia B, a unidade de estimativa de movimento 122 pode realizar a uni-previsão ou bi-previsão para a PU. Para realizar a uni-

previsão para a PU, a unidade de estimativa de movimento 122 pode buscar as imagens de referência da RefPicList0 ou uma segunda lista de imagem de referência ("RefPicList1") para uma região de referência para a PU. A unidade de estimativa de movimento 122 pode produzir, como a informação de movimento da PU, um índice de referência que indica uma posição em RefPicList0 ou RefPicList1 da imagem de referência que contém uma região de referência, um vetor de movimento que indica um deslocamento espacial entre uma bloco de amostra da PU e um local de referência associada com a região de referência, e um ou mais indicadores de direção de previsão que indicam se a imagem de referência está na RefPicList0 ou RefPicList1. A unidade de compensação de movimento 124 pode gerar os blocos de amostra preditivos da PU com base pelo menos em parte nas amostras reais ou interpoladas em uma região de referência indicada pelo vetor de movimento da PU.

**[0173]** Para realizar a inter-previsão bi-direcional para uma PU, a unidade de estimativa de movimento 122 pode buscar as imagens de referência na RefPicList0 para uma região de referência para a PU e também pode buscar as imagens de referência na RefPicList1 para outra região de referência para a PU. A unidade de estimativa de movimento 122 pode gerar índices de imagem de referência (i.e., índices de referência) que indicam as posições nas RefPicList0 e RefPicList1 das imagens de referência que contém as regiões de referência. Além disso, a unidade de estimativa de movimento 122 pode gerar vetores de movimento que indicam deslocamentos espaciais entre o local de referência associado com as regiões de referência e um bloco de amostra da PU. A informação de movimento da PU pode incluir os índices de referência e os vetores de movimento da PU. A unidade de compensação de

movimento 124 pode gerar os blocos de amostra preditivos da PU com base pelo menos em parte nas amostras reais ou interpoladas em uma região de referência indicada pelo vetor de movimento da PU.

**[0174]** De acordo com uma ou mais técnicas desta revelação, uma ou mais unidades dentro da unidade de processamento de previsão 100 (como unidade de processamento de intra-previsão 126) pode realizar uma ou mais das técnicas descritas aqui como parte de um processo de codificação de vídeo.

**[0175]** A unidade de processamento de intra-previsão 126 pode gerar dados preditivos para uma PU pela realização da intra-previsão na PU. Os dados preditivos para a PU podem incluir bloco preditivos para a PU e vários elementos de sintaxe. A unidade de processamento de intra-previsão 126 pode realizar a intra-previsão nas PUs nas fatias I, fatias P, e fatias B.

**[0176]** Para realizar a intra-previsão em uma PU, a unidade de processamento de intra-previsão 126 pode usar vários modos de intra-previsão para gerar vários conjuntos de dados preditivos para a PU. Para usar alguns modos de intra-previsão para gerar um conjunto de dados preditivos para a PU, a unidade de processamento de intra-previsão 126 pode estender as amostras a partir de blocos de amostra das PUs vizinhas através dos blocos de amostra da PU em uma direção associada com o modo de intra-previsão. As PUs vizinhas podem estar acima, acima e à direita, acima e à esquerda, ou à esquerda da PU, assumindo uma ordem de codificação da esquerda para a direita, de cima para baixo para PUs, CUs, e CTUs. A unidade de processamento de intra-previsão 126 pode usar vários números de modos de intra-previsão, ex., 33 modos de intra-previsão direcional. Em alguns exemplos, o número de modos

de intra-previsão pode depender do tamanho da região associada com uma PU. Além disso, como descrito acima, a unidade de processamento de intra-previsão 126 pode implementar modos de codificação sem perdas e as modificações descritas aqui para melhorar tais modos de codificação. De acordo com algumas técnicas desta revelação, a unidade de processamento de intra-previsão 126 pode usar um ou mais valores de amostra originais dentro de um bloco para realizar a previsão intra DC de outros valores de amostra dentro do bloco.

**[0177]** A unidade de processamento de previsão 100 pode selecionar os dados preditivos para as PUs de uma CU dentre os dados preditivos gerados pela unidade de processamento de inter-previsão 120 para as PUs ou os dados preditivos gerados pela unidade de processamento de intra-previsão 126 para as PUs. Em alguns exemplos, a unidade de processamento de previsão 100 seleciona os dados preditivos para as PUs da CU baseado nas métricas de taxa/distorção dos conjuntos de dados preditivos. Os blocos de amostra preditivos dos dados preditivos selecionados podem ser referidos aqui como os blocos de amostra preditivos selecionados.

**[0178]** A unidade de geração residual 102 pode gerar, baseado nos blocos de codificação (ex., os blocos de codificação luma, Cb e Cr) de uma CU e os blocos preditivos selecionados (ex., blocos luma, Cb e Cr) das PUs da CU, blocos residuais (ex., blocos residuais luma, Cb e Cr) para a CU. Por exemplo, a unidade de geração residual 102 pode gerar os blocos residuais da CU de modo que cada amostra nos blocos residuais tenha um valor igual a uma diferença entre uma amostra em um bloco de codificação da CU e uma amostra correspondente em um bloco preditivo selecionado correspondente de uma PU da CU.

**[0179]** A unidade de processamento de transformada 104 pode realizar a partição de quad-tree para dividir os blocos residuais associados com uma CU nos blocos de transformada associados com as TUs da CU. Assim, a TU pode ser associada com um bloco de transformada luma e dois blocos de transformada croma. Os tamanhos e posições dos blocos de transformada luma e croma das TUs de uma CU podem ou não ser baseados nos tamanhos e posições dos blocos de previsão das PUs da CU. Uma estrutura de quad-tree conhecida como um "quad-tree residual" (RQT) pode incluir nós associados com cada uma das regiões. As TUs de uma CU podem corresponder aos nós de folha da RQT.

**[0180]** A unidade de processamento de transformada 104 pode gerar blocos de coeficientes de transformada para cada TU de uma CU pela aplicação de uma ou mais transformadas a um bloco de transformada das TU. A unidade de processamento de transformada 104 pode aplicar várias transformadas a um bloco de transformada associado com a TU. Por exemplo, a unidade de processamento de transformada 104 pode aplicar uma transformada de cosseno discreta (DCT), uma transformada direcional, ou uma transformada conceitualmente similar a um bloco de transformada. Em alguns exemplos, a unidade de processamento de transformada 104 não aplica transformadas a um bloco de transformada. Em tais exemplos, um bloco transformado pode ser tratado como um bloco de coeficiente de transformada.

**[0181]** A unidade de quantificação 106 pode quantificar os coeficientes de transformada em um bloco de coeficiente. O processo de quantificação pode reduzir a profundidade de bit associada com alguns ou todos de coeficientes de transformada. Por exemplo, um coeficiente de transformada de n-bit pode ser arredondado para baixo

para um coeficiente de transformada  $m$ -bit durante a quantificação, onde  $n$  é maior que  $m$ . A unidade de quantificação 106 pode quantificar um bloco de coeficiente associado com a TU de uma CU baseado em um valor de parâmetro de quantificação (QP) associado com a CU. O codificador de vídeo 20 pode ajustar o grau de quantificação aplicado ao bloco de coeficientes associado com uma CU pelo ajuste do valor QP associado com a CU. A quantificação pode introduzir perda de informação, assim os coeficientes de transformada quantificados podem ter precisão inferior do que os originais.

**[0182]** De acordo com alguns exemplos desta revelação, o seguinte pode ser realizado para  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ , onde  $M$  é uma altura de um bloco e  $N$  é a largura de um bloco. Em tais exemplos, o bloco é um bloco residual que inclui valores residuais que indicam diferenças entre valores de amostra em um bloco preditivo gerado usando valores de intra-previsão e amostras originais. Além disso, em tais exemplos, o bloco é um bloco de pulso de transformada. A unidade de geração residual 102 pode determinar um valor residual modificado  $\tilde{r}_{i,j}$  para um valor residual  $r_{i,j}$ . Se o bloco é codificado usando um modo de intra-previsão vertical,  $\tilde{r}_{i,j}$  é definido como:



$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases} '$$

Na equação acima,  $Q(r_{(i-1),j})$  denota um valor residual reconstruído para um valor residual  $r_{i-1,j}$  uma coluna à esquerda do valor residual  $r_{i,j}$ . Se o bloco é codificado usando um modo de intra-previsão horizontal,  $\tilde{r}_{i,j}$  é definido como:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j=0 \\ r_{i,j} - Q(r_{i,(j-1)}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} '$$

Na equação acima,  $Q(r_{i,(j-1)})$  denota um valor residual reconstruído para um valor residual  $r_{i,j-1}$  uma fileira acima do valor residual  $r_{i,j}$ . A unidade de quantificação 106 pode quantificar o valor residual modificado  $\tilde{r}_{i,j}$  para produzir um valor residual modificado quantificado  $Q(\tilde{r}_{i,j})$ .

**[0183]** A unidade de quantificação inversa 108 e unidade de processamento de transformada inversa 110 pode aplicar a quantificação inversa e transformadas inversas a um bloco de coeficiente, respectivamente, para reconstruir um bloco residual a partir do bloco de coeficiente. A unidade de reconstrução 112 pode adicionar o bloco residual reconstruído às amostras correspondentes de um ou mais bloco preditivos gerados pela unidade de processamento de previsão 100 para produzir um bloco de transformada reconstruído associado com a TU. Pela reconstrução dos blocos de transformada para cada TU de uma CU desta maneira, o codificador de vídeo 20 pode reconstruir os blocos de codificação da CU.

**[0184]** A unidade de filtro 114 pode realizar uma ou mais operações de desbloqueio para reduzir artefatos de bloqueio nos blocos de codificação associados com uma CU. O buffer da imagem codificada 116 pode armazenar os blocos de codificação reconstruídos após a unidade de filtro 114 realizar as uma ou mais operações de desbloqueio nos blocos de codificação reconstruídos. A unidade de processamento de inter-previsão 120 pode usar a imagem de referência que contém os blocos de codificação reconstruídos para realizar a inter-previsão nas PUs de outras imagens. Além disso, a unidade de processamento de intra-previsão 126 pode usar blocos de codificação reconstruídos no buffer da imagem codificada 116 para realizar a intra-previsão em outras PUs na mesma imagem como a CU.

**[0185]** A unidade de codificação de entropia 118 pode receber dados de outros componentes funcionais do codificador de vídeo 20. Por exemplo, a unidade de codificação de entropia 118 pode receber blocos de coeficientes da unidade de quantificação 106 e pode receber elementos de sintaxe da unidade de processamento de previsão 100. A unidade de codificação de entropia 118 pode realizar uma ou mais operações de codificação de entropia nos dados para gerar os dados codificados por entropia. Por exemplo, a unidade de codificação de entropia 118 pode realizar uma operação de codificação de comprimento variável de contexto adaptativo (CAVLC), uma operação de CABAC, uma operação de codificação de comprimento de variável para variável (V2V), uma operação de Codificação Aritmética Binária Adaptativo ao Contexto (SBAC) baseada na sintaxe, uma operação de codificação de Entropia de Divisão de Intervalo de Probabilidade (PIPE), uma operação de codificação Exponencial de Golomb, ou outro tipo de

operação de codificação de entropia nos dados. O codificador de vídeo 20 pode produzir um bitstream que inclui dados codificados por entropia gerados pela unidade de codificação de entropia 118. Por exemplo, o bitstream pode incluir dados que representam uma RQT para uma CU.

**[0186]** De acordo com alguns exemplos desta revelação, a unidade de codificação de entropia 118 pode determinar que a ocultação de dados de sinal é desabilitada para um bloco atual se o bloco atual é gerado sem a aplicação de uma transformada aos dados residuais e o bloco atual é intra previsto usando um modo de intra-previsão no qual a DPCM residual é usada. Quando a ocultação de dados de sinal é desabilitada para o bloco atual, a unidade de codificação de entropia 118 pode incluir, no bitstream, um elemento de sintaxe que indica se um valor no bloco atual é positivo ou negativo.

**[0187]** O elemento 130 Na FIG. 6 pode representar uma comutação (ou uma comutação conceitual) para selecionar entre a codificação sem perdas e codificação com perda. O sinal de controle 132 pode representar um sinal da unidade de processamento de previsão 100 que determina a codificação sem perda ou com perda e o elemento 134 pode representar um ciclo de decodificação que ignora os processos de transformada inversa e quantificação inversa. Em alguns exemplos, a codificação sem perdas elimina as transformadas e quantificação. Em outros exemplos, a codificação sem perdas realiza as transformadas e elimina somente o processo de quantificação. Ainda em outros exemplos, a codificação sem perdas pode ser implementada com o uso de transformadas e quantificação, mas o parâmetro de quantificação pode ser selecionado de modo a evitar

qualquer perda de dados de quantificação. Estes e outros exemplos estão dentro do escopo desta revelação.

**[0188]** Os elementos 136 e 138 representam comutadores (ou comutadores conceituais) que podem ser usados para implementar um modo de pulo de transformada. Nos modos de pulo de transformada, os dados residuais não são transformados pela unidade de processamento de transformada 104 mas é quantificada pela unidade de quantificação 106. Assim, as linhas pontilhadas do elemento 136 representam duas vias de dados possíveis. Em um dado, os dados residuais são quantificados pela unidade de quantificação 106 e na outra via de dados os dados residuais não são quantificados pela unidade de quantificação 106. De modo similar, no ciclo de decodificação do codificador de vídeo 20, os dados residuais são quantificados inversamente pela unidade de quantificação inversa 108 mas não são transformados pela unidade de processamento de transformada inversa 110. Assim, as linhas pontilhadas do elemento 138 representam uma via de dados alternativa onde os dados residuais são quantificados inversamente pela unidade de quantificação inversa 108 mas não são transformados pela unidade de processamento de transformada inversa 110.

**[0189]** FIG. 7 é um diagrama em bloco ilustrando um decodificador de vídeo exemplar 30 que é configurado para implementar as técnicas desta revelação. A FIG. 7 é fornecida para fins de explicação e não é limitadora das técnicas como amplamente exemplificado e descrito nesta revelação. Para fins de explicação, esta revelação descreve o decodificador de vídeo 30 no contexto da codificação HEVC. No entanto, as técnicas desta revelação podem ser aplicáveis a outros padrões de codificação ou métodos.

**[0190]** No exemplo da FIG. 7, o decodificador de vídeo 30 inclui uma unidade de decodificação de entropia 150, uma unidade de processamento de previsão 152, uma unidade de quantificação inversa 154, uma unidade de processamento de transformada inversa 156, uma unidade de reconstrução 158, uma unidade de filtro 160, e um buffer de imagem codificada 162. A unidade de processamento de previsão 152 inclui uma unidade de compensação de movimento 164 e uma unidade de processamento de intra-previsão 166. Em outros exemplos, o decodificador de vídeo 30 pode incluir mais, menos, ou componentes funcionais diferentes.

**[0191]** Decodificador de vídeo 30 pode receber um bitstream. A unidade de decodificação de entropia 150 pode analisar o bitstream para decodificar os elementos de sintaxe a partir do bitstream. A unidade de decodificação de entropia 150 pode decodificar por entropia os elementos de sintaxe com a entropia codificada no bitstream. A unidade de processamento de previsão 152, unidade de quantificação inversa 154, unidade de processamento de transformada inversa 156, unidade de reconstrução 158, e unidade de filtro 160 podem gerar dados de vídeo decodificados baseado nos elementos de sintaxe extraídos a partir do bitstream.

**[0192]** O bitstream pode compreender uma série de unidades NAL. As unidades NAL do bitstream podem incluir unidades NAL de fatia codificada. Como parte da decodificação dos bitstream, a unidade de decodificação de entropia 150 pode extrair e decodificar a entropia dos elementos de sintaxe a partir das unidades NAL de fatia codificada. Cada uma das fatias codificadas pode incluir um cabeçalho de fatia e dados de fatia. O cabeçalho de fatia pode conter elementos de sintaxe que pertencem a uma fatia. Os elementos de sintaxe no cabeçalho de fatia podem

incluir um elemento de sintaxe que identifica um PPS associado com a imagem que contém a fatia.

**[0193]** De acordo com alguns exemplos desta revelação, a unidade de decodificação de entropia 150 determina que a ocultação de dados de sinal é desabilitada para um bloco atual se o bloco atual é gerado sem a aplicação de uma transformada aos dados residuais e o bloco atual é intra previsto usando um modo de intra-previsão no qual uma DPCM residual é usada. Em tais exemplos, quando a ocultação de dados de sinal é desabilitada para o bloco atual, a unidade de decodificação de entropia 150 obtém, a partir do bitstream, para cada respectivo valor significativo no bloco, um respectivo elemento de sintaxe indicando se o respectivo valor significativo é positivo ou negativo.

**[0194]** Além da decodificação (i.e., obtenção) dos elementos de sintaxe a partir do bitstream, o decodificador de vídeo 30 pode realizar uma operação de reconstrução em uma CU não dividida. Para realizar a operação de reconstrução em uma CU não dividida, o decodificador de vídeo 30 pode realizar uma operação de reconstrução em cada TU da CU. Pela realização da operação de reconstrução para cada TU da CU, o decodificador de vídeo 30 pode reconstruir blocos residuais da CU.

**[0100]** Como parte da realização da operação de reconstrução em uma TU de uma CU, a unidade de quantificação inversa 154 pode quantificar inversamente, i.e., desquantificar, blocos de coeficiente associados com a TU. A unidade de quantificação inversa 154 pode usar um valor de QP associado com a CU da TU para determinar um grau de quantificação e, da mesma forma, um grau de quantificação inversa para a unidade de quantificação inversa 154 se aplicar. Ou seja, a razão de compressão,

i.e., a razão do número de bits usada para representar a sequência original e uma comprimida, pode ser controlada pelo ajuste do valor do QP usado ao quantificar coeficientes de transformada. A razão de compressão também pode depender do método de codificação de entropia empregado.

**[0101]** Após a unidade de quantificação inversa 154 quantificar inversamente um bloco de coeficiente, a unidade de processamento de transformada inversa 156 pode aplicar uma ou mais transformadas inversa ao bloco de coeficiente para gerar um bloco residual associado com a TU. Por exemplo, a unidade de processamento de transformada inversa 156 pode aplicar uma DCT inversa, uma transformada de número inteiro inversa, uma transformada Karhunen-Loeve inversa (KLT), uma transformada rotacional inversa, uma transformada direcional inversa, ou outra transformada inversa ao bloco de coeficiente.

**[0102]** Se uma PU é codificada usando a intra-previsão, a unidade de processamento de intra-previsão 166 pode realizar a intra-previsão para gerar blocos preditivos para a PU. A unidade de processamento de intra-previsão 166 pode usar um modo de intra-previsão para gerar os blocos preditivos (ex., blocos luma, Cb e Cr preditivos) para a PU baseado nos blocos de previsão das PUs espacialmente vizinhas. A unidade de processamento de intra-previsão 166 pode determinar o modo de intra-previsão para a PU baseado em um ou mais elementos de sintaxe decodificados a partir do bitstream. De acordo com algumas técnicas desta revelação, a unidade de processamento de intra-previsão 166 pode usar um ou mais valores de amostra originais dentro de um bloco para realizar a previsão intra DC de outros valores de amostra dentro do bloco. Ou seja, a unidade de processamento de intra-previsão 166 pode gerar

um bloco preditivo. Como parte da geração do bloco preditivo, a unidade de processamento de intra-previsão 166 pode usar pelo menos uma de uma amostra reconstruída sem perdas à esquerda de uma amostra atual em uma fileira atual de um bloco preditivo e uma amostra reconstruída sem perdas para uma fileira do bloco preditivo acima da fileira atual para a previsão DC da amostra atual,

**[0103]** A unidade de processamento de previsão 152 pode construir uma primeira lista de imagem de referência (RefPicList0) e uma segunda lista de imagem de referência (RefPicList1) baseada nos elementos de sintaxe extraídos a partir do bitstream. Além disso, se uma PU é codificada usando a inter previsão, a unidade de decodificação de entropia 150 pode extrair a informação de movimento para a PU. A unidade de compensação de movimento 164 pode determinar, baseado na informação de movimento da PU, uma ou mais regiões de referência para a PU. A unidade de compensação de movimento 164 pode gerar, baseado nos blocos de amostra em um ou mais blocos de referência para a PU, blocos preditivos (ex., blocos luma, Cb e Cr) para a PU. De acordo com uma ou mais técnicas desta revelação, uma ou mais unidades dentro da unidade de processamento de previsão 152 (como unidade de processamento de intra-previsão 166) pode realizar as técnicas descritas aqui como parte de um processo de decodificação de vídeo.

**[0104]** A unidade de reconstrução 158 pode usar um bloco transformado (ex., blocos de transformada luma, Cb e Cr) associado com as TUs de uma CU e os blocos preditivos (ex., blocos preditivos luma, Cb e Cr) das PUs da CU, i.e., ou dados de intra-previsão ou dados de inter-previsão, como aplicável, para reconstruir os blocos de codificação (ex., blocos de codificação luma, Cb e Cr) da CU. Por exemplo, a unidade de reconstrução 158 pode adicionar amostras dos



blocos de transformada luma, Cb e Cr às amostras correspondentes dos blocos luma, Cb e Cr preditivos para reconstruir os blocos de codificação luma, Cb e Cr da CU.

**[0200]** De acordo com alguns exemplos desta revelação, a unidade de decodificação de entropia 150 pode gerar um bloco de valores residuais. Este bloco pode ser um bloco de pulo de transformada. Além disso, o bloco pode ser um bloco residual que inclui valores residuais que indicam diferenças entre valores de amostra originais e valores de amostra em um bloco preditivo gerado usando a intra-previsão. Além disso, para  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ , onde M é uma altura do bloco e N é a largura do bloco, a unidade de quantificação inversa 154 pode calcular um valor residual reconstruído  $Q(r_{i,j})$  para um valor residual  $r_{i,j}$  no bloco. Se o bloco é codificado usando um modo de intra-previsão vertical (ou, em alguns exemplos, um modo de intra-previsão quase vertical),  $Q(r_{i,j})$  é definido como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}.$$

Na equação acima,  $Q(\tilde{r}_{i,j})$  denota uma versão quantificada de um valor residual modificado  $\tilde{r}_{i,j}$ , o valor residual modificado  $\tilde{r}_{i,j}$  é uma versão modificada do valor residual  $r_{i,j}$ , e  $Q(r_{i-1,j})$  é um valor residual reconstruído para (i.e., correspondente a) um valor residual de uma coluna à esquerda do valor residual  $r_{i,j}$ . A unidade de decodificação de entropia 150 pode ter sido determinada anteriormente  $Q(r_{i-1,j})$  da mesma maneira que a unidade de decodificação de entropia 150 determina  $Q(r_{i,j})$ . Se o bloco é codificado usando um modo de intra-previsão horizontal,  $Q(r_{i,j})$  é definido como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}.$$

Na equação acima,  $Q(r_{i,j-1})$  é um valor residual reconstruído para um valor residual de uma fileira acima do valor residual  $r_{i,j}$ . A unidade de decodificação de entropia 150 pode ter sido determinada anteriormente  $Q(r_{i,j-1})$  da mesma maneira que a unidade de decodificação de entropia 150 determina  $Q(r_{i,j})$ . A unidade de reconstrução 158 pode adicionar o valor residual reconstruído  $Q(r_{i,j})$  a um valor de previsão para reconstruir um valor de amostra.

**[0201]** A unidade de filtro 160 pode realizar uma operação de desbloqueio para reduzir artefatos de bloqueio associados com os blocos de codificação (ex., os blocos de codificação luma, Cb e Cr) da CU. O decodificador de vídeo 30 pode armazenar os blocos de codificação (ex., blocos de codificação luma, Cb e Cr) da CU no buffer da imagem codificada 162. O buffer da imagem codificada 162 pode fornecer imagens de referência para a compensação de movimento subsequente, intra-previsão, e apresentação em um dispositivo de exibição, como dispositivo de exibição 32 da FIG. 1. Por exemplo, o decodificador de vídeo 30 pode realizar, baseado nos blocos (ex., blocos luma, Cb e Cr) no buffer da imagem codificada 162, operações de intra-previsão ou inter-previsão nas PUs de outras CUs. Desta maneira, o decodificador de vídeo 30 pode extrair, a partir do bitstream, o nível de coeficientes de transformadas do bloco de coeficiente significativo luma, quantificar inversamente os níveis de coeficiente de transformada, aplicar uma transformada ao nível de coeficiente de transformadas para gerar um bloco de transformada, gerar, com base pelo menos em parte nos blocos de transformada, um

bloco de codificação, e produzir o bloco de codificação para exibição.

**[0202]** O elemento 170 pode representar uma via de codificação normal para a compressão com perda, e o elemento 172 pode representar uma via de codificação de desvio que ignora os processos de transformada inversa e quantificação inversa. Estas vias diferentes são meramente exemplares e a codificação sem perdas pode ser realizada sem qualquer desvio. Em alguns exemplos, a codificação sem perdas elimina as transformadas e quantificação. Em outros exemplos, a codificação sem perdas realiza transformadas e elimina somente o processo de quantificação. Ainda em outros exemplos, a codificação sem perdas pode ser implementada com o uso de transformadas e quantificação, mas o parâmetro de quantificação pode ser selecionado de modo a evitar qualquer perda de dados de quantificação. O elemento 174 representa um exemplo de uma via que pode ser usada para um modo de pulo de transformada. Em um modo de pulo de transformada, os dados residuais podem ser quantificados inversamente pela unidade de quantificação inversa 154, mas a transformação inversa da unidade de processamento de transformada inversa 156 pode ser pulada. Estes e outros exemplos estão dentro do escopo desta revelação.

**[0203]** FIG. 8A é um fluxograma ilustrando uma operação exemplar de codificador de vídeo 20, de acordo com uma ou mais técnicas desta revelação. Em outros exemplos desta revelação, operações similares à operação da FIG. 8A podem incluir mais, menos ou ações diferentes. Além disso, em alguns exemplos, um ou mais ações da operação de FIG. 8A podem ser omitidas ou rearranjadas. Por exemplo, nas FIG. 8A e FIG. 8B, as linhas pontilhadas indicam ações não realizadas em alguns exemplos.

**[0204]** No exemplo da FIG. 8A, o codificador de vídeo 20 pode gerar um bloco preditivo (200). Como parte da geração do bloco preditivo, codificador de vídeo 20 pode usar pelo menos um de: uma amostra reconstruída sem perdas à esquerda de uma amostra atual em uma fileira atual de um bloco preditivo e uma amostra reconstruída sem perdas para uma fileira do bloco preditivo acima da fileira atual para a previsão DC da amostra atual (202). Em alguns exemplos, o decodificador de vídeo 30 processa as amostras no bloco preditivo em uma ordem de varredura de rastreamento horizontal, uma ordem de varredura de rastreo vertical, uma ordem de varredura diagonal, ou uma ordem de varredura em ziguezague. Em tais exemplos, o processamento das amostras no bloco preditivo pode compreender determinar as previsões DC para as amostras assim como reconstruir as amostras sem perdas. Além disso, no exemplo da FIG. 8A, o codificador de vídeo 20 pode gerar amostras residuais que tem valores iguais a uma diferença entre uma amostra em um bloco de codificação e uma amostra correspondente no bloco preditivo.

**[0205]** FIG. 8B é um fluxograma ilustrando uma operação exemplar de decodificador de vídeo 30, de acordo com uma ou mais técnicas desta revelação. Em outros exemplos desta revelação, operações similares à operação da FIG. 8B podem incluir mais, menos ou ações diferentes. Além disso, em alguns exemplos, um ou mais ações da operação de FIG. 8B podem ser omitidas ou rearranjadas.

**[0206]** No exemplo da FIG. 8B, o decodificador de vídeo 30 pode gerar um bloco preditivo (250). Como parte da geração do bloco preditivo, o decodificador de vídeo 30 pode usar pelo menos um de a amostra reconstruída sem perdas à esquerda de uma amostra atual em uma fileira atual de um bloco preditivo e a amostra reconstruída sem

perdas para uma fileira do bloco preditivo acima da fileira atual para a previsão DC da amostra atual (252). Em alguns exemplos, o decodificador de vídeo 30 processa as amostras no bloco preditivo em uma ordem de varredura de rastreamento horizontal, uma ordem de varredura de rastreo vertical, uma diagonal ordem de varredura, ou uma ordem de varredura em ziguezague. Em tais exemplos, o processamento das amostras no bloco preditivo pode compreender determinar as previsões DC para a amostras. Além disso, no exemplo da FIG. 8B, o decodificador de vídeo 30 pode reconstruir um bloco de codificação pela adição das amostras do bloco preditivo às amostras residuais correspondentes (254).

**[0207]** FIG. 9A é um fluxograma ilustrando uma operação exemplar de codificador de vídeo 20, de acordo com uma ou mais técnicas desta revelação. Em outros exemplos desta revelação, as operações similares à operação da FIG. 9A podem incluir mais, menos ou ações diferentes. Além disso, em alguns exemplos, um ou mais ações da operação de FIG. 9A podem ser omitidas ou rearranjadas. O exemplo da FIG. 9A é explicado com referência aos componentes mostrados na FIG. 6. No entanto, a operação da FIG. 9A pode ser realizada pelos componentes e tipos dos codificadores de vídeo diferentes daqueles mostrados na FIG. 6.

**[0208]** Como indicado no exemplo da FIG. 9A, a unidade de geração residual 102 do codificador de vídeo 20 pode gerar um bloco de valores residuais (350). Neste exemplo, o bloco de valores residuais é um bloco de pulso de transformada. O bloco pode ser um bloco residual que inclui valores residuais que indicam diferenças entre valores de amostra originais e valores de amostra em um bloco preditivo gerado usando intra-previsão. O codificador de vídeo pode realizar as ações restantes da

FIG. 9A para cada local  $(i, j)$  do bloco, onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ ,  $M$  é uma altura de um bloco, e  $N$  é a largura do bloco.

**[0209]** Além disso, a unidade de geração residual 102 do codificador de vídeo 20 pode determinar um valor residual modificado  $\tilde{r}_{i,j}$  para um valor residual  $r_{i,j}$  no bloco (352). Se o bloco é codificado usando um modo de intra-previsão vertical, ou em alguns exemplos, um modo de intra-previsão entre 22 e 30 como definido em HEVC Working Draft 10,  $\tilde{r}_{i,j}$  é definido como:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases} '$$

$Q(r_{(i-1),j})$  denota um valor residual reconstruído para (i.e., correspondente a) um valor residual  $r_{i-1,j}$ . Se o bloco é codificado usando um modo de intra-previsão horizontal, ou, em alguns exemplos, um modo de intra-previsão entre 6 e 14,  $\tilde{r}_{i,j}$  é definido como:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j=0 \\ r_{i,j} - Q(r_{i,(j-1)}), & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} '$$

$Q(r_{i,(j-1)})$  denota um valor residual reconstruído para um valor residual  $r_{i,j-1}$ .

**[0210]** Além disso, a unidade de quantificação 106 do codificador de vídeo 20 pode quantificar o valor residual modificado  $\tilde{r}_{i,j}$  para produzir um valor residual modificado quantificado  $Q(\tilde{r}_{i,j})$  (354). O codificador de vídeo 20 pode sinalizar o valor residual modificado quantificado  $Q(\tilde{r}_{i,j})$  em um bitstream (356). Por exemplo, o codificador de

vídeo 20 pode gerar um ou mais elementos de sintaxe que indicam o valor residual modificado quantificado  $Q(\tilde{r}_{i,j})$ . Neste exemplo, a unidade de codificação de entropia 118 do codificador de vídeo 20 pode codificar a entropia de um ou mais elementos de sintaxe e incluem os dados resultantes no bitstream.

**[0211]** Além disso, no exemplo da FIG. 9A, o codificador de vídeo 20 pode calcular um valor residual reconstruído  $Q(r_{i,j})$  (358). Em alguns exemplos, o codificador de vídeo 20 pode calcular o valor residual reconstruído  $Q(r_{i,j})$  como parte de um ciclo de realimentação para determinar os valores de amostra reconstruídos para uso na intra-previsão ou inter-previsão adicional. Se o bloco é codificado usando o modo de intra-previsão vertical (ou em alguns exemplos, um modo de intra-previsão quase vertical), o valor residual reconstruído  $Q(r_{i,j})$  pode ser definido como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases} \quad '$$

Se o bloco é codificado usando o modo de intra-previsão horizontal (ou em alguns exemplos, um modo de intra-previsão quase horizontal),  $Q(r_{i,j})$  pode ser definido como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}), & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad '$$

**[0212]** Porque o bloco é um bloco de pulso de transformada, a unidade de processamento de transformada inversa 110 não aplica uma transformada inversa ao valor

residual  $r_{i,j}$ . Consequentemente, a unidade de reconstrução 112 do codificador de vídeo 20 pode adicionar o valor residual reconstruído  $Q(r_{i,j})$  a um valor de previsão para determinar um valor de amostra reconstruído (360). O valor de previsão pode ser uma amostra em um bloco preditivo. A unidade de processamento de previsão 100 do codificador de vídeo 20 pode usar o valor de amostra reconstruído para a intra-previsão ou inter-previsão de outros blocos (362).

**[0213]** FIG. 9B é um fluxograma ilustrando uma operação exemplar de decodificador de vídeo 30, de acordo com uma ou mais técnicas desta revelação. Em outros exemplos desta revelação, operações similares à operação da FIG. 9B podem incluir mais, menos ou ações diferentes. Além disso, em alguns exemplos, um ou mais ações da operação de FIG. 9B podem ser omitidas ou rearranjadas. O exemplo da FIG. 9B é explicado com referência aos componentes mostrados na FIG. 7. No entanto, a operação da FIG. 9B pode ser realizada pelos componentes e tipos de decodificadores de vídeo diferentes daqueles mostrados na FIG. 7.

**[0214]** Decodificador de vídeo 30 pode realizar a operação da FIG. 9B para cada local  $(i, j)$  de um bloco de pulso de transformada, onde  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ ,  $M$  é uma altura de um bloco, e  $N$  é a largura do bloco. O bloco pode ser um bloco residual que inclui valores residuais que indicam diferenças entre valores de amostra originais e valores de amostra em um bloco preditivo gerado usando intra-previsão. Como indicado no exemplo da FIG. 9A, a unidade de decodificação de entropia 150 do decodificador de vídeo 30 pode obter, a partir de um bitstream, um ou mais elementos de sintaxe indicando um valor residual quantificado modificado  $Q(\tilde{r}_{i,j})$  (400). A



unidade de decodificação de entropia 150 pode decodificar por entropia alguns ou todos de um ou mais elementos de sintaxe.

**[0215]** Além disso, no exemplo da FIG. 9B, a unidade de quantificação inversa 154 do decodificador de vídeo 30 pode calcular um valor residual reconstruído  $Q(r_{i,j})$  para um valor residual  $r_{i,j}$  (402). Em alguns exemplos, o valor residual  $r_{i,j}$  é um valor residual deslocado em bit como descrito em outro lugar nesta revelação. Se o bloco é codificado usando um modo de intra-previsão vertical,  $Q(r_{i,j})$  é definido como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases} .$$

$Q(\tilde{r}_{i,j})$  denota uma versão quantificada de um valor residual modificado  $\tilde{r}_{i,j}$ , o valor residual modificado  $\tilde{r}_{i,j}$  é uma versão modificada do valor residual  $r_{i,j}$ , e  $Q(r_{i-1,j})$  é um valor residual reconstruído para um valor residual de uma coluna à esquerda do valor residual  $r_{i,j}$ . Se o bloco é codificado usando um modo de intra-previsão horizontal,  $Q(r_{i,j})$  é definido como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}), & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} .$$

$Q(r_{i,j-1})$  é um valor residual reconstruído para um valor residual de uma fileira acima do valor residual  $r_{i,j}$ .

**[0216]** Se o bloco é codificado usando o modo de intra-previsão vertical, o valor residual modificado  $\tilde{r}_{i,j}$  é definido como:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}.$$

Se o bloco é codificado usando o modo de intra-previsão horizontal, o valor residual modificado  $\tilde{r}_{i,j}$  é definido como:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j=0 \\ r_{i,j} - Q(r_{i,(j-1)}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}.$$

**[0217]** Porque o bloco é um bloco de pulso de transformada, a unidade de processamento de transformada inversa 156 não aplica uma transformada inversa ao valor residual reconstruído  $Q(r_{i,j})$ . A unidade de reconstrução 158 do decodificador de vídeo 30 pode adicionar o valor residual reconstruído  $Q(r_{i,j})$  a um valor de previsão para reconstruir um valor de amostra (406). O valor de previsão pode ser uma amostra em um bloco preditivo.

**[0218]** FIG. 10A é um fluxograma ilustrando uma operação do codificador de vídeo exemplar para ocultação de dados de sinal, de acordo com uma ou mais técnicas desta revelação. Em outros exemplos desta revelação, operações similares à operação da FIG. 10A podem incluir mais, menos ou ações diferentes. Além disso, em alguns exemplos, uma ou mais ações da operação da FIG. 10A podem ser omitidas ou rearranjadas.

**[0219]** No exemplo da FIG. 10A, o codificador de vídeo 20 gera um bitstream que inclui uma sequência de bits que forma uma representação codificada dos dados de vídeo (600). Como parte da geração do bitstream, codificador de vídeo 20 pode determinar que a ocultação de

dados de sinal é desabilitada para um bloco atual se o bloco atual é gerado sem a aplicação de uma transformada aos dados residuais e o bloco atual é intra previsto usando um modo de intra-previsão no qual uma técnica de DPCM residual é usada (602). No contexto da FIG. 10A, o bloco atual pode ser um sub-bloco 4x4 de um bloco de amostras residuais aos quais a técnica de DPCM residual foi aplicada. Subsequentemente, no exemplo da FIG. 10A, o codificador de vídeo 20 pode produzir o bitstream (604).

**[0220]** Em alguns exemplos, quando a ocultação de dados de sinal é desabilitada para o bloco atual, codificador de vídeo 20 pode sinalizar, no bitstream, para cada respectivo valor residual significativo no bloco atual, um elemento de sintaxe indicando se o respectivo valor residual significativo é positivo ou negativo. Em tais exemplos, quando a ocultação de dados de sinal não está desabilitada para o bloco atual, o codificador de vídeo 20 pode não sinalizar, no bitstream, um elemento de sintaxe que indica se um valor de pelo menos um valor residual significativo ou coeficiente de transformada no bloco atual é positivo ou negativo.

**[0221]** FIG. 10B é um fluxograma ilustrando uma operação do decodificador de vídeo exemplar para ocultação de dados de sinal, de acordo com uma ou mais técnicas desta revelação. Em outros exemplos desta revelação, operações similares à operação da FIG. 10B podem incluir mais, menos ou ações diferentes. Além disso, em alguns exemplos, um ou mais ações da operação de FIG. 10B podem ser omitidas ou rearranjadas.

**[0222]** No exemplo da FIG. 10B, o decodificador de vídeo 30 obtém elementos de sintaxe a partir de um bitstream que inclui uma sequência de bits que forma uma representação codificada dos dados de vídeo (650). Como

parte da obtenção dos elementos de sintaxe do bitstream, o decodificador de vídeo 30 pode determinar que a ocultação de dados de sinal é desabilitada para um bloco atual se o bloco atual é gerado sem a aplicação de uma transformada aos dados residuais e o bloco atual é intra previsto usando um modo de intra-previsão no qual uma técnica de DPCM residual é usada (652). No contexto da FIG. 10B, o bloco atual pode ser um sub-bloco 4x4 de um bloco de amostras residuais ao qual a técnica de DPCM residual foi aplicada. Subsequentemente, no exemplo da FIG. 10B, o decodificador de vídeo 30 pode reconstruir uma imagem dos dados de vídeo com base pelo menos em parte nos elementos de sintaxe obtidos a partir do bitstream (654).

**[0223]** Em alguns exemplos, quando a ocultação de dados de sinal é desabilitada para o bloco atual, o decodificador de vídeo 30 pode obter, a partir do bitstream, para cada respectivo valor residual significativo no bloco atual, um elemento de sintaxe indicando se o respectivo valor residual significativo é positivo ou negativo. Em tais exemplos, quando a ocultação de dados de sinal não está desabilitada para o bloco atual, o decodificador de vídeo 30 pode não obter a partir do bitstream um elemento de sintaxe indicando se um valor de um valor residual significativo no bloco atual é positivo ou negativo.

**[0224]** Os parágrafos a seguir fornecem uma primeira série de exemplos de acordo com uma ou mais técnicas desta revelação.

**[0225]** Exemplo 1. Um método de codificação de dados de vídeo, o método compreendendo: usar um ou mais valores de amostra originais dentro de um bloco de dados de vídeo para realizar a previsão intra DC de outros valores de amostra dentro do bloco.

**[0226]** Exemplo 2. O método do exemplo 1, sendo que usar um ou mais valores de amostra originais dentro do bloco para realizar a previsão de outros valores de amostra dentro do bloco compreende: usar valores de amostra originais que ocorrem mais cedo em uma ordem de varredura para prever valores de amostra que ocorrem depois na ordem de varredura.

**[0227]** Exemplo 3. O método do exemplo 1, sendo que usar um ou mais valores de amostra originais dentro do bloco para realizar a previsão de outros valores de amostra dentro do bloco compreende: usar valores de amostra originais de uma fileira anteriormente varrida para prever valores de amostra de uma fileira subsequentemente varrida.

**[0228]** Exemplo 4. O método do exemplo 1, sendo que usar um ou mais valores de amostra originais dentro do bloco para realizar a previsão de outros valores de amostra dentro do bloco compreende: usar valores de amostra originais que correspondem aos vizinhos causais de uma amostra para prever valores de amostra para a amostra.

**[0229]** Exemplo 5. O método de qualquer um dos exemplos 1 - 4, sendo que um valor de previsão DC  $DC_{i,j}$ , para uma amostra atual  $P_{i,j}, 0 \leq i \leq (M-1), 0 \leq j \leq (N-1)$  é calculado como  $DC_{i,j} = (P_{i,j-1} + P_{i-1,j} + 1) \gg 1$ .

**[0230]** Exemplo 6. O método de qualquer um dos exemplos 1 - 4, sendo que um valor de previsão DC  $DC_{i,j}$ , para uma amostra atual  $P_{i,j}, 0 \leq i \leq (M-1), 0 \leq j \leq (N-1)$  é calculado como  $DC_{i,j} = (P_{i,j-1} + P_{i-1,j}) \gg 1$ .

**[0231]** Exemplo 7. O método de qualquer um dos exemplos 1 - 6, compreendendo ainda: realizar a previsão DC de um modo canalizado.

**[0232]** Exemplo 8. O método do exemplo 7, sendo que um atraso de ciclo para a previsão DC existe entre as fileiras do bloco.

**[0233]** Exemplo 9. O método de qualquer um dos exemplos 1 - 4, sendo que um valor de previsão DC  $DC_{i,j}, 0 \leq i \leq (M-1), 0 \leq j \leq (N-1)$  é calculado como  $DC_{i,j} = (P_{i,j-1} + P_{i-1,j} - P_{i-1,j-1})$ .

**[0234]** Exemplo 10. O método do exemplo 9, sendo que  $DC_{i,j+1}$  pode ser calculado sem esperar por  $P_{i,j}$  como  $DC_{i,j+1} = ((P_{i,j-1} + P_{i-1,j} - P_{i-1,j-1}) + r_{i,j} + P_{i-1,j+1} - P_{i-1,j})$ , onde  $r_{i,j}$  é o erro de previsão residual para a amostra  $P_{i,j}$ .

**[0235]** Exemplo 11. O método de qualquer um dos exemplos 1 - 4, sendo que aqueles disponíveis de uma amostra à esquerda, uma amostra superior esquerda, e uma amostra superior direita são usados para a previsão DC.

**[0236]** Exemplo 12. O método do exemplo 11, sendo que um valor de previsão DC  $DC_{i,j}, 0 \leq i \leq (M-1), 0 \leq j \leq (N-1)$  é calculado como um de:  $DC_{i,j} = (P_{i,j-1} + P_{i-1,j} + P_{i-1,j-1} + P_{i-1,j+1} + 2) \gg 2$ , ou  $DC_{i,j} = (P_{i,j-1} + P_{i-1,j} + P_{i-1,j-1} + P_{i-1,j+1}) \gg 2$ .

**[0237]** Exemplo 13. O método do exemplo 12, sendo que para as amostras em uma última coluna ( $j = (N-1), i > 0$ ), uma amostra superior direita não está disponível e as amostra superiores e superiores à direita ( $P_{i-1,j}$  e  $P_{i-1,j+1}$ ) tem o mesmo valor.

**[0238]** Exemplo 14. O método do exemplo 13, sendo que a amostra superior direita não é usada para a previsão DC.

**[0239]** Exemplo 15. Um método de codificação de dados de vídeo, o método compreendendo: realizar a previsão DC em um tamanho de bloco que é menor do que um tamanho de unidade de transformada (TU).

**[0240]** Exemplo 16. O método do exemplo 15, sendo que independentemente do tamanho da TU, a previsão DC é realizada em um tamanho de bloco de  $2 \times 2$ , sendo que pelo menos alguns tamanhos de TU são maiores do que o tamanho de bloco de  $2 \times 2$ .

**[0241]** Exemplo 17. O método do exemplo 15 ou 16, sendo que para as amostras  $P_{2i,2j}, P_{2i,2j+1}, P_{2i+1,2j}, e P_{2i+1,2j+1}$ , os valores de previsão DC são calculados como um de:  $(P_{2i-1,2j} + P_{2i-1,2j+1} + P_{2i,2j-1} + P_{2i+1,2j-1} + 2) \gg 2$  ou  $(P_{2i-1,2j} + P_{2i-1,2j+1} + P_{2i,2j-1} + P_{2i+1,2j-1}) \gg 2$ , sendo que,  $0 \leq i \leq ((M/2) - 1), 0 \leq j \leq ((N/2) - 1)$ .

**[0242]** Exemplo 18. O método do exemplo 17, sendo que M e N são ambos pares.

**[0243]** Exemplo 19. O método do exemplo 18, sendo que quatro amostras podem ser processadas em paralelo.

**[0244]** Exemplo 20. Um método de codificação de dados de vídeo, o método compreendendo: realizar a previsão DC em um tamanho de bloco de  $2 \times 2$  independente de um tamanho de uma unidade de transformada (TU).

**[0245]** Exemplo 21. Um método de codificação de dados de vídeo, o método compreendendo: explorar uma correlação entre residuais após realizar uma previsão DC normal.

**[0246]** Exemplo 22. O método do exemplo 21, sendo que  $r_{i,j}, 0 \leq i \leq (M - 1), 0 \leq j \leq (N - 1)$  representa residuais de previsão após realizar a previsão DC como especificado de acordo com um padrão HEVC, o método compreendendo ainda: gerar valores intermediários  $s_{i,j}, 0 \leq i \leq (M - 1), 0 \leq j \leq (N - 1)$  de acordo com:  $s_{i,j} = r_{i,2j}, s_{i,(j+(N/2))} = r_{i,2j} - r_{i,2j+1}, 0 \leq i \leq (M - 1), 0 \leq j \leq ((N/2) - 1)$ .

**[0247]** Exemplo 23. O método do exemplo 22, o método compreendendo ainda: gerar residuais modificados,  $t_{i,j}, 0 \leq i \leq (M-1), 0 \leq j \leq (N-1)$  de acordo com:  $t_{i,j} = s_{2i,j}, t_{(i+(M/2)),j} = s_{2i,j} - s_{2i+1,j}, 0 \leq i \leq ((M/2)-1), 0 \leq j \leq (N-1).$

**[0248]** Exemplo 24. O método do exemplo 23, sendo que os residuais modificados,  $t_{i,j}$ , são codificados por entropia.

**[0249]** Exemplo 25. O método do exemplo 21, sendo que o método é realizado durante a decodificação e sendo que:

$$s_{2i,j} = t_{i,j}, s_{2i+1,j} = t_{i,j} - t_{(i+(M/2)),j}, 0 \leq i \leq ((M/2)-1), 0 \leq j \leq (N-1) \text{ and } r_{i,2j} = s_{i,j}, r_{i,2j+1} = s_{i,j} - s_{i,(j+(N/2))}, 0 \leq i \leq (M-1), 0 \leq j \leq ((N/2)-1).$$

**[0250]** Exemplo 26. O método do exemplo 25, sendo que os M e N são ambos pares.

**[0251]** Exemplo 27. O método do exemplo 21, sendo que:

$$s_{i,j} = r_{i,2j+1}, s_{i,(j+(N/2))} = P_{i,2j} - ((P_{i,2j-1} + P_{i,2j+1} + 1) \gg 1), \\ 0 \leq i < M, 0 \leq j < (N/2), \\ t_{i,j} = s_{2i+1,j}, t_{(i+(M/2)),j} = s_{2i,j} - ((s_{2i+1,j} + s_{2i+1,j} + 1) \gg 1), \\ 0 \leq i < M/2, 0 \leq j < N.$$

**[0252]** Exemplo 28. O método de qualquer combinação dos exemplos 1 - 27.

**[0253]** Exemplo 29. O método de qualquer um dos exemplos 1 - 24 e 27 ou combinações dos mesmos, sendo que o método é realizado por um codificador e sendo que a codificação refere-se à codificação (encoding).

**[0254]** Exemplo 30. O método de qualquer um dos exemplos 1 - 21 e 25 - 27 ou combinações dos mesmos, sendo que o método é realizado por um decodificador e sendo que a codificação refere-se à decodificação.



**[0255]** Exemplo 31. Um sistema configurado para realizar o método de qualquer um dos exemplos 1 - 27 ou combinações dos mesmos.

**[0256]** Exemplo 32. Uma mídia de armazenamento legível por computador não transitória que armazena as instruções que quando executadas fazem com que um ou mais processadores realizem o método de qualquer um dos exemplos 1 - 27 ou combinações dos mesmos.

**[0257]** Exemplo 33. Um dispositivo de codificação de vídeo configurado para realizar o método de qualquer um dos exemplos 1 - 24 e 27 ou combinações dos mesmos.

**[0258]** Exemplo 34. Um dispositivo de decodificação de vídeo configurado para realizar o método de qualquer um dos exemplos 1 - 21 e 25 - 27 ou combinações dos mesmos.

**[0259]** Exemplo 35. Um dispositivo de codificação de vídeo compreendendo meios para realizar as etapas do método de qualquer um dos exemplos 1 - 24 e 27 ou combinações dos mesmos.

**[0260]** Exemplo 36. Um dispositivo de decodificação de vídeo compreendendo meios para realizar as etapas do método de qualquer um dos exemplos 1 - 21 e 25 - 27 ou combinações dos mesmos.

**[0261]** Os parágrafos a seguir fornecem uma segunda série de exemplos de acordo com uma ou mais técnicas desta revelação.

**[0262]** Exemplo 1. Um método de codificação de dados de vídeo, o método compreendendo: usar um ou mais valores de amostra originais dentro de um bloco de dados de vídeo para realizar a previsão de outros valores de amostra dentro do bloco.

**[0263]** Exemplo 2. O método do exemplo 1, sendo que usar um ou mais valores de amostra originais dentro do bloco para realizar a previsão de outros valores de amostra dentro do bloco compreende: usar valores de amostra originais correspondente a uma última fileira e uma última coluna do bloco para realizar a previsão de outros valores de amostra.

**[0264]** Exemplo 3. O método do exemplo 1, sendo que usar um ou mais valores de amostra originais dentro do bloco para realizar a previsão de outros valores de amostra dentro do bloco compreende: usar valores de amostra originais correspondente a uma primeira fileira e uma primeira coluna do bloco para realizar a previsão de outros valores de amostra.

**[0265]** Exemplo 4. O método de qualquer um dos exemplos 1 - 3, sendo que a FIG. 4 ilustra os locais das amostras dos valores de amostra usados para realizar a previsão de outros valores de amostra.

**[0266]** Exemplo 5. O método de qualquer um dos exemplos 1 - 4, sendo que o método é realizado para um modo de codificação sem perdas.

**[0267]** Exemplo 6. O método de qualquer um dos exemplos 1 - 5, sendo que o método é realizado para um modo de codificação plano.

**[0268]** Exemplo 7. O método de qualquer um dos exemplos 1 - 5, sendo que o método é realizado para um modo de intra-codificação angular.

**[0269]** Exemplo 8. O método de qualquer um dos exemplos 1 - 7, o método compreendendo ainda: realizar uma operação de rotação no conjunto de valores residuais gerado pela previsão; e realizar a codificação por entropia com relação ao conjunto girado dos valores residuais.

**[0270]** Exemplo 9. O método do exemplo 8, sendo que o conjunto de valores residuais são valores transformados.

**[0271]** Exemplo 10. O método de qualquer um dos exemplos 1 - 9, compreendendo ainda realizar um processo de previsão para codificar os valores de amostra originais.

**[0272]** Exemplo 11. O método de qualquer combinação dos exemplos 1 - 10.

**[0273]** Exemplo 12. O método de qualquer um dos exemplos 1 - 10 ou combinações dos mesmos, sendo que o método é realizado por um codificador e sendo que codificação refere-se à codificação (encoding).

**[0274]** Exemplo 13. O método de qualquer um dos exemplos 1 - 10 ou combinações dos mesmos, sendo que o método é realizado por um decodificador e sendo que codificação refere-se à decodificação.

**[0275]** Exemplo 14. Um sistema configurado para realizar o método de qualquer um dos exemplos 1 - 10 ou combinações dos mesmos.

**[0276]** Exemplo 15. Uma mídia de armazenamento legível por computador não transitória que armazena as instruções que quando executadas fazem com que um ou mais processadores realizem o método de qualquer um dos exemplos 1 - 10 ou combinações dos mesmos.

**[0277]** Exemplo 16. Um dispositivo de codificação de vídeo configurado para realizar o método de qualquer um dos exemplos 1 - 10 ou combinações dos mesmos.

**[0278]** Exemplo 17. Um dispositivo de decodificação de vídeo configurado para realizar o método de qualquer um dos exemplos 1 - 10 ou combinações dos mesmos.

**[0279]** Exemplo 18. Um dispositivo de codificação de vídeo compreendendo meios para realizar as etapas do método de qualquer um dos exemplos 1 - 10 ou combinações dos mesmos.

**[0280]** Exemplo 19. Um dispositivo de decodificação de vídeo compreendendo meios para realizar as etapas do método de qualquer um dos exemplos 1 - 10 ou combinações dos mesmos.

**[0281]** Exemplo 20. Qualquer dispositivo ou método descrito nesta revelação.

**[0282]** Os parágrafos a seguir fornecem uma terceira série dos exemplos de acordo com uma ou mais técnicas desta revelação.

**[0283]** Exemplo 1. Um método de codificação de dados de vídeo, o método compreendendo: determinar uma matriz modificado de amostras residuais; determinar para uma amostra residual modificada uma versão desquantificada da amostra residual; e adicionar a versão desquantificada da amostra residual a um valor de previsão para determinar um valor reconstruído.

**[0284]** Exemplo 2. O método do exemplo 1, compreendendo ainda: codificar os valores residuais desquantificados usando DPCM.

**[0285]** Exemplo 3. O método do exemplo 1, compreendendo ainda: codificar uma versão de bit deslocado dos valores residuais desquantificados usando DPCM.

**[0286]** Exemplo 4. O método de qualquer um dos exemplos 1-3, sendo que o método é realizado para um modo de intra-previsão quase vertical

**[0287]** Exemplo 5. O método de qualquer um dos exemplos 1-3, sendo que o método é realizado para um modo de intra-previsão quase horizontal.

**[0288]** Exemplo 6. O método de qualquer um dos exemplos 1-5, compreendendo ainda: desabilitar a ocultação de dados de sinal.

**[0289]** Exemplo 7. O método de qualquer um dos exemplos 1-5, compreendendo ainda: desabilitar a ocultação de dados de sinal baseado no tipo de bloco.

**[0290]** Exemplo 8. O método de qualquer um dos exemplos 1-5, compreendendo ainda: desabilitar seletivamente a ocultação de dados de sinal.

**[0291]** Exemplo 9. O método de qualquer um dos exemplos 1-8, compreendendo ainda qualquer técnica descrita nesta revelação.

**[0292]** Exemplo 10. Um método de codificação de dados de vídeo, o método compreendendo: determinar uma matriz modificado de amostras residuais; determinar para uma amostra residual modificada uma versão quantificada da amostra residual; e sinalizar em um bitstream codificado a versão quantificada da amostra residual.

**[0293]** Exemplo 11. O método do exemplo 10, sendo que o método é realizado para um modo de intra-previsão quase vertical.

**[0294]** Exemplo 12. O método do exemplo 10, sendo que o método é realizado para um modo de intra-previsão quase horizontal.

**[0295]** Exemplo 13. O método de qualquer um dos exemplos 10-12, compreendendo ainda qualquer técnica descrita nesta revelação.

**[0296]** Exemplo 14. O método de qualquer um dos exemplos 10-13 ou combinações dos mesmos, sendo que o método é realizado por um codificador e sendo que codificação refere-se à codificação (encoding).

**[0297]** Exemplo 15. O método de qualquer um dos exemplos 1 - 9 ou combinações dos mesmos, sendo que o

método é realizado por um decodificador e sendo que codificação refere-se à decodificação.

**[0298]** Exemplo 16. Um sistema configurado para realizar o método de qualquer um dos exemplos 1 - 15 ou combinações dos mesmos.

**[0299]** Exemplo 17. Uma mídia de armazenamento legível por computador não transitória que armazena as instruções que quando executadas fazem com que um ou mais processadores realize o método de qualquer um dos exemplos 1 - 13 ou combinações dos mesmos.

**[0300]** Exemplo 18. Um dispositivo de codificação de vídeo configurado para realizar o método de qualquer um dos exemplos 10 - 13 ou combinações dos mesmos.

**[0301]** Exemplo 19. Um dispositivo de decodificação de vídeo configurado para realizar o método de qualquer um dos exemplos 1 - 9 ou combinações dos mesmos.

**[0302]** Exemplo 20. Um dispositivo de codificação de vídeo compreendendo meios para realizar as etapas do método de qualquer um dos exemplos 10-13 ou combinações dos mesmos.

**[0303]** Exemplo 21. Um dispositivo de decodificação de vídeo compreendendo meios para realizar as etapas do método de qualquer um dos exemplos 1 - 9 ou combinações dos mesmos.

**[0304]** Exemplo 22. Qualquer dispositivo ou método descrito nesta revelação incluindo nesta revelação.

**[0305]** Os parágrafos a seguir fornecem uma quarta série de exemplos de acordo com uma ou mais técnicas desta revelação.

**[0306]** Exemplo 1. Um método para decodificar dados de vídeo, o método compreendendo: receber um bloco de dados de vídeo codificado usando a codificação

sem perdas e intra-previsão; reconstruir amostras residuais a partir do bloco codificado sem perdas dos dados de vídeo de acordo com um processo de modulação de código de pulso diferencial residual (DPCM); e realizar a intra-previsão de acordo com um modo de intra-previsão usando as amostras residuais para produzir amostras de vídeo reconstruídas, sendo que o modo de intra-previsão não é um de um modo de intra-previsão vertical e um modo de intra-previsão horizontal.

**[0307]** Exemplo 2. O método do exemplo 1, sendo que o modo de intra-previsão é um de um modo de intra-previsão quase vertical e um modo de intra-previsão quase horizontal.

**[0308]** Exemplo 3. O método do exemplo 2, sendo que o modo de intra-previsão quase vertical é um dos modos de intra-previsão 22 a 30, e sendo que o modo de intra-previsão quase horizontal é um dos modos de intra-previsão 6 a 14 como os intra modos são descritos em HEVC Working Draft 9.

**[0309]** Exemplo 4. O método do exemplo 2, sendo que o processo de DPCM residual é um processo de DPCM residual vertical para modos de intra-previsão quase verticais, e sendo que o processo de DPCM residual é um processo de DPCM residual horizontal para modos de intra-previsão quase horizontais.

**[0310]** Exemplo 5. O método do exemplo 1, sendo que o modo de intra-previsão é um modo de intra-previsão de baixo-direita diagonal, e sendo que reconstruir amostras residuais de acordo com o processo de DPCM residual compreende reconstruir amostras residuais de

$$\text{acordo com a equação } r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 1 \leq i \leq (M-1), j=0 \\ \tilde{r}_{i,j} + r_{i-1,j-1}, & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

onde  $r$  é uma amostra residual reconstruída,  $\tilde{r}$  é uma amostra do bloco codificado sem perdas dos dados de vídeo,  $M$  e  $N$  definem um tamanho do bloco de dados de vídeo, e  $i$  e  $j$  definem um local de uma amostra dentro do bloco de dados de vídeo.

**[0311]** Exemplo 6. O método do exemplo 1, sendo que o modo de intra-previsão é um modo de intra-previsão de baixo-esquerda diagonal, e sendo que reconstruir amostras residuais de acordo com o processo de DPCM residual compreende reconstruir amostras residuais de acordo com a equação:

$$r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 1 \leq i \leq (M-1), j=(N-1) \\ \tilde{r}_{i,j} + r_{i-1,j+1}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-2) \end{cases}$$

onde  $r$  é uma amostra residual reconstruída,  $\tilde{r}$  é uma amostra do bloco codificado sem perdas dos dados de vídeo,  $M$  e  $N$  definem um tamanho do bloco de dados de vídeo, e  $i$  e  $j$  definem um local de uma amostra dentro do bloco dos dados de vídeo.

**[0312]** Exemplo 7. O método do exemplo 1, sendo que o modo de intra-previsão é um modo de intra-previsão vertical diagonal, e sendo que reconstruir amostras residuais de acordo com o processo de DPCM residual compreende reconstruir amostras residuais de acordo com a equação:

$$r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i=(M-1), 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 0 \leq i \leq (M-2), j=0 \\ \tilde{r}_{i,j} + r_{i+1,j-1}, & 0 \leq i \leq (M-2), 1 \leq j \leq (N-1) \end{cases}$$

onde  $r$  é uma amostra residual reconstruída,  $\tilde{r}$  é uma amostra do bloco codificado sem perdas dos dados de vídeo,  $M$  e  $N$  definem um tamanho do bloco de dados de vídeo,



e  $i$  e  $j$  definem um local de uma amostra dentro do bloco dos dados de vídeo.

**[0313]** Exemplo 8. O método de qualquer combinação dos exemplos 1 a 7.

**[0314]** Exemplo 9. O método do exemplo 1, sendo que o modo de intra-previsão é um modo de intra-previsão quase vertical, o método compreendendo ainda: não reconstruir amostras residuais de acordo com o processo de DPCM residual para uma fileira do bloco codificado sem perdas de dados de vídeo.

**[0315]** Exemplo 10. O método do exemplo 1, sendo que o modo de intra-previsão é um modo de intra-previsão quase horizontal, o método compreendendo ainda: não reconstruir amostras residuais de acordo com o processo de DPCM residual para uma primeira coluna do bloco codificado sem perdas de dados de vídeo.

**[0316]** Exemplo 11. O método do exemplo 1, sendo que o modo de intra-previsão é um de um modo de intra-previsão DC e um modo de intra-previsão plano, e sendo que reconstruir amostras residuais de acordo com os processos de DPCM residuais compreende reconstruir amostras residuais de acordo com um de um processo de DPCM residual vertical e um processo de DPCM horizontal.

**[0317]** Exemplo 12. O método do exemplo 1, sendo que o modo de intra-previsão é um de um modo de intra-previsão DC e um modo de intra-previsão plano, e sendo que reconstruir amostras residuais de acordo com os processos de DPCM residuais compreende reconstruir amostras residuais de acordo com ambos um processo de DPCM residual vertical e um processo de DPCM horizontal.

**[0318]** Exemplo 13. O método do exemplo 1, sendo que o modo de intra-previsão é um modo de intra-previsão plano, e sendo que reconstruir amostras residuais

de acordo com os processos de DPCM residuais compreende reconstruir amostras residuais de acordo com um processo de DPCM diagonal, um processo de DPCM horizontal, e um processo de DPCM vertical.

**[0319]** Exemplo 14. Um método para codificar dados de vídeo, o método compreendendo: receber um bloco de dados de vídeo; realizar a intra-previsão no bloco dos dados de vídeo de acordo com um modo de intra-previsão para produzir um bloco preditivo de amostras e amostras residuais, sendo que o modo de intra-previsão não é um de um modo de intra-previsão vertical e um modo de intra-previsão horizontal; e gerar um bloco codificado sem perdas de dados de vídeo a partir das amostras residuais usando um processo de modulação de código de pulso diferencial residual (DPCM).

**[0320]** Exemplo 15. O método do exemplo 14, sendo que o modo de intra-previsão é um de um modo de intra-previsão quase vertical e um modo de intra-previsão quase horizontal.

**[0321]** Exemplo 16. O método do exemplo 15, sendo que o modo de intra-previsão quase vertical é um dos modos de intra-previsão 22 a 30, e sendo que o modo de intra-previsão quase horizontal é um dos modos de intra-previsão 6 a 14.

**[0322]** Exemplo 17. O método do exemplo 15, sendo que o processo de DPCM residual é um processo de DPCM residual vertical para modos de intra-previsão quase verticais, e sendo que o processo de DPCM residual é um processo de DPCM residual horizontal para modos de intra-previsão quase horizontais.

**[0323]** Exemplo 18. O método do exemplo 14, sendo que o modo de intra-previsão é um modo de intra-previsão de baixo-direita diagonal, e sendo que gerar o

bloco codificado sem perdas dos dados de vídeo a partir das amostras residuais usando o processo de DPCM residual compreende gerar o bloco codificado sem perdas dos dados de vídeo de acordo com a equação:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j=0 \\ r_{i,j} - r_{i-1,j-1}, & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

onde  $r$  é uma amostra residual,  $\tilde{r}$  é uma amostra do bloco codificado sem perdas dos dados de vídeo,  $M$  e  $N$  definem um tamanho do bloco de dados de vídeo, e  $i$  e  $j$  definem um local de uma amostra dentro do bloco dos dados de vídeo.

**[0324]** Exemplo 19. O método do exemplo 14, sendo que o modo de intra-previsão é um modo de intra-previsão de baixo-direita diagonal, e sendo que gerar o bloco codificado sem perdas dos dados de vídeo a partir das amostras residuais usando o processo de DPCM residual compreende gerar o bloco codificado sem perdas dos dados de vídeo de acordo com a equação:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j=(N-1) \\ r_{i,j} - r_{i-1,j+1}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-2) \end{cases}$$

onde  $r$  é uma amostra residual,  $\tilde{r}$  é uma amostra do bloco codificado sem perdas dos dados de vídeo,  $M$  e  $N$  definem um tamanho do bloco de dados de vídeo, e  $i$  e  $j$  definem um local de uma amostra dentro do bloco dos dados de vídeo.

**[0325]** Exemplo 20. O método do exemplo 14, sendo que o modo de intra-previsão é um modo de intra-previsão de baixo-direita diagonal, e sendo que gerar o bloco codificado sem perdas dos dados de vídeo a partir das amostras residuais usando o processo de DPCM residual

compreende gerar o bloco codificado sem perdas dos dados de vídeo de acordo com a equação:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = (M-1), 0 \leq j \leq (N-1) \\ r_{i,j}, & 0 \leq i \leq (M-2), j = 0 \\ r_{i,j} - r_{i+1,j-1}, & 0 \leq i \leq (M-2), 1 \leq j \leq (N-1) \end{cases}$$

onde  $r$  é uma amostra residual,  $\tilde{r}$  é uma amostra do bloco codificado sem perdas dos dados de vídeo,  $M$  e  $N$  definem um tamanho do bloco de dados de vídeo, e  $i$  e  $j$  definem um local de uma amostra dentro do bloco dos dados de vídeo.

**[0326]** Exemplo 21. O método de qualquer combinação dos exemplos 14 a 20.

**[0327]** Exemplo 22. O método do exemplo 14, sendo que o modo de intra-previsão é um modo de intra-previsão quase vertical, o método compreendendo ainda: não gerar o bloco codificado sem perdas de dados de vídeo usando o processo de DPCM residual para uma fileira do bloco de dados de vídeo.

**[0328]** Exemplo 23. O método do exemplo 14, sendo que o modo de intra-previsão é um modo de intra-previsão quase horizontal, o método compreendendo ainda: não gerar o bloco codificado sem perdas de dados de vídeo usando o processo de DPCM residual para uma primeira coluna do bloco de dados de vídeo.

**[0329]** Exemplo 24. O método do exemplo 14, sendo que o modo de intra-previsão é um de um modo de intra-previsão DC e um modo de intra-previsão plano, e sendo que gerar o bloco codificado sem perdas dos dados de vídeo a partir das amostras residuais usando os processos de DPCM residuais compreende gerar o bloco codificado sem perdas de dados de vídeo a partir das amostras residuais de acordo com um de um processo de DPCM residual vertical e um processo de DPCM horizontal.

**[0330]** Exemplo 25. O método do exemplo 14, sendo que o modo de intra-previsão é um de um modo de intra-previsão DC e um modo de intra-previsão plano, e sendo que gerar o bloco codificado sem perdas de dados de vídeo a partir das amostras residuais usando os processos de DPCM residuais compreende gerar o bloco codificado sem perdas dos dados de vídeo a partir das amostras residuais de acordo com ambos um processo de DPCM residual vertical e um processo de DPCM horizontal.

**[0331]** Exemplo 26. O método do exemplo 14, sendo que o modo de intra-previsão é um modo de intra-previsão plano, e sendo que gerar o bloco codificado sem perdas dos dados de vídeo a partir das amostras residuais usando os processos de DPCM residuais compreende gerar o bloco codificado sem perdas dos dados de vídeo a partir das amostras residuais de acordo com um processo de DPCM diagonal, um processo de DPCM horizontal, e um processo de DPCM vertical.

**[0332]** Exemplo 27. Um aparelho configurado para decodificar dados de vídeo, o aparelho compreendendo: meios para receber um bloco de dados de vídeo codificado usando codificação sem perdas e intra-previsão; meios para reconstruir amostras residuais a partir do bloco codificado sem perdas dos dados de vídeo de acordo com um processo de modulação de código de pulso diferencial residual (DPCM); e meios para realizar a intra-previsão de acordo com um modo de intra-previsão usando as amostras residuais para produzir amostras de vídeo reconstruídas, sendo que o modo de intra-previsão não é um de um modo de intra-previsão vertical e um modo de intra-previsão horizontal.

**[0333]** Exemplo 28. O aparelho do exemplo 27, sendo que o modo de intra-previsão é um de um modo de

intra-previsão quase vertical e um modo de intra-previsão quase horizontal.

**[0334]** Exemplo 29. O aparelho do exemplo 28, sendo que o modo de intra-previsão quase vertical é um dos modos de intra-previsão 22 a 30, e sendo que o modo de intra-previsão quase horizontal é um dos modos de intra-previsão 6 a 14.

**[0335]** Exemplo 30. O aparelho do exemplo 28, sendo que o processo de DPCM residual é um processo de DPCM residual vertical para modos de intra-previsão quase verticais, e sendo que o processo de DPCM residual é um processo de DPCM residual horizontal para modos de intra-previsão quase horizontais.

**[0336]** Exemplo 31. O aparelho do exemplo 27, sendo que o modo de intra-previsão é um modo de intra-previsão de baixo-direita diagonal, e sendo que os meios para reconstruir amostras residuais de acordo com o processo de DPCM residual compreende meios para reconstruir amostras residuais de acordo com a equação:

$$r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 1 \leq i \leq (M-1), j=0 \\ \tilde{r}_{i,j} + r_{i-1,j-1}, & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

onde  $r$  é uma amostra residual reconstruída,  $\tilde{r}$  é uma amostra do bloco codificado sem perdas dos dados de vídeo,  $M$  e  $N$  definem um tamanho do bloco de dados de vídeo, e  $i$  e  $j$  definem um local de uma amostra dentro do bloco dos dados de vídeo.

**[0337]** Exemplo 32. O aparelho do exemplo 27, sendo que o modo de intra-previsão é um modo de intra-previsão de baixo-esquerda diagonal, e sendo que os meios para reconstruir amostras residuais de acordo com o processo de DPCM residual compreende meios para reconstruir amostras residuais de acordo com a equação:

$$r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 1 \leq i \leq (M-1), j = (N-1) \\ \tilde{r}_{i,j} + r_{i-1,j+1}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-2) \end{cases}$$

onde  $r$  é uma amostra residual reconstruída,  $\tilde{r}$  é uma amostra do bloco codificado sem perdas dos dados de vídeo,  $M$  e  $N$  definem um tamanho do bloco de dados de vídeo, e  $i$  e  $j$  definem um local de uma amostra dentro do bloco dos dados de vídeo.

**[0338]** Exemplo 33. O aparelho do exemplo 27, sendo que o modo de intra-previsão é um modo de intra-previsão vertical diagonal, e sendo que os meios para reconstruir amostras residuais de acordo com o processo de DPCM residual compreende meios para reconstruir amostras residuais de acordo com a equação:

$$r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i = (M-1), 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 0 \leq i \leq (M-2), j = 0 \\ \tilde{r}_{i,j} + r_{i+1,j-1}, & 0 \leq i \leq (M-2), 1 \leq j \leq (N-1) \end{cases}$$

onde  $r$  é uma amostra residual reconstruída,  $\tilde{r}$  é uma amostra do bloco codificado sem perdas dos dados de vídeo,  $M$  e  $N$  definem um tamanho do bloco de dados de vídeo, e  $i$  e  $j$  definem um local de uma amostra dentro do bloco dos dados de vídeo.

**[0339]** Exemplo 34. O aparelho de qualquer combinação dos exemplos 27 a 33.

**[0340]** Exemplo 35. O aparelho do exemplo 27, sendo que o modo de intra-previsão é um modo de intra-previsão quase vertical, o aparelho compreendendo ainda: meios para não reconstruir amostras residuais de acordo com o processo de DPCM residual para uma fileira do bloco codificado sem perdas dos dados de vídeo.

**[0341]** Exemplo 36. O aparelho do exemplo 27, sendo que o modo de intra-previsão é um modo de intra-

previsão quase horizontal, o aparelho compreendendo ainda: meios para não reconstruir amostras residuais de acordo com o processo de DPCM residual para uma primeira coluna do bloco codificado sem perdas dos dados de vídeo.

**[0342]** Exemplo 37. O aparelho do exemplo 27, sendo que o modo de intra-previsão é um de um modo de intra-previsão DC e um modo de intra-previsão plano, e sendo que os meios para reconstruir amostras residuais de acordo com os processos de DPCM residuais compreende meios para reconstruir amostras residuais de acordo com um de um processo de DPCM residual vertical e um processo de DPCM horizontal.

**[0343]** Exemplo 38. O aparelho do exemplo 27, sendo que o modo de intra-previsão é um de um modo de intra-previsão DC e um modo de intra-previsão plano, e sendo que os meios para reconstruir amostras residuais de acordo com os processos de DPCM residuais compreende meios para reconstruir amostras residuais de acordo com ambos um processo de DPCM residual vertical e um processo de DPCM horizontal.

**[0344]** Exemplo 39. O aparelho do exemplo 27, sendo que o modo de intra-previsão é um modo de intra-previsão plano, e sendo que os meios para reconstruir amostras residuais de acordo com os processos de DPCM residuais compreende meios para reconstruir amostras residuais de acordo com um processo de DPCM diagonal, um processo de DPCM horizontal, e um processo de DPCM vertical.

**[0345]** Exemplo 40. Um aparelho configurado para codificar dados de vídeo, o aparelho compreendendo: meios para receber um bloco de dados de vídeo; meios para realizar a intra-previsão no bloco dos dados de vídeo de acordo com um modo de intra-previsão para produzir amostras



residuais, sendo que o modo de intra-previsão não é um de um modo de intra-previsão vertical e um modo de intra-previsão horizontal; e meios para gerar um bloco codificado sem perdas dos dados de vídeo a partir das amostras residuais usando um processo de modulação de código de pulso diferencial residual (DPCM).

**[0346]** Exemplo 41. O aparelho do exemplo 40, sendo que o modo de intra-previsão é um de um modo de intra-previsão quase vertical e um modo de intra-previsão quase horizontal.

**[0347]** Exemplo 42. O aparelho do exemplo 41, sendo que o modo de intra-previsão quase vertical é um dos modos de intra-previsão 22 a 30, e sendo que o modo de intra-previsão quase horizontal é um dos modos de intra-previsão 6 a 14.

**[0348]** Exemplo 43. O aparelho do exemplo 41, sendo que o processo de DPCM residual é um processo de DPCM residual vertical para modos de intra-previsão quase verticais, e sendo que o processo de DPCM residual é um processo de DPCM residual horizontal para modos de intra-previsão quase horizontais.

**[0349]** Exemplo 44. O aparelho do exemplo 40, sendo que o modo de intra-previsão é um modo de intra-previsão de baixo-direita diagonal, e sendo que os meios para gerar o bloco codificado sem perdas dos dados de vídeo a partir das amostras residuais usando o processo de DPCM residual compreende meios para gerar o bloco codificado sem perdas dos dados de vídeo de acordo com a equação:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j=0 \\ r_{i,j} - r_{i-1,j-1}, & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

onde  $r$  é uma amostra residual,  $\tilde{r}$  é uma amostra do bloco codificado sem perdas dos dados de vídeo,  $M$  e  $N$

definem um tamanho do bloco de dados de vídeo, e  $i$  e  $j$  definem um local de uma amostra dentro do bloco dos dados de vídeo.

**[0350]** Exemplo 45. O aparelho do exemplo 40, sendo que o modo de intra-previsão é um modo de intra-previsão de baixo-direita diagonal, e sendo que os meios para gerar o bloco codificado sem perdas dos dados de vídeo a partir das amostras residuais usando o processo de DPCM residual compreende meios para gerar o bloco codificado sem perdas dos dados de vídeo de acordo com a equação:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j=(N-1) \\ r_{i,j} - r_{i-1,j+1}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-2) \end{cases}$$

onde  $r$  é uma amostra residual,  $\tilde{r}$  é uma amostra do bloco codificado sem perdas dos dados de vídeo,  $M$  e  $N$  definem um tamanho do bloco de dados de vídeo, e  $i$  e  $j$  definem um local de uma amostra dentro do bloco dos dados de vídeo.

**[0351]** Exemplo 46. O aparelho do exemplo 40, sendo que o modo de intra-previsão é um modo de intra-previsão de baixo-direita diagonal, e sendo que os meios para gerar o bloco codificado sem perdas dos dados de vídeo a partir das amostras residuais usando o processo de DPCM residual compreende meios para gerar o bloco codificado sem perdas dos dados de vídeo de acordo com a equação:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=(M-1), 0 \leq j \leq (N-1) \\ r_{i,j}, & 0 \leq i \leq (M-2), j=0 \\ r_{i,j} - r_{i+1,j-1}, & 0 \leq i \leq (M-2), 1 \leq j \leq (N-1) \end{cases}$$

onde  $r$  é uma amostra residual,  $\tilde{r}$  é uma amostra do bloco codificado sem perdas dos dados de vídeo,  $M$  e  $N$  definem um tamanho do bloco de dados de vídeo, e  $i$  e  $j$

definem um local de uma amostra dentro do bloco dos dados de vídeo.

**[0352]** Exemplo 47. O aparelho de qualquer combinação dos exemplos 40 a 46.

**[0353]** Exemplo 48. O aparelho do exemplo 40, sendo que o modo de intra-previsão é um modo de intra-previsão quase vertical, o aparelho compreendendo ainda: meios para não gerar o bloco codificado sem perdas dos dados de vídeo usando o processo de DPCM residual para uma fileira do bloco de dados de vídeo.

**[0354]** Exemplo 49. O aparelho do exemplo 40, sendo que o modo de intra-previsão é um modo de intra-previsão quase horizontal, o aparelho compreendendo ainda: meios para não gerar o bloco codificado sem perdas dos dados de vídeo usando o processo de DPCM residual para uma primeira coluna do bloco de dados de vídeo.

**[0355]** Exemplo 50. O aparelho do exemplo 40, sendo que o modo de intra-previsão é um de um modo de intra-previsão DC e um modo de intra-previsão plano, e sendo que os meios para gerar o bloco codificado sem perdas dos dados de vídeo a partir das amostras residuais usando os processos de DPCM residuais compreende meios para gerar o bloco codificado sem perdas dos dados de vídeo a partir das amostras residuais de acordo com um de um processo de DPCM residual vertical e um processo de DPCM horizontal.

**[0356]** Exemplo 51. O aparelho do exemplo 40, sendo que o modo de intra-previsão é um de um modo de intra-previsão DC e um modo de intra-previsão plano, e sendo que os meios para gerar o bloco codificado sem perdas dos dados de vídeo a partir das amostras residuais usando os processos de DPCM residuais compreende meios para gerar o bloco codificado sem perdas dos dados de vídeo a partir

das amostras residuais de acordo com ambos um processo de DPCM residual vertical e um processo de DPCM horizontal.

**[0357]** Exemplo 52. O aparelho do exemplo 40, sendo que o modo de intra-previsão é um modo de intra-previsão plano, e sendo que os meios para gerar o bloco codificado sem perdas dos dados de vídeo a partir das amostras residuais usando os processos de DPCM residuais compreende meios para gerar o bloco codificado sem perdas dos dados de vídeo a partir das amostras residuais de acordo com um processo de DPCM diagonal, um processo de DPCM horizontal, e um processo de DPCM vertical.

**[0358]** Exemplo 53. Um decodificador de vídeo configurado para realizar qualquer combinação dos métodos dos exemplos 1 a 13.

**[0359]** Exemplo 54. Um codificador de vídeo configurado para realizar qualquer combinação dos métodos dos exemplos 14 a 26.

**[0360]** Exemplo 55. Uma mídia de armazenamento legível por computador que armazena as instruções que, quando executadas, fazem com que um ou mais processadores de um dispositivo configurado para decodificar dados de vídeo para realizar qualquer combinação dos métodos dos exemplos 1 a 13.

**[0361]** Exemplo 56. Uma mídia de armazenamento legível por computador que armazena as instruções que, quando executadas, fazem com que um ou mais processadores de um dispositivo configurado para codificar os dados de vídeo para realizar de qualquer combinação dos métodos dos exemplos 14 a 26.

**[0362]** Os parágrafos a seguir fornecem uma quinta série de exemplos de acordo com uma ou mais técnicas desta revelação.

**[0363]** Exemplo 1. Um método de codificação de dados de vídeo, o método compreendendo: gerar amostras de previsão para a intra-codificação horizontal de um bloco de dados de vídeo, sendo que para cada coluna do bloco de dados de vídeo, as amostras de previsão incluem um termo gradiente.

**[0364]** Exemplo 2. O método do exemplo 1, sendo que o termo gradiente para as amostras de previsão em uma fileira inicial são dados por  $\left((P_{-1,j} - P_{-1,j-1}) \gg 1\right)$ .

**[0365]** Exemplo 3. O método de qualquer um dos exemplos 1 - 2, sendo que uma amostra de previsão para  $P_{0,j}, 0 \leq j \leq N-1$  é:  $\text{Clip}\left(P_{0,j-1} + \left((P_{-1,j} - P_{-1,j-1}) \gg 1\right)\right)$ .

**[0366]** Exemplo 4. O método de qualquer um dos exemplos 1 - 3, sendo que as amostras de previsão para  $P_{i,j}, 0 \leq i \leq M-1, 0 \leq j \leq N-1$  são dadas por  $\text{Clip}\left(P_{i,j-1} + \left((P_{i-1,j} - P_{i-1,j-1}) \gg 1\right)\right)$ .

**[0367]** Exemplo 5. O método de qualquer um dos exemplos 5 - 8, sendo que o método é aplicado para a intra-codificação horizontal sem perda.

**[0368]** Exemplo 6. Um método de codificação de dados de vídeo, o método compreendendo: gerar amostras de previsão para a intra-codificação vertical de um bloco de dados de vídeo, sendo que para cada fileira do bloco de dados de vídeo, as amostras de previsão incluem um termo gradiente.

**[0369]** Exemplo 7. O método do exemplo 6, sendo que o termo gradiente para as amostras de previsão em uma coluna inicial são dadas como  $\left((P_{i,-1} - P_{i-1,-1}) \gg 1\right)$ .

[0370] Exemplo 8. O método de qualquer um dos exemplos 6 - 7, sendo que uma amostra de previsão para

$$P_{i,0}, 0 \leq i \leq M-1 \text{ é } \text{Clip} \left( P_{i-1,0} + \left( (P_{i-1} - P_{i-1,1}) \gg 1 \right) \right).$$

[0371] Exemplo 9. O método de qualquer um dos exemplos 6 - 8, sendo que as amostras de previsão para  $P_{i,j}, 0 \leq i \leq M-1, 0 \leq j \leq N-1$  são dadas por

$$\text{Clip} \left( P_{i-1,j} + \left( (P_{i,j-1} - P_{i-1,j-1}) \gg 1 \right) \right)$$

[0372] Exemplo 10. O método de qualquer um dos exemplos 6 - 9, sendo que o método é aplicado para a intra-codificação vertical sem fio.

[0373] Exemplo 11. O método de qualquer combinação dos exemplos 1 - 10.

[0374] Exemplo 12. O método de qualquer um dos exemplos 1 - 10 ou combinações dos mesmos, sendo que o método é realizado por um codificador e sendo que codificação refere-se à codificação (encoding).

[0375] Exemplo 13. O método de qualquer um dos exemplos 1 - 10 ou combinações dos mesmos, sendo que o método é realizado por um decodificador e sendo que codificação refere-se à decodificação.

[0376] Exemplo 14. Um sistema configurado para realizar o método de qualquer um dos exemplos 1 - 10 ou combinações dos mesmos.

[0377] Exemplo 15. Uma mídia de armazenamento legível por computador não transitória que armazena as instruções que quando executadas fazem com que um ou mais processadores realizem o método de qualquer um dos exemplos 1 - 10 ou combinações dos mesmos.

**[0378]** Exemplo 16. Um dispositivo de codificação de vídeo configurado para realizar o método de qualquer um dos exemplos 1 - 10 ou combinações dos mesmos.

**[0379]** Exemplo 17. Um dispositivo de decodificação de vídeo configurado para realizar o método de qualquer um dos exemplos 1 - 10 ou combinações dos mesmos.

**[0380]** Exemplo 18. Um dispositivo de codificação de vídeo compreendendo meios para realizar as etapas do método de qualquer um dos exemplos 1 - 10 ou combinações dos mesmos.

**[0381]** Exemplo 19. Um dispositivo de decodificação de vídeo compreendendo meios para realizar as etapas do método de qualquer um dos exemplos 1 - 10 ou combinações dos mesmos.

**[0382]** Em um ou mais exemplos, as funções descritas na presente invenção podem ser implementadas em hardware, software, firmware, ou em qualquer combinação dos mesmos. Se implementadas em software, as funções podem ser armazenadas em ou transmitidas sobre, as uma ou mais instruções ou código, uma mídia legível por computador e executada por uma unidade de processamento baseada em hardware. A mídia legível por computador pode incluir a mídia de armazenamento legível por computador, que corresponde a uma mídia tangível como mídia de armazenamento de dados, ou meio de comunicação incluindo qualquer mídia que facilita a transferência de um programa de computador de um lugar para outro, ex., de acordo com um protocolo de comunicação. Desta maneira, a mídia legível por computador geralmente pode corresponder a (1) mídia de armazenamento legível por computador tangível que é não-transitória ou (2) um meio de comunicação como um sinal ou onda portadora. A mídia de armazenamento de dados pode

qualquer mídia disponível que pode ser acessada por um ou mais computadores ou um ou mais processadores para recuperar as instruções, código e/ou estruturas de dados para implementação das técnicas descritas nesta revelação. Um produto de programa de computador pode incluir uma mídia legível por computador.

**[0383]** Por meio de exemplo, e não limitação, tal mídia de armazenamento legível por computador pode compreender RAM, ROM, EEPROM, CD-ROM ou outro armazenamento em disco ótico, armazenamento em disco magnético, ou outros dispositivos de armazenamento magnéticos, memória flash, ou qualquer outro meio que pode ser usado para armazenar o código de programa desejado na forma de instruções ou estruturas de dados e que pode ser acessada por um computador. Também, qualquer conexão é adequadamente chamada de mídia legível por computador. Por exemplo, se as instruções são transmitidas a partir de um site da web, servidor, ou outra fonte remota usando um cabo coaxial, cabo de fibra ótica, par trançado, linha de assinante digital (DSL), ou tecnologias remotas como infravermelho, rádio, e micro-onda, então o cabo coaxial, cabo de fibra ótica, par trançado, DSL, ou tecnologias sem fio como infravermelho, rádio, e micro-onda são incluídos na definição de mídia. Deve-se compreender, no entanto, que mídia de armazenamento legível por computador e mídia de armazenamento de dados não incluem conexões, ondas portadoras, sinais, ou outra mídia transiente, mas são, ao invés, direcionadas à mídia de armazenamento não-transiente, tangível. Disco e disquete, são usados na presente invenção, incluem disco compacto (CD), disco a laser, disco ótico, disco versátil digital (DVD), disquete e disco Blu-ray, onde disquetes normalmente reproduzem dados magneticamente, enquanto discos reproduzem dados



oticamente com lasers. Combinações dos acima também devem ser incluídas dentro do escopo de mídia legível por computador.

**[0384]** As instruções podem ser executadas por um ou mais processadores, como um ou mais processadores de sinal digital (DSPs), microprocessadores de uso geral, circuito integrados de aplicação específica (ASICs), arranjos lógicos programáveis em campo (FPGAs), ou outro conjunto de circuito integrado ou lógico discreto equivalente. Dessa forma, o termo "processador", como usado na presente invenção pode se referir a qualquer uma das estruturas anteriores ou qualquer outra estrutura adequada para implementação das técnicas descritas na presente invenção. Além disso, em alguns aspectos, a funcionalidade descrita na presente invenção pode ser fornecida dentro de módulos de hardware e/ou software dedicados configurados para codificação e decodificação, ou incorporados em um codec combinado. Também, as técnicas poderiam ser completamente implementadas em um ou mais circuitos ou elementos lógicos.

**[0385]** As técnicas desta revelação podem ser implementadas em uma ampla variedade de dispositivos ou aparelhos, incluindo um aparelho remoto, um circuito integrado (IC) ou um conjunto de ICs (ex., um chip set). Vários componentes, módulos, ou unidades são descritos nesta revelação para enfatizar os aspectos funcionais dos dispositivos configurados para realizar as técnicas reveladas, mas não necessariamente requerem a realização por diferentes unidades de hardware. Ao invés, como descrito acima, várias unidades podem ser combinadas em uma unidade de hardware de codec ou fornecidas por uma coleção de unidades de hardware inter-operativas, incluindo um ou

mais processadores como descrito acima, em conjunto com software e/ou firmware adequado.

**[0386]** Vários exemplos foram descritos. Estes e outros exemplos estão dentro do escopo das seguintes reivindicações.

### REIVINDICAÇÕES

1. Método de decodificação de dados de vídeo, **caracterizado pelo fato de que** compreende:

obter elementos de sintaxe de um bitstream que inclui uma sequência de bits que forma uma representação codificada dos dados de vídeo,

sendo que obter os elementos de sintaxe do bitstream compreende determinar que a ocultação de dados de sinal está desabilitada para um bloco atual com base (i) no bloco atual sendo gerado usando codificação de perda sem a aplicação de uma transformada aos dados residuais e (ii) no bloco atual sendo intra-previsto usando um modo de intra-previsão no qual uma técnica de modulação de código de pulso diferencial residual (DPCM) é usada; e

reconstruir uma imagem dos dados de vídeo com base pelo menos em parte nos elementos de sintaxe obtidos a partir do bitstream.

2. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** obter os elementos de sintaxe do bitstream compreende ainda:

quando a ocultação de dados de sinal está desabilitada para o bloco atual, obter, a partir do bitstream, para cada respectivo valor residual significativo no bloco atual, um elemento de sintaxe indicando se o respectivo valor residual significativo é positivo ou negativo, e

quando a ocultação de dados de sinal não está desabilitada para o bloco atual, não obter a partir do bitstream um elemento de sintaxe indicando se um valor de um valor residual significativo no bloco atual é positivo ou negativo.

3. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** obter os elementos de sintaxe

do bitstream compreende obter, a partir do bitstream, uma indicação explícita de que a ocultação de dados de sinal está habilitada para o bloco atual.

4. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** determinar que a ocultação de dados de sinal está desabilitada para o bloco atual compreende determinar que a ocultação de dados de sinal está desabilitada para o bloco atual com base:

no bloco atual sendo codificado sem a aplicação da transformada aos dados residuais e,

no bloco atual sendo intra-codificado usando um modo de intra-previsão DC ou um modo de intra-previsão plano.

5. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** reconstruir a imagem compreende:

gerar um bloco de valores residuais que inclui o bloco atual, sendo que o bloco é um bloco de pulso de transformada; e

para  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ , onde M é uma altura do bloco e N é a largura do bloco, o método compreende:

calcular um valor residual reconstruído  $Q(r_{i,j})$  correspondente a um valor residual  $r_{i,j}$  no bloco,

sendo que, se o bloco é codificado usando um modo de intra-previsão vertical,  $Q(r_{i,j})$  é definido como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}$$

sendo que  $Q(\tilde{r}_{i,j})$  denota uma versão quantificada de um valor residual modificado  $\tilde{r}_{i,j}$ , o valor residual modificado  $\tilde{r}_{i,j}$  é uma versão modificada do valor residual  $r_{i,j}$ , e  $Q(r_{i-1,j})$  é um valor residual reconstruído

correspondente a um valor residual uma coluna à esquerda do valor residual  $r_{i,j}$ , e

sendo que, se o bloco é codificado usando um modo de intra-previsão horizontal,  $Q(r_{i,j})$  é definida como:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases},$$

sendo que  $Q(r_{i,j-1})$  é um valor residual reconstruído correspondente a um valor residual uma linha acima do valor residual  $r_{i,j}$ ; e

adicionar o valor residual reconstruído  $Q(r_{i,j})$  a um valor de previsão para reconstruir um valor de amostra.

6. Método de codificação de dados de vídeo, **caracterizado pelo fato de que** compreende:

gerar um bitstream que inclui uma sequência de bits que forma uma representação codificada dos dados de vídeo,

em que gerar o bitstream compreende determinar que a ocultação de dados de sinal está desabilitada para um bloco atual com base (i) no bloco atual sendo gerado utilizando codificação de perda sem a aplicação de uma transformada aos dados residuais e (ii) no bloco atual sendo intra-previsto usando um modo de intra-previsão no qual uma técnica de modulação de código de pulso diferencial residual (DPCM) é usada; e

emitir o bitstream.

7. Método, de acordo com a reivindicação 6, **caracterizado pelo fato de que** gerar o bitstream compreende:

quando a ocultação de dados de sinal está desabilitada para o bloco atual, sinalizar, no bitstream, para cada respectivo valor residual significativo no bloco atual, um elemento de sintaxe indicando se o respectivo valor residual significativo é positivo ou negativo, e

quando a ocultação de dados de sinal não está desabilitada para o bloco atual, não sinalizar, no bitstream,

um elemento de sintaxe indicando se um valor de um valor residual significativo no bloco atual é positivo ou negativo.

8. Método, de acordo com a reivindicação 6, **caracterizado pelo fato de que** gerar o bitstream compreende sinalizar, no bitstream, uma indicação explícita de que a ocultação de dados de sinal está habilitada para o bloco atual.

9. Método, de acordo com a reivindicação 6, **caracterizado pelo fato de que** determinar que a ocultação de dados de sinal está desabilitada para o bloco atual compreende determinar que a ocultação de dados de sinal está desabilitada para o bloco atual com base:

no bloco atual sendo codificado sem a aplicação da transformada aos dados residuais, e

no bloco atual sendo intra-codificado usando um modo de intra-previsão DC ou um modo de intra-previsão plano.

10. Método, de acordo com a reivindicação 1 ou 6, **caracterizado pelo fato de que** o modo de intra-previsão no qual a DPCM residual é aplicada é um modo de intra-previsão horizontal ou vertical.

11. Método, de acordo com a reivindicação 6, **caracterizado pelo fato de que** compreende ainda:

para  $0 \leq i \leq (M - 1)$  e  $0 \leq j \leq (N - 1)$ , onde M é uma altura de um bloco que inclui o bloco atual e N é a largura do bloco, sendo que o bloco é um bloco de pulso de transformada;

determinar um valor residual modificado  $\tilde{r}_{i,j}$  para um valor residual  $r_{i,j}$ ,

sendo que, se o bloco é codificado usando um modo de intra-previsão vertical,  $\tilde{r}_{i,j}$  é definido como:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N - 1), \\ r_{i,j} - Q(r_{(i-1),j}) & 1 \leq i \leq (M - 1), 0 \leq j \leq (N - 1) \end{cases},$$

sendo que  $Q(r_{(i-1),j})$  denota um valor residual reconstruído correspondente a um valor residual  $r_{i-1,j}$  uma coluna à esquerda do valor residual  $r_{i,j}$ , e

se o bloco é codificado usando um modo de intra-previsão horizontal,  $\tilde{r}_{i,j}$  é definido como:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), \quad j = 0 \\ r_{i,j} - Q(r_{i,(j-1)}) & 0 \leq i \leq (M-1), \quad 1 \leq j \leq (N-1) \end{cases},$$

sendo que  $Q(r_{i,(j-1)})$  denota um valor residual reconstruído correspondente a um valor residual  $r_{i,j-1}$  uma linha acima do valor residual  $r_{i,j}$ ; e

quantificar o valor residual modificado  $\tilde{r}_{i,j}$  para produzir um valor residual modificado quantificado  $Q(\tilde{r}_{i,j})$ .

12. Aparelho de decodificação de vídeo, **caracterizado pelo fato de que** compreende:

meios para obter elementos de sintaxe de um bitstream que inclui uma sequência de bits que forma uma representação codificada de dados de vídeo,

sendo que obter os elementos de sintaxe do bitstream compreende determinar que a ocultação de dados de sinal está desabilitada para um bloco atual com base (i) no bloco atual sendo gerado usando codificação de perda sem a aplicação de uma transformada aos dados residuais e (ii) no bloco atual sendo intra-previsto usando um modo de intra-previsão no qual uma técnica de modulação de código de pulso diferencial residual (DPCM) é usada; e

meios para reconstruir uma imagem dos dados de vídeo com base pelo menos em parte nos elementos de sintaxe obtidos a partir do bitstream.

13. Aparelho de codificação de vídeo, **caracterizado pelo fato de que** compreende:

meios para gerar um bitstream que inclui uma sequência de bits que forma uma representação codificada de dados de vídeo,

sendo que gerar o bitstream compreende determinar que a ocultação de dados de sinal está desabilitada para um bloco atual com base (i) no bloco atual sendo gerado usando codificação de perda sem a aplicação de uma transformada aos dados residuais e (ii) no bloco atual sendo intra-previsto usando um modo de intra-previsão no qual uma técnica de modulação de código de pulso diferencial residual (DPCM) é usada; e

meios para emitir o bitstream.

14. Memória caracterizada pelo fato de que possui instruções nela armazenadas que, quando executadas, fazem com que um ou mais processadores realizem o método conforme definido em qualquer uma das reivindicações de 1 a 11.

15. Aparelho de decodificação de vídeo, de acordo com a reivindicação 12, ou aparelho de codificação de vídeo, de acordo com a reivindicação 13, caracterizado pelo fato de que compreende adicionalmente pelo menos um de:

um circuito integrado;

um microprocessador;

um dispositivo de comunicação sem fio que inclui um ou mais processadores;

um monitor configurado para exibir uma imagem decodificada que contém o bloco atual; e

uma câmera configurada para capturar uma imagem que contém o bloco atual.



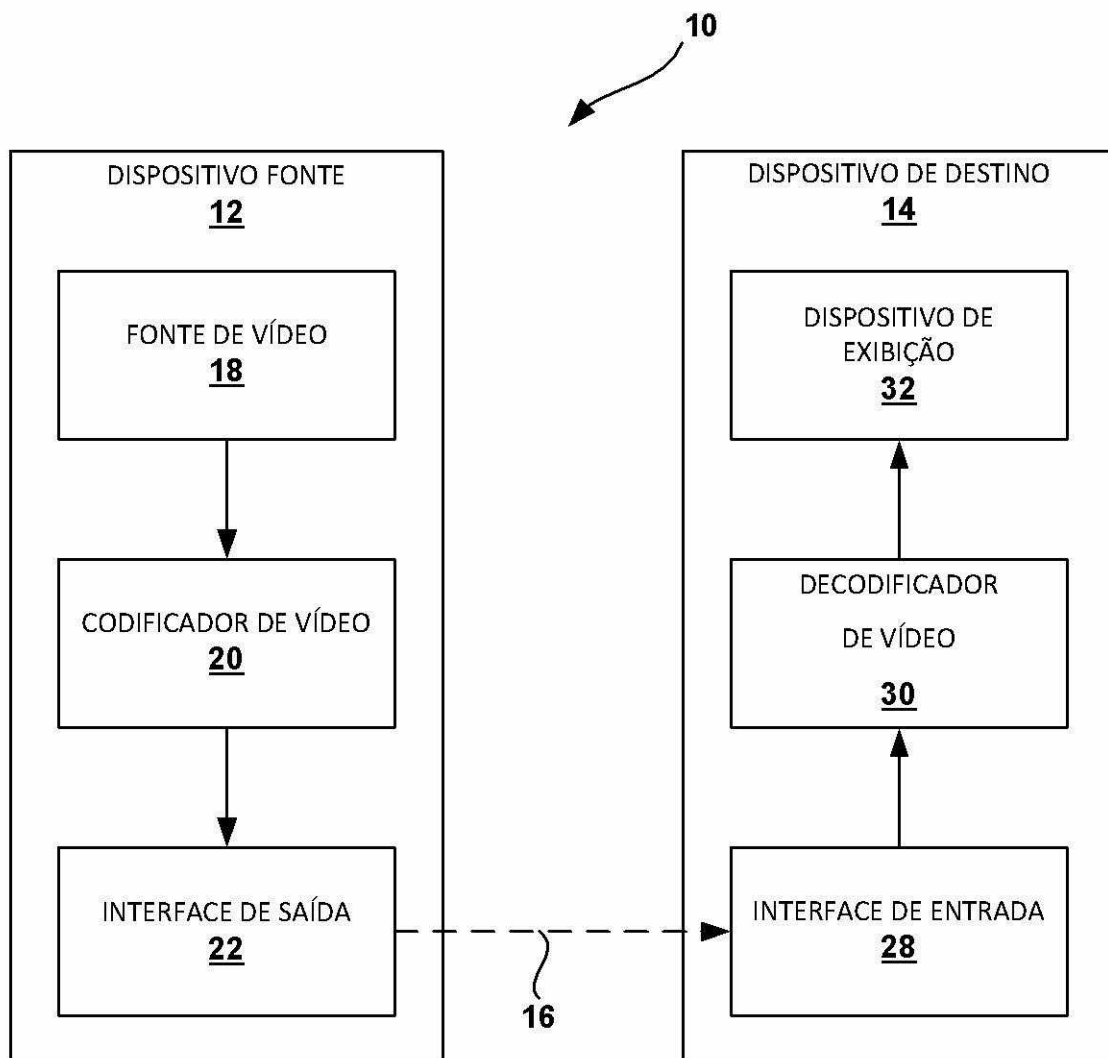


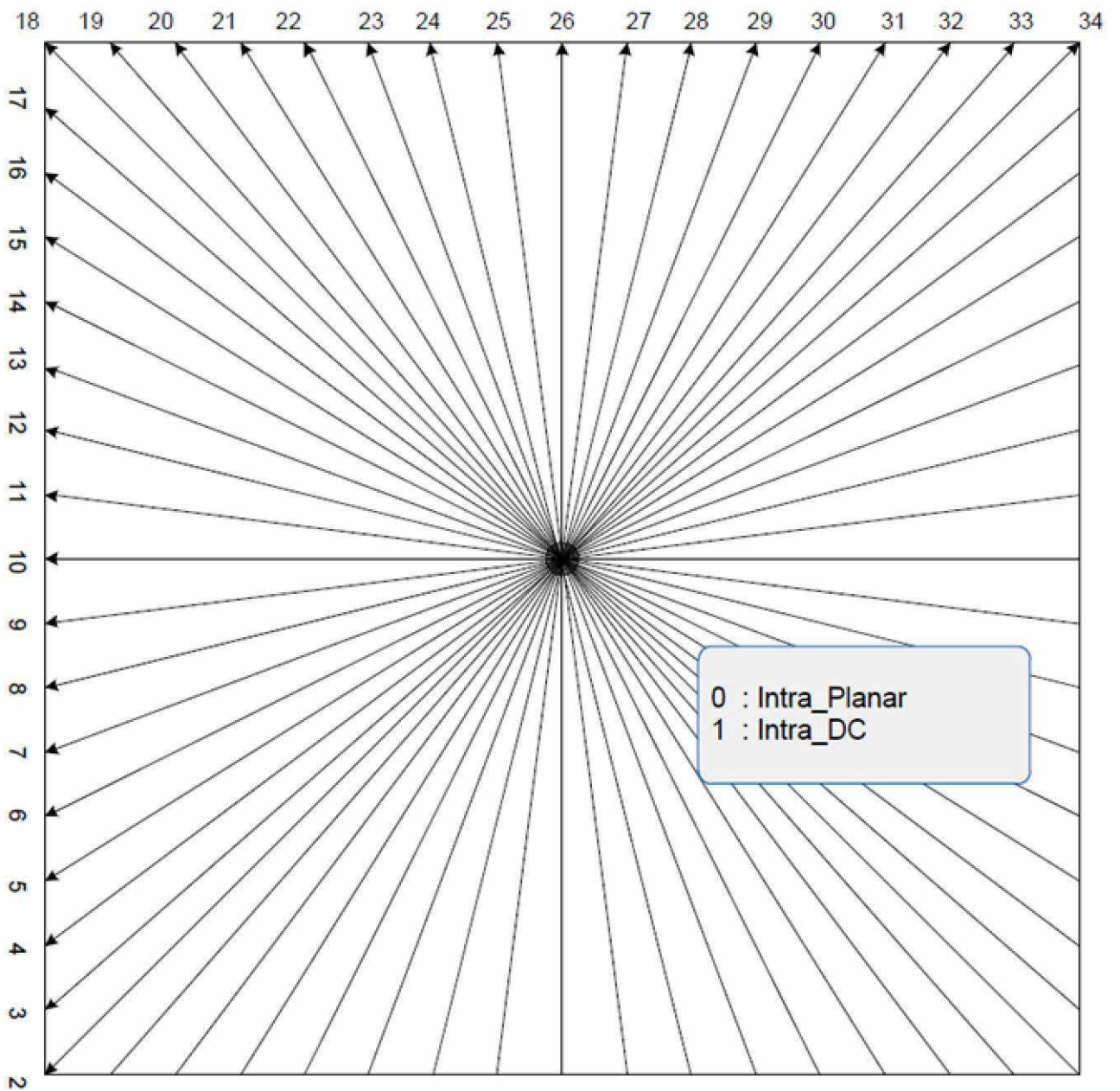
FIG. 1

$P_{-1,-1}$	$P_{-1,0}$	$P_{-1,1}$	$\dots$	$P_{-1,N-1}$	$P_{-1,N}$	$\dots$		$P_{-1,2N-1}$
$P_{0,-1}$	$P_{0,0}$	$P_{0,1}$	$\dots$	$P_{0,N-1}$				
$P_{1,-1}$	$P_{1,0}$			$\vdots$				
$\vdots$	$\vdots$							
$P_{M-1,-1}$	$P_{M-1,0}$	$\dots$		$P_{M-1,N-1}$				
$P_{M,-1}$								
$\vdots$								
$P_{2M-1,-1}$								

M = N = 4 NESTA FIGURA

$M = N = 4$  NESTA FIGURA

**FIG. 2**

**FIG. 3**

$P_{-1,-1}$	$P_{-1,0}$	$P_{-1,1}$	$\dots$	$P_{-1,N-1}$
$P_{0,-1}$	$P_{0,0}$	$P_{0,1}$	$\dots$	$P_{0,N-1}$
$P_{1,-1}$	$P_{1,0}$			$\vdots$
$\vdots$	$\vdots$			
$P_{M-1,-1}$	$P_{M-1,0}$	$\dots$		$P_{M-1,N-1}$

$M = N = 4$  NESTA FIGURA

FIG. 4

$r_{0,0}$	$r_{0,1}$	$r_{0,2}$	...	$r_{0,(N-1)}$
$r_{1,0}$	$r_{1,1}$	$r_{1,2}$	...	$r_{1,(N-1)}$
$r_{2,0}$	$r_{2,1}$	$r_{2,2}$	...	$r_{2,(N-1)}$
...	...	...	...	...
$r_{(M-1),0}$	$r_{(M-1),1}$	$r_{(M-1),2}$	...	$r_{(M-1),(N-1)}$

FIG. 5A

$r_{0,0}$	$r_{0,1}$	$r_{0,2}$	...	$r_{0,(N-1)}$
$r_{1,0}$	$r_{1,1}$	$r_{1,2}$	...	$r_{1,(N-1)}$
$r_{2,0}$	$r_{2,1}$	$r_{2,2}$	...	$r_{2,(N-1)}$
...	...	...	...	...
$r_{(M-1),0}$	$r_{(M-1),1}$	$r_{(M-1),2}$	...	$r_{(M-1),(N-1)}$

FIG. 5B

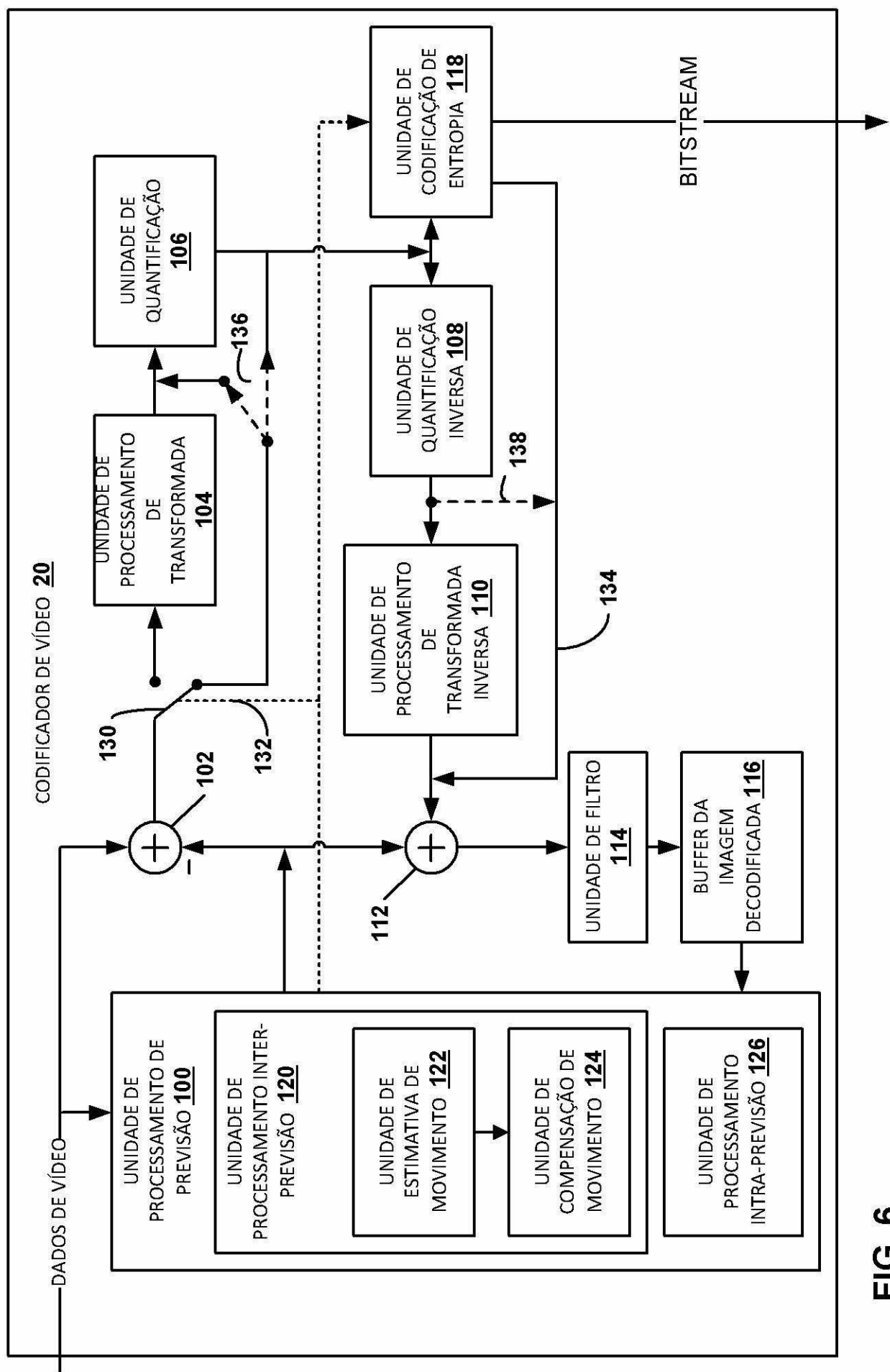


FIG. 6

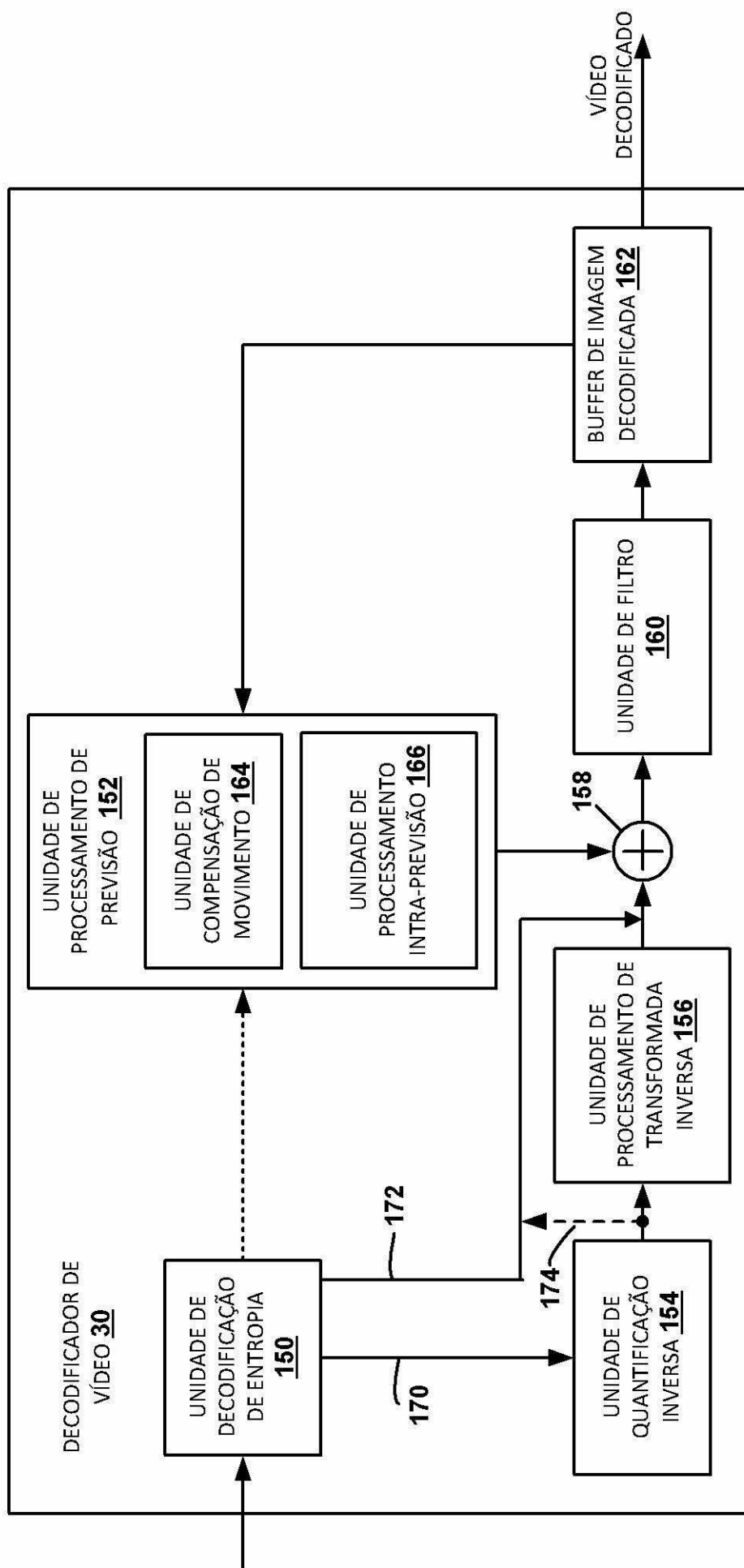


FIG. 7

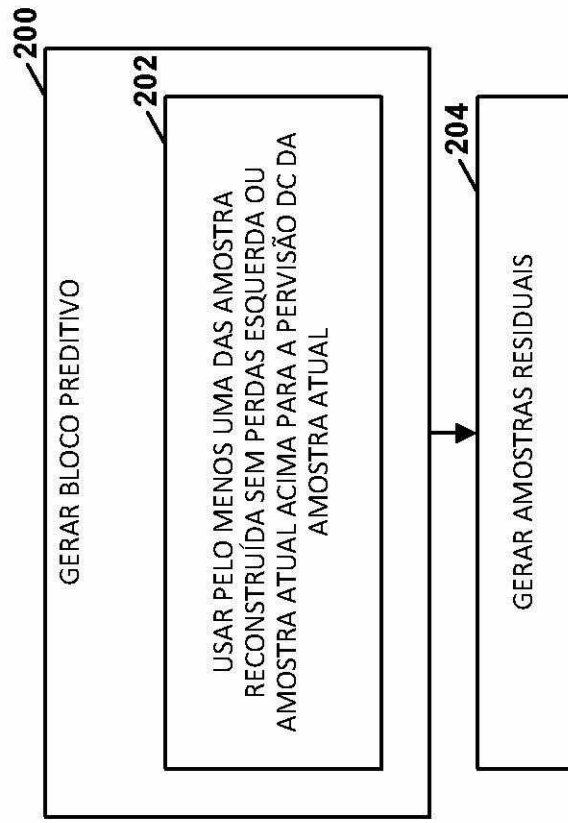


FIG. 8A

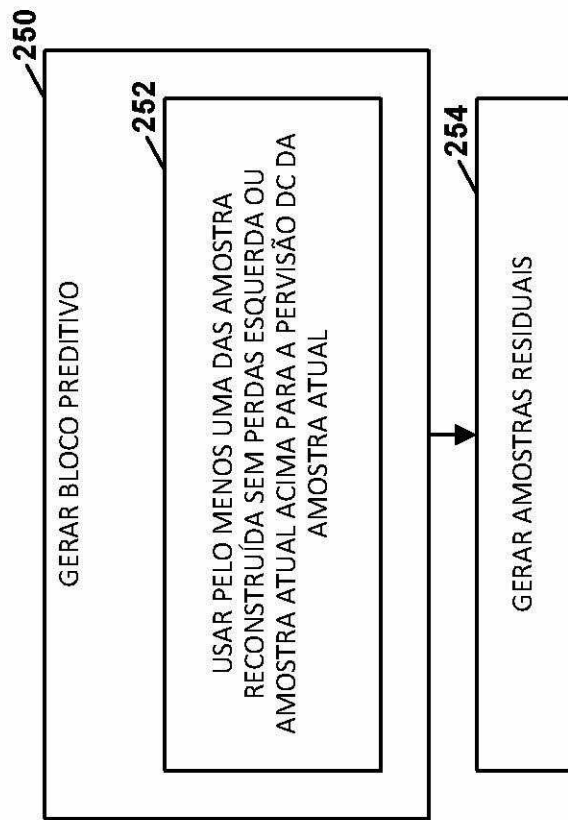


FIG. 8B



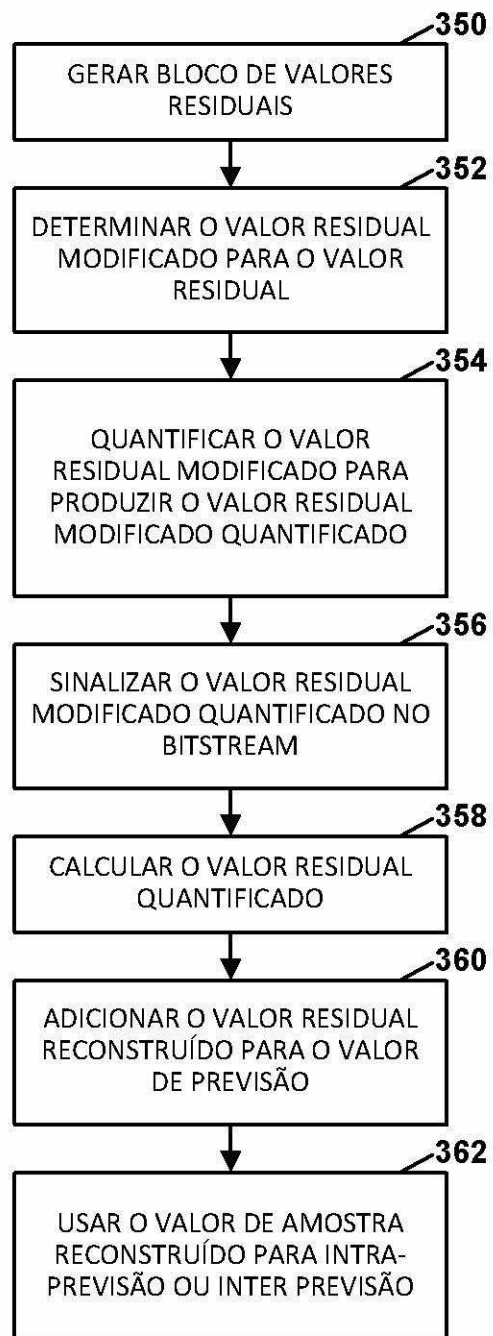


FIG. 9A

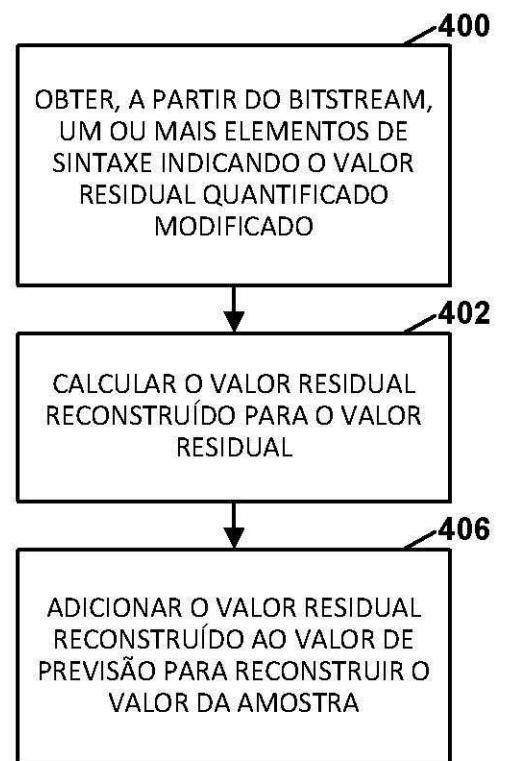


FIG. 9B

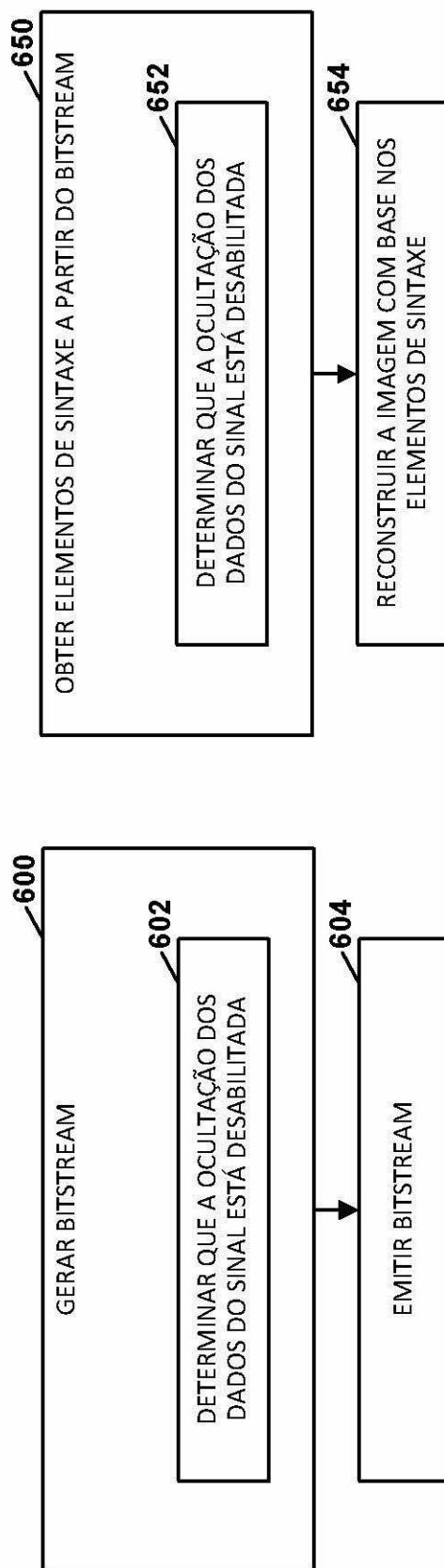


FIG. 10A

FIG. 10B