



República Federativa do Brasil

Ministério do Desenvolvimento, Indústria,
Comércio e Serviços

Instituto Nacional da Propriedade Industrial

(11) BR 112015000601-9 B1

(22) Data do Depósito: 08/07/2013

(45) Data de Concessão: 11/04/2023

(54) Título: CODIFICAÇÃO DE INFORMAÇÕES DE TEMPORIZAÇÃO PARA CODIFICAÇÃO DE VÍDEO

(51) Int.Cl.: H04N 19/31; H04N 19/46; H04N 19/573; H04N 19/70.

(52) CPC: H04N 19/31; H04N 19/46; H04N 19/573; H04N 19/70.

(30) Prioridade Unionista: 10/07/2012 US 61/670,066; 13/03/2013 US 13/802,045.

(73) Titular(es): QUALCOMM INCORPORATED.

(72) Inventor(es): YE-KUI WANG.

(86) Pedido PCT: PCT US2013049614 de 08/07/2013

(87) Publicação PCT: WO 2014/011570 de 16/01/2014

(85) Data do Início da Fase Nacional: 09/01/2015

(57) Resumo: CODIFICAÇÃO DE INFORMAÇÕES DE TEMPORIZAÇÃO PARA CODIFICAÇÃO DE VÍDEO. Em um exemplo, um aparelho para apresentar dados de vídeo inclui um processador configurado para determinar um valor de número inteiro para os dados de vídeo, determinar um valor de diferença entre o tempo de apresentação de uma primeira imagem e o tempo de apresentação de uma segunda imagem, em que o valor de diferença é igual ao valor de número inteiro multiplicado por um valor de pulso de relógio, a apresentar a primeira imagem e a segunda imagem de acordo com o valor de diferença determinado.

**"CODIFICAÇÃO DE INFORMAÇÕES DE TEMPORIZAÇÃO PARA
CODIFICAÇÃO DE VÍDEO"**

PEDIDOS CORRELATOS

[0001] Este pedido reivindica o benefício do:
pedido provisório norte-americano N°. 61/670 066,
depositado a 10 de julho de 2012, o qual é por este
incorporado em sua totalidade à guisa de referência.

CAMPO TÉCNICO

[0002] Esta revelação refere-se de maneira geral ao
processamento de dados de vídeo e, mais especificamente, a
imagens de acesso aleatório utilizadas em dados de vídeo.

ANTECEDENTES

[0003] Capacidades de vídeo digital podem ser
incorporadas a uma ampla faixa de aparelhos, que incluem
televisões digitais, sistemas de broadcast diretos
digitais, sistemas de broadcast sem fio, assistentes
digitais pessoais (PDA), computadores lap-top ou de mesa,
computadores tablet, leitoras de e-books, câmeras digitais,
aparelhos de gravação digital, tocadores de meios digitais,
aparelhos para jogos de vídeo, consoles para jogos de
vídeo, telefones celulares ou de rádio-satélite, os
chamados "telefones inteligentes", aparelhos de
teleconferência de vídeo, aparelhos de fluxo contínuo de
vídeo e semelhantes. Os aparelhos de vídeo digital
implementam técnicas de codificação de vídeo tais como as
descritas nos padrões definidos pelo ITU-T H.261, ISO/IEC
MPEG-1 Visual, ITU-T H.262, ISO/IEC MPEG-2 Visual, ITU-T
H.263, ITU-T H.264/MPEG-4 Parte 10, Codificação Avançada de
Vídeo (AVC) e o Padrão de Codificação de Vídeo de Alta
Eficácia (HEVC) atualmente em desenvolvimento e extensões
de tais padrões. Os aparelhos de vídeo podem transmitir,
receber, codificar, decodificar e/ou armazenar informações

de vídeo digital de maneira mais eficaz pela implementação de tais técnicas de codificação de vídeo.

[0004] As técnicas de codificação de vídeo incluem predição espacial (intra-imagem) e/ou predição temporal (inter-imagem) para reduzir ou remover a redundância inerente a sequências de vídeo. Para a codificação de vídeo preditiva, baseada em blocos, uma fatia de vídeo (um quadro de vídeo ou uma parte de um quadro de vídeo, por exemplo) pode ser particionada em blocos de vídeo, que podem ser também referidos como macro-blocos, blocos de árvore, unidades de árvore de codificação (CTUs), blocos de árvore de codificação (CTBs), unidades de codificação (CUs) e/ou nós de codificação. Os blocos de vídeo em uma fatia intra-codificada (I) de uma imagem são codificados utilizando-se predição espacial com relação a amostras de referência em blocos vizinhos na mesma imagem. Os blocos de vídeo em uma fatia inter-codificada (P ou B) de uma imagem pode utilizar predição espacial com relação a amostras de referência em blocos vizinhos na mesma imagem ou predição temporal com relação a amostras de referência em outras imagens de referência. As imagens podem ser referidas como quadros, e as imagens de referência podem ser referidas como quadros de referência.

[0005] A predição espacial ou temporal resulta em um bloco preditivo para um bloco a ser codificado. Os dados residuais representam diferenças de pixel entre o bloco original a ser codificado e o bloco preditivo. Os pixels podem ser referidos como elementos de imagem, pels ou amostras. Um bloco inter-codificado é codificado de acordo com um vetor de movimento que indica um bloco de amostras de referência que formam o bloco preditivo e os dados residuais que indicam a diferença entre o bloco codificado e o bloco preditivo. Um bloco intra-codificado é codificado

de acordo com um modo de intra-codificação e os dados residuais. Para compactação adicional, os dados residuais podem ser transformados do domínio de pixel em um domínio de transformada do que resultam em coeficientes de transformada residuais que podem ser então codificados. Os coeficientes de transformada quantificados, inicialmente dispostos em um arranjo bidimensional, podem ser varridos de modo a se produzir um vetor unidimensional de coeficientes de transformada, e a codificação por entropia pode ser aplicada de modo a se obter uma compactação ainda maior.

SUMÁRIO

[0006] Em geral, esta revelação descreve técnicas para processar dados de vídeo. Em particular, esta revelação descreve técnicas que podem ser utilizadas para reduzir retardos em aplicativos de vídeo, tais como aplicativos de conversação, para proporcionar aperfeiçoamentos no acesso aleatório de uma sequência de vídeo codificada e fornecer informações para conteúdo de vídeo que é de taxa de imagem fixa e suporta capacidade de escalonamento temporal.

[0007] Em um exemplo, um método para decodificar dados de vídeo inclui desencapsular uma fatia de uma imagem de ponto de acesso aleatório (RAP) de um fluxo de bits de uma unidade de camada de abstração de rede (NAL), em que a unidade NAL inclui um valor de tipo de unidade NAL que indica se a imagem RAP é de um tipo que pode ter imagens dianteiras conexas e se a imagem RAP é uma imagem de renovação de decodificador instantânea (IDR) ou uma imagem de acesso aleatório limpa (CRA), determinar se a imagem RAP pode ter imagens dianteiras conexas com base no valor de tipo de unidade NAL e decodificar os dados de vídeo do fluxo de bits que se segue à imagem RAP com base na

determinação de se a imagem RAP pode ter imagens dianteiras conexas.

[0008] Em outro exemplo, um aparelho para decodificar dados de vídeo, o aparelho compreendendo um processador configurado para desencapsular uma fatia de uma imagem de ponto de acesso aleatório (RAP) de um fluxo de bits de uma unidade de camada de abstração de rede (NAL), em que a unidade NAL inclui um valor de tipo de unidade NAL que indica se a imagem RAP é de um tipo de pode ter imagens dianteiras conexas e se a imagem RAP é uma imagem de renovação de decodificador instantânea (IDR) ou uma imagem de acesso aleatório limpa (CRA), determinar se a imagem RAP pode ter imagens dianteiras conexas com base no valor de tipo de unidade NAL e decodificar os dados de vídeo do fluxo de bits que se segue à imagem RAP com base na determinação de se a imagem RAP pode ter imagens dianteiras conexas.

[0009] Em outro exemplo, um aparelho para decodificar dados de vídeo inclui um dispositivo para desencapsular uma fatia de uma imagem de ponto de acesso aleatório (RAP) de um fluxo de bits de uma unidade de camada de abstração de rede (NAL), em que a unidade NAL inclui um valor de tipo de unidade NAL que indica se a imagem RAP é de um tipo que pode ter imagens dianteiras conexas e se a imagem RAP é uma imagem de renovação de decodificador instantânea (IDR) ou uma imagem de acesso aleatório limpa (CRA), um dispositivo para determinar se a imagem RAP pode ter imagens dianteiras conexas com base no valor de tipo de unidade NAL, e um dispositivo para decodificar os dados de vídeo do fluxo de bits que se segue à imagem RAP com base na determinação de se a imagem RAP pode ter imagens dianteiras conexas.

[0010] Em outro exemplo, um meio de armazenamento passível de leitura por computador tem armazenadas nele

instruções que, quando executadas, fazem com que um processador desencapsule uma fatia de uma imagem de ponto de acesso aleatório (RAP) de um fluxo de bits de uma unidade de camada de abstração de rede (NAL) em que a unidade NAL inclui um valor de tipo de unidade NAL que indica se a imagem RAP é de um tipo que pode ter imagens dianteiras conexas e se a imagem RAP é uma imagem de renovação de decodificador instantânea (IDR) ou imagem de acesso aleatório limpa (CRA), determine se a imagem RAP pode ter imagens dianteiras conexas com base no valor de tipo de unidade NAL, e decodifique os dados de vídeo do fluxo de bits que se segue à imagem RAP com base na determinação de se a imagem RAP pode ter imagens dianteiras conexas.

[0011] Em outro exemplo, um método para gerar um fluxo de bits que inclui dados de vídeo, o método compreendendo determinar se uma imagem de ponto de acesso aleatório (RAP) é de um tipo que pode ter imagens dianteiras conexas e se a imagem RAP compreende uma imagem de renovação de decodificador instantânea (IDR) ou uma imagem de acesso aleatório limpa (CRA), encapsular uma fatia da imagem RAP em uma unidade de camada de abstração de rede (NAL), em que a unidade NAL inclui um valor de tipo de unidade NAL que indica se a imagem RAP é de um tipo que pode ter imagens dianteiras conexas e gerar um fluxo de bits que inclui a unidade NAL.

[0012] Em outro exemplo, um aparelho para gerar um fluxo de bits que inclui dados de vídeo inclui um processador configurado para determinar se uma imagem de ponto de acesso aleatório (RAP) é de um tipo que pode ter imagens dianteiras conexas e se a imagem RAP compreende uma imagem de renovação de decodificador instantânea (IDR) ou uma imagem de acesso aleatório limpa (CRA), encapsular uma

fatia da imagem RAP em uma unidade de camada de abstração de rede (NAL), em que a unidade NAL inclui um valor de tipo de unidade NAL que indica se a imagem RAP é de um tipo que pode ter imagens dianteiras conexas e gerar um fluxo de bits que inclui a unidade NAL.

[0013] Em outro exemplo, um aparelho para gerar um fluxo de bits que inclui dados de vídeo inclui um dispositivo para determinar se uma imagem de ponto de acesso aleatório (RAP) é de tipo que pode ter imagens dianteiras conexas e se a imagem RAP compreende uma imagem de renovação de decodificador instantânea (IDR) ou uma imagem de acesso aleatório limpa (CRA), um dispositivo para encapsular uma fatia da imagem RAP em uma unidade de camada de abstração de rede (NAL), em que a unidade NAL inclui um valor de tipo de unidade NAL que indica se a imagem RAP é de um tipo que pode ter imagens dianteiras conexas, e um dispositivo para gerar um fluxo de bits que inclui a unidade NAL.

[0014] Em outro exemplo, um meio de armazenamento passível de leitura por computador tem armazenadas nele instruções que, quando executadas fazem com que um processador determine se uma imagem de ponto de acesso aleatório (RAP) é de um tipo que pode ter imagens dianteiras conexas e se a imagem RAP compreende uma imagem de renovação de decodificador instantânea (IDR) ou uma imagem de acesso aleatório limpa (CRA), encapsule uma fatia da imagem RAP em uma unidade de camada de abstração de rede (NAL), em que a unidade NAL inclui um valor de tipo de unidade NAL que indica se a imagem RAP é de um tipo que pode ter imagens dianteiras conexas, e gere um fluxo de bits que inclui a unidade NAL.

[0015] Em outro exemplo, um método para decodificar dados de vídeo inclui determinar, para uma unidade de camada de abstração de rede (NAL) de informações de

aperfeiçoamento suplementares (SEI) de um fluxo de bits, se um valor de tipo de unidade NAL para a unidade NAL SEI indica que a unidade NAL compreende uma unidade NAL SEI de prefixo que inclui uma mensagem SEI de prefixo ou uma unidade NAL SEI de sufixo que inclui uma mensagem SEI de sufixo e decodificar os dados de vídeo do fluxo de bits que se segue à unidade NAL SEI com base em se a unidade NAL SEI é a unidade NAL SEI de prefixo ou a unidade NAL SEI de sufixo e os dados da unidade NAL SEI.

[0016] Em outro exemplo, um aparelho para decodificar dados de vídeo inclui um processador configurado para determinar para uma unidade de camada de abstração de rede (NAL) de informações de aperfeiçoamento suplementares (SEI) de um fluxo de bits, se um valor de tipo de unidade NAL para unidade NAL SEI indica que a unidade NAL compreende uma unidade NAL SEI de prefixo que inclui uma mensagem SEI de prefixo ou uma unidade NAL SEI de sufixo que inclui uma mensagem SEI de sufixo e decodificar os dados de vídeo do fluxo de bits que se segue à unidade NAL SEI com base em se a unidade NAL SEI é a unidade NAL SEI de prefixo ou a unidade NAL SEI de sufixo e nos dados da unidade NAL SEI.

[0017] Em outro exemplo, um aparelho para decodificar dados de vídeo inclui um dispositivo para determinar, para uma unidade de camada de abstração de rede (NAL) de informações de aperfeiçoamento suplementares (SEI) de um fluxo de bits, se um valor de tipo de unidade NAL para a unidade NAL SEI indica que a unidade NAL compreende uma unidade NAL SEI de prefixo que inclui um mensagem SEI de prefixo ou uma unidade NAL SEI de sufixo que inclui uma mensagem SEI de sufixo, e um dispositivo para decodificar os dados de vídeo do fluxo de bits que se segue à unidade NAL SEI com base em se a unidade NAL SEI é a unidade NAL

SEI de prefixo ou a unidade NAL SEI de sufixo e nos dados da unidade NAL SEI.

[0018] Em outro exemplo, um meio de armazenamento passível de leitura por computador tem armazenadas nele instruções que, quando executadas fazem com que um processador determine para uma unidade de camada de abstração de rede (NAL) de informações de aperfeiçoamento suplementares (SEI) de um fluxo de bits, se um valor de tipo de unidade NAL para a unidade NAL SEI indica que a unidade NAL compreende uma unidade NAL SEI de prefixo que inclui uma mensagem SEI de prefixo ou uma unidade NAL SEI de sufixo que inclui uma mensagem SEI de sufixo, e decodifique os dados de vídeo do fluxo de bits que se segue à unidade NAL SEI com base em se a unidade NAL é a unidade NAL SEI de prefixo ou a unidade NAL SEI de sufixo e nos dados da unidade NAL SEI.

[0019] Em outro exemplo, um método para gerar um fluxo de bits que inclui dados de vídeo inclui determinar se uma mensagem de informações de aperfeiçoamento suplementares (SEI) é uma mensagem SEI de prefixo ou uma mensagem SEI de sufixo, em que a mensagem SEI inclui dados relacionados com dados de vídeo codificados, encapsular a mensagem SEI em uma unidade NAL SEI, em que a unidade NAL SEI inclui um valor de tipo de unidade NAL que indica se a unidade NAL SEI é uma unidade NAL SEI de prefixo ou uma unidade NAL SEI de sufixo e se a mensagem SEI é uma mensagem SEI de prefixo ou uma mensagem SEI de sufixo, e gerar um fluxo de bits que inclui pelo menos a unidade NAL SEI.

[0020] Em outro exemplo, um aparelho para gerar um fluxo de bits que inclui vídeo inclui um processador configurado para determinar se uma mensagem de informações de aperfeiçoamento suplementares (SEI) é uma mensagem SEI de prefixo ou uma mensagem SEI de sufixo, em que a mensagem

SEI inclui dados relacionados com dados de vídeo codificados, encapsular a mensagem SEI em uma unidade NAL SEI, em que a unidade NAL SEI inclui um valor de tipo de unidade NAL que indica se a unidade NAL SEI é uma unidade NAL SEI de prefixo ou uma unidade NAL SEI de sufixo e em se a mensagem SEI é uma mensagem SEI de prefixo ou uma mensagem SEI de sufixo, e gerar um fluxo de bits que inclui pelo menos a unidade NAL SEI.

[0021] Em outro exemplo, um aparelho para gerar um fluxo de bits que inclui dados de vídeo inclui um dispositivo para determinar se uma mensagem de informações de aperfeiçoamento suplementares (SEI) é uma mensagem de SEI de prefixo ou uma mensagem SEI de sufixo, em que a mensagem SEI inclui dados relacionados com dados de vídeo codificados, um dispositivo para encapsular a mensagem SEI em uma unidade NAL SEI, em que a unidade NAL SEI inclui um valor de tipo de unidade NAL que indica se a unidade NAL SEI é uma unidade NAL SEI de prefixo ou uma unidade NAL SEI de sufixo e se a mensagem SEI é uma mensagem SEI de prefixo ou uma mensagem SEI de sufixo, e um dispositivo para gerar um fluxo de bits que inclui pelo menos a unidade NAL SEI.

[0022] Em outro exemplo, um meio de armazenamento passível de leitura por computador que tem armazenadas nele instruções que, quando executadas fazem com que um processador determine se uma mensagem de informações de aperfeiçoamento suplementares (SEI) é uma mensagem SEI de prefixo ou uma mensagem SEI de sufixo, em que a mensagem SEI inclui dados relacionados com dados de vídeo codificados, encapsule a mensagem SEI em uma unidade NAL SEI, em que a unidade NAL SEI inclui um valor de tipo de unidade NAL que indica se a unidade NAL SEI é uma unidade NAL SEI de prefixo ou unidade NAL SEI de sufixo e se a mensagem SEI é uma mensagem SEI de prefixo ou uma mensagem

SEI de sufixo, e gere um fluxo de bits que inclui pelo menos a unidade NAL SEI.

[0023] Em outro exemplo, um método para apresentar dados de vídeo inclui determinar um valor de número inteiro para os dados de vídeo, determinar um valor de diferença entre o tempo de apresentação de uma primeira imagem e o tempo de apresentação de uma segunda imagem, em que o valor de diferença é igual ao valor de número inteiro multiplicado por um valor de pulso de relógio e apresentar a primeira imagem e a segunda imagem de acordo com o valor de diferença determinado.

[0024] Em outro exemplo, um aparelho para processar dados de vídeo inclui um processador configurado para determinar um valor de número inteiro para os dados de vídeo, determinar um valor de diferença entre o tempo de apresentação de uma primeira imagem e o tempo de apresentação de uma segunda imagem, em que o valor de diferença é igual ao valor de número inteiro multiplicado por um valor de pulso de relógio, e apresentar a primeira imagem e a segunda imagem de acordo com o valor de diferença determinado.

[0025] Em outro exemplo, um aparelho para apresentar dados de vídeo inclui um dispositivo para determinar um valor de número inteiro para os dados de vídeo, um dispositivo para determinar um valor de diferença entre o tempo de apresentação de uma primeira imagem e o tempo de apresentação de uma segunda imagem, em que o valor de diferença é igual ao valor de número inteiro multiplicado por um valor de pulso de relógio e um dispositivo para apresentar a primeira imagem e a segunda imagem de acordo com o valor de diferença determinado.

[0026] Em outro exemplo, um meio de armazenamento passível de leitura por computador tem armazenadas nele

instruções que, quando executadas, fazem com que um processador determine um valor de número inteiro para os dados de vídeo, determine um valor de diferença entre o tempo de apresentação de uma primeira imagem e o tempo de apresentação de uma segunda imagem, em que o valor de diferença é igual ao valor de número inteiro multiplicado por um valor de pulso de relógio, e apresente a primeira imagem e a segunda imagem de acordo com o valor de diferença determinado.

[0027] Em outro exemplo, um método para gerar um fluxo de bits que inclui dados de vídeo inclui gerar dados que indicam se a diferença entre o tempo de apresentação de uma primeira imagem e o tempo de apresentação de uma segunda imagem é um múltiplo de número inteiro de um valor de pulso de relógio, e, quando os dados indicam que a diferença é um múltiplo de número inteiro do valor de pulso de relógio, gerar dados que representam o múltiplo de número inteiro.

[0028] Em outro exemplo, um aparelho para gerar um fluxo de bits que inclui dados de vídeo inclui um processador configurado para gerar dados que indicam se a diferença entre o tempo de apresentação de uma primeira imagem e o tempo de apresentação de uma segunda imagem é um múltiplo de número inteiro de um valor de pulso de relógio e, quando os dados indicam que a diferença é um múltiplo de número inteiro do valor de pulso de relógio, gerar dados que representam o múltiplo de número inteiro.

[0029] Em outro exemplo, um aparelho para gerar um fluxo de bits que inclui dados de vídeo inclui um dispositivo para gerar dados que indicam se a diferença entre o tempo de apresentação de uma primeira imagem e o tempo de apresentação de uma segunda imagem é um múltiplo de número inteiro de um valor de pulso de relógio, e um dispositivo para gerar, quando os dados indicam que a diferença é um

múltiplo de número inteiro do valor de pulso de relógio, dados que representam o múltiplo de número inteiro.

[0030] Em outro exemplo, um meio de armazenamento passível de leitura por computador tem armazenadas nele instruções que, quando executadas, fazem com que, um processador gere dados que indicam se a diferença entre o tempo de apresentação de uma primeira imagem e o tempo de apresentação de uma segunda imagem é um múltiplo de número inteiro de um valor de pulso de relógio e, quando os dados indicam que a diferença é um múltiplo de número inteiro do valor de pulso de relógio gere dados que representam o múltiplo de número inteiro.

[0031] Os detalhes de um ou mais exemplos são apresentados nos desenhos anexos e na descrição que se segue. Outros recursos, objetos e vantagens serão evidentes a partir da descrição e dos desenhos, e a partir das reivindicações.

DESCRIÇÃO RESUMIDA DOS DESENHOS

[0032] A Figura 1 é um diagrama conceptual que mostra uma sequência de vídeo codificada de acordo com técnicas de codificação preditiva de vídeo.

[0033] A Figura 2 é um diagrama conceptual que mostra um exemplo de sequência de vídeo codificada.

[0034] A Figura 3 é um diagrama de blocos que mostra um sistema de codificação e decodificação de vídeo exemplar que pode utilizar as técnicas descritas nesta revelação.

[0035] A Figura 4 é um diagrama de blocos que mostra uma unidade de encapsulamento exemplar que pode implementar as técnicas descritas nesta revelação.

[0036] A Figura 5 é um fluxograma que mostra um exemplo de geração de unidade NAL VCL de acordo com as técnicas desta revelação.

[0037] A Figura 6 é um fluxograma que mostra um exemplo de geração de unidades NAL não VCL de acordo com as técnicas desta revelação.

[0058] A Figura 7 é um fluxograma que mostra um exemplo de sinalização de um valor delta de tempo de apresentação.

[0039] A Figura 8 é um diagrama de blocos que mostra um codificador de vídeo exemplar que pode implementar as técnicas descritas nesta revelação.

[0040] A Figura 9 é um fluxograma que mostra um exemplo de determinação de um valor delta de tempo de apresentação.

[0041] A Figura 10 é um diagrama de blocos que mostra um decodificador de vídeo exemplar que pode implementar as técnicas descritas nesta revelação.

DESCRIÇÃO DETALHADA

[0042] Esta revelação descreve desenhos de codificação de vídeo aperfeiçoados. Em particular, esta revelação descreve técnicas que podem ser utilizadas para reduzir retardo em aplicativos de vídeo, tais como aplicativos de conversação, e proporcionar aperfeiçoamentos no acesso aleatório de uma sequência de vídeo codificada.

[0043] Aparelhos de vídeo digital implementam técnicas de compactação de vídeo para codificar e decodificar informações de vídeo digital de maneira mais eficaz. As técnicas de compactação de vídeo podem ser definidas de acordo com um padrão de codificação de vídeo, tal como AVC ou AHEVC. O padrão ITU-T H.264/MPEG-4 AVC foi formulado pelo Grupo de Especialistas em Codificação de Vídeo ITU-T (VCEG) juntamente com o Grupo de Especialistas de Imagens em Movimento ISO/TEC-MPEG como o produto de uma parceria coletiva conhecida como Equipe de Vídeo Conjunta (JVT). O padrão H.264 é descrito na recomendação ITU-T H.264, Codificação Avançada de Vídeo para Serviços Áudio Visuais Genéricos, pelo Grupo de Estudos ITU-T, e datada de março

de 2005, a qual pode ser aqui referida como o padrão H.264 ou especificação H.264, ou o padrão ou especificação H.264/AVC. A Equipe de Vídeo Conjunta (JVT) continua a trabalhar nas extensões do H.264/MPEG-4 AVC.

[0044] Um Rascunho operacional (WD) recente da HEVC referido como "Rascunho Operacional HEVC 7" ou "WD7", é descrito no documento JCTVC-I1003_d5, Bross *et alii*, "WD7: Rascunho Operacional 7 da Codificação de Vídeo de Alta Eficácia (HEVC)", Equipe de Colaboração Conjunta em Codificação de Vídeo (JCT-VC) do ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 9°. Encontro, Genebra, Suíça, 27 de maio de 2012 a 7 de maio de 2012. Além disto, outro rascunho operacional recente, o Rascunho Operacional 9, é descrito no documento HCTVCK1003_d7, Bross *et alii*, "Rascunho de Especificação de Texto de Codificação de Vídeo de Alta Eficácia 9", Equipe de Colaboração Conjunta em Codificação de Vídeo (JCT-VC) do ITU-T SG16 WP3 e ISO IEC JT CI/SC29/WG11, 11°. Encontro, Xangai, CN, outubro de 2012. O padrão HEVC vindouro pode ser também referido como ISO/IEC 23008-HEVC que é destinado a ser um número de padrão para a versão lançada da HEVC. Sob alguns aspectos, as técnicas descritas nesta revelação podem ser aplicadas a aparelhos que se conformam geralmente ao padrão H.264 e/ou ao padrão HEVC vindouro. Embora as técnicas desta revelação sejam descritas com relação ao padrão H.264 e ao padrão HEVC vindouro, as técnicas desta revelação são geralmente aplicáveis a qualquer padrão de codificação de vídeo.

[0045] Uma sequência de vídeo inclui tipicamente uma série de quadros de vídeo também referidos como imagens. Exemplos de aplicativos de vídeo nos quais uma sequência de vídeo é codificada e/ou decodificada inclui repetição local, fluxo contínuo, aplicativos de broadcast, multicast e de conversação. Os aplicativos de conversação incluem

telefonia com vídeo e conferência de vídeo e são também referidos como aplicativos de baixo retardo. Os aplicativos de conversação exigem um retardo de uma extremidade à outra relativamente baixo dos sistemas inteiros, isto é, o retardo entre o momento em que um quadro de vídeo é captado em um primeiro aparelho de vídeo digital e o momento em que o quadro de vídeo é exibido em um segundo aparelho de vídeo digital. Um retardo de uma extremidade à outra, tipicamente aceitável para aplicativos de conversação deve ser inferior a 400 mseg, e um retardo de extremidade à outra de cerca de 150 mseg é considerado muito bom.

[0046] Cada etapa associada ao processamento de uma sequência de vídeo pode contribuir para o retardo de uma extremidade à outra total. Exemplos de retardos associados ao processamento de uma sequência de vídeo incluem retardo de captação, retardo de pré-processamento, retardo de codificação, retardo de transmissão, retardo de armazenamento de recepção (para anulação de flutuação de fase), retardo de decodificação, retardo de saída de imagens decodificadas, retardo de pós-processamento e retardo de exibição. O retardo associado à codificação de uma sequência de vídeo de acordo com um padrão de codificação de vídeo específico pode ser referido como retardo de CODEC e pode incluir um retardo de codificação, um retardo de decodificação e um retardo de saída de imagens decodificadas. O retardo de CODEC deve ser reduzido ao mínimo em aplicativos de conversação. Em particular, a estrutura de codificação de uma sequência de vídeo deve assegurar que a ordem de saída das imagens na sequência de vídeo seja idêntica à ordem de decodificação de imagens na sequência de vídeo, de modo que o retardo de saída de imagens decodificadas seja igual a zero. A estrutura de codificação de uma sequência de vídeo refere-se em parte à

alocação de tipos de imagem utilizados para codificar uma sequência de vídeo.

[0047] Um grupo de imagens (GOP) compreende geralmente uma sequência de uma ou mais imagens dispostas de acordo com uma ordem de exibição. De acordo com a HEVC, um codificador de vídeo pode dividir um quadro ou imagem de vídeo em uma série de blocos de vídeo de tamanho igual. Um bloco de vídeo pode ter um componente de luminância (denotado como Y) e dois componentes croma (denotados como U e V ou Cb e Cr). Estes blocos de vídeo podem ser também referidos como maiores unidades de codificação (LCUs), blocos de árvore ou unidades de bloco de árvore de codificação (CTUs). As LCUs da HEVC podem ser amplamente análogas aos macro-blocos de padrões anteriores, tais como o H.264/AVC. Entretanto, uma LCU não está necessariamente limitada a um tamanho específico. De acordo com a HEVC, os dados de sintaxe dentro de um fluxo de bits podem definir uma LCU de acordo com um número de amostras luma horizontais e/ou verticais. Por exemplo, uma LCU pode ser definida como incluindo 64x64 ou 32x32 amostras luma. Além disto, uma LCU pode ser particionada em várias unidades de codificação (CUs) de acordo com uma esquema de particionamento de transformação quad-tree. Em geral, o particionamento de transformação quad-tree refere-se à divisão recursiva de CUs em quatro sub-CUs. Os dados de sintaxe associados a um fluxo de bits codificado pode definir um número máximo de vezes em que uma LCU pode ser dividida, referido como profundidade de CU máxima, e podem definir também o tamanho mínimo de uma CU. Por conseguinte, um fluxo de bits pode definir também a menor unidade de codificação (SCU). Por exemplo, uma SCU pode ser definida como incluindo 8x8 amostras luma.

[0048] Além disso, de acordo com a HEVC, um codificador de vídeo pode particionar uma imagem em uma série de fatias, em que cada uma das fatias inclui um número inteiro de LCUs. As fatias podem ser fatias I, fatias P ou fatias B, onde I, P e B definem como outros blocos de vídeo são utilizados para prever as CUs. Uma fatia I é prevista utilizando-se um modo intra-predição (a partir de blocos de vídeo dentro do mesmo quadro, por exemplo). A intra-codificação recorre à predição espacial para reduzir ou remover a redundância espacial em vídeo dentro de um dado quadro ou imagem de vídeo. Uma fatia P é prevista utilizando-se um modo de intra-predição unidirecional (a partir de blocos de vídeo em um quadro anterior, por exemplo). Uma fatia B é prevista utilizando-se um modo de inter-predição bidirecional (a partir de blocos de vídeo dentro de um quadro anterior e um quadro subsequente, por exemplo). A inter-codificação recorre à predição temporal para reduzir ou remover a redundância temporal em vídeo dentro de quadros ou imagens adjacentes de uma sequência de vídeo.

[0049] A Figura 1 é um diagrama conceptual que mostra uma sequência de vídeo codificada de acordo com técnicas de codificação de vídeo preditiva. Conforme mostrado na Figura 1, a sequência de vídeo 100 inclui imagens Pic₁-Pic₁₀. No diagrama conceptual da Figura 1 as imagens Pic₁-Pic₁₀ são dispostas e numeradas sequencialmente de acordo com a ordem na qual elas serão exibidas. Conforme descrito mais detalhadamente em seguida, a ordem de exibição não corresponde necessariamente à ordem de decodificação. Conforme mostrado na Figura 1, a sequência de vídeo 100 inclui um GOP₁ e um GOP₂, onde as imagens Pic₁-Pic₅ são incluídas no GOP₁ e as imagens Pic₆-Pic₁₀ são incluídas no

GOP₂. A Figura 1 mostra onde a Pic₅ é particionada na fatia 1 e na fatia 2, onde cada uma da fatia 1 e da fatia 2 inclui LCUs consecutivas de acordo com uma varredura de rastreamento da esquerda para a direita e do topo para a base. Embora não mostrado, as outras imagens mostradas na Figura 1 podem ser particionadas em uma ou mais fatias de maneira semelhante. A Figura 1 mostra também o conceito de fatias I, fatias P ou fatias B com relação ao GOP₂. As setas associadas a cada uma das Pic₆-Pic₁₀ no GOP₂ indicam se uma imagem inclui fatias I, fatias P ou fatias B com base em uma imagem referida indicada pelas setas. Na Figura 1 as imagens Pic₆ e Pic₉ representam imagens que incluem fatias I (isto é, as referências são com as imagens propriamente ditas), as imagens Pic₇ e Pic₁₀ representam imagens que incluem fatias P (isto é, cada uma se refere a uma imagem anterior) e Pic₈ representa uma imagem que inclui fatias B (isto é, refere uma imagem anterior e uma imagem subsequente).

[0050] Na HEVC, cada uma de uma sequência de vídeo, um GOP, uma imagem, uma fatia e uma CU, pode ser associado a um dado de sintaxe que descreve propriedades de codificação de vídeo. Por exemplo, uma fatia inclui um cabeçalho que inclui um elemento de sintaxe que indica se a fatia é uma fatia I, uma fatia P ou uma fatia B. Além disto, a HEVC inclui o conceito de conjunto de parâmetros. Um conjunto de parâmetros é uma estrutura de sintaxe que inclui elementos de sintaxe que permitem que um decodificador de vídeo reconstrua uma sequência de vídeo. A HEVC utiliza um mecanismo de conjunto de parâmetros hierárquico no qual elementos de sintaxe são incluídos em um tipo de conjunto de parâmetros com base na frequência na qual se espera que os elementos de sintaxe se alterem. O mecanismo de conjunto

de parâmetros na HEVC desacopla a transmissão de informações que se alteram de maneira não freqüente da transmissão de dados de bloco codificados. Além disto, em alguns aplicativos os conjuntos de parâmetro podem ser transmitidos "fora da banda", isto é, não transportados juntamente com as unidades que contêm dados de vídeo codificados. A transmissão fora da banda é tipicamente confiável.

[0051] No WD7 HEVC, um conjunto de parâmetros específico é identificado utilizando-se um ID de conjunto de parâmetros. No WD7 HEVC, um ID de conjunto de parâmetros é um elemento de sintaxe codificado por Golomb Exp. de número inteiro não sinalizado com o bit esquerdo primeiro. O WD7 HEVC define os conjuntos de parâmetros seguintes:

Conjunto de Parâmetros de Vídeo (VPS): um VPS é uma estrutura de sintaxe que contém elementos de sintaxe que se aplicam a zero ou mais sequências de vídeo codificadas inteiras. Ou seja, um VPS inclui elementos de sintaxe que se espera permaneçam inalterados para uma sequência de quadros (como, por exemplo, ordem de imagens, número de quadros de referência e tamanho de imagem). Um VPS é identificado utilizando-se um ID de VPS. Um Conjunto de Parâmetros de Sequência inclui um ID de VPS.

Conjunto de Parâmetros de Sequência (SPS) - um SPS é uma estrutura de sintaxe que inclui elementos de sintaxe que se aplicam a zero ou mais sequências de vídeo codificadas inteiras. Ou seja, um SPS inclui elementos de sintaxe que se espera permaneçam inalterados para uma sequência de quadros (como, por exemplo, ordem de imagens, número de quadros de referência e tamanho de imagem). Um SPS é identificado utilizando-se um ID de SPS. Um Conjunto de Parâmetros de Imagem inclui um ID de SPS.

Conjunto de Parâmetros de Imagem (PPS) - um PPS é uma estrutura de sintaxe que inclui elementos de sintaxe que se aplicam a uma ou mais imagens. Ou seja, um PPS inclui elementos de sintaxe que podem alterar-se de imagem para imagem dentro de uma sequência (como, por exemplo, modo de codificação por entropia, parâmetros de quantificação e profundidade de bits). Um Conjunto de Parâmetros PPS é identificado utilizando-se um ID de PPS. Um cabeçalho de fatia inclui um ID de PPS.

Conjunto de Parâmetros Adaptativos (APS) - um APS é uma estrutura de sintaxe que inclui elementos de sintaxe que se aplicam a uma ou mais imagens. Um APS inclui elementos de sintaxe que se espera que se alterem dentro de imagens de uma sequência (como, por exemplo, tamanho de bloco e filtragem de desbloqueio) um Conjunto APS é identificado utilizando-se um ID de APS. Um cabeçalho de fatia pode incluir um ID de APS.

[0052] De acordo com os tipos de conjunto de parâmetros definidos no WD7 HEVC, cada SPS refere-se a um ID de VPS, cada PPS refere-se a um ID de SPS, e cada cabeçalho de fatia refere-se a um ID de PPS e possivelmente a um ID de APS. Deve-se observar que, em alguns casos, a relação de referência linear da inclusão de um ID de VPS em um SPS e de um ID de SPS em um PPS pode ser ineficaz. Por exemplo, embora um VPS seja suportado no WD7 HEVC a maioria dos parâmetros de informação ao nível de sequência ainda estão somente presentes no SPS. Além do conceito de conjunto de parâmetros, a HEVC inclui o conceito de sequências de vídeo codificadas e unidades de acesso. De acordo com o WD7 HEVC, uma sequência de vídeo codificada e uma unidade de acesso são definidas de maneira seguinte:

Sequência de Vídeo Codificada: uma sequência de unidades de acesso que consiste, em ordem de decodificação, em uma

unidade de acesso CRA que é a primeira unidade de acesso no fluxo de bits, uma unidade de acesso IDR ou uma unidade de acesso BLA, seguida de zero ou mais unidades de acesso não IDR e não BLA que incluem todas as unidades de acesso subseqüentes até, mas não incluindo, qualquer unidade de acesso IDR ou BLA subseqüente [unidades de acesso CRA, IDR e BLA são descritas em detalhe em seguida].

Unidade de Acesso: um conjunto de unidades NAL que são consecutivas em ordem de decodificação e contêm uma imagem codificada. Além das unidades NAL de fatia codificadas da imagem codificada, a unidade de acesso pode conter também outras unidades NAL que não contêm fatias da imagem codificada. A decodificação da unidade de acesso resulta sempre em uma imagem decodificada.

[0053] Uma unidade NAL refere-se a uma Unidade de Camada de Abstração de Rede. Assim, de acordo com a HEVC, um fluxo de bits de dados de vídeo codificados inclui uma sequência de unidades NAL. Uma unidade de acesso é um conjunto de unidades NAL que são dispostas consecutivamente em ordem de decodificação e contêm exatamente uma imagem codificada e uma sequência de vídeo codificada inclui uma sequência de unidades de acesso dispostas em ordem de decodificação. A Figura 2 é um diagrama conceptual que mostra um exemplo de sequência de vídeo codificada. A Figura 2 representa um exemplo de sequência de vídeo codificada 200 que pode corresponder ao GOP₂ mostrado na Figura 1. Conforme mostrado na Figura 2, a sequência de vídeo codificada 200 inclui uma unidade de acesso que corresponde a cada uma das Pic₆-Pic₁₀. As unidades de acesso da sequência de vídeo codificada 200 são dispostas sequencialmente de acordo com a ordem de decodificação. Deve-se observar que a unidade de acesso que corresponde à Pic₉ é localizada antes da unidade

de acesso que corresponde à Pic₈. Assim, a ordem de decodificação não corresponde à ordem de exibição mostrada na Figura 1. Neste exemplo, isto é devido ao fato de a Pic₈ referir a Pic₉. Assim, a Pic₉ deve ser decodificada antes que a Pic₈ possa ser decodificada. A Figura 2 mostra onde a unidade de acesso que corresponde à Pic₉ inclui unidades NAL: unidade NAL delimitadora AU 202, unidade NAL PPS 204, unidade NAL de fatia₁ e unidade NAL de fatia₂ 208. Cada unidade NAL pode incluir um cabeçalho que identifica o tipo de unidade NAL.

[0054] A HEVC define duas classes de tipo de unidade NAL. Unidades NAL de fatias codificadas (VCL) e unidades NAL não VCL. Uma unidade NAL de fatia codificada contém uma fatia de dados de vídeo. No exemplo mostrado na Figura 2, a unidade NAL de fatia₁ 206 e a unidade NAL de fatia₂ 208 contêm, cada uma, uma fatia de dados de vídeo que são exemplos de unidades NAL VCL. No exemplo da Figura 2 cada uma da unidade NAL de fatia₁ 206 e unidade NAL de fatia₂ 208 podem ser fatias I. uma unidade não VCL inclui informações que não uma fatia de dados de vídeo. Por exemplo, uma não VCL pode conter dados delimitadores ou um conjunto de parâmetros. No exemplo mostrado na Figura 2, a unidade NAL delimitadora AU 202 inclui informações para delimitar a unidade de acesso que corresponde à Pic₉ da unidade de acesso que corresponde à Pic₇. Além disto, a unidade NAL PPS 204 inclui um conjunto de parâmetros de imagem. Assim, a unidade NAL delimitadora AU 202 e a unidade NAL PPS 204 são exemplos de unidades NAL não VCL.

[0055] Outro exemplo de unidade NAL não VCL na HEVC é a unidade NAL de informações de aperfeiçoamento suplementares (SEI). O mecanismo SEI suportado tanto na AVC quanto na

HEVC permite que os codificadores incluam metadados no fluxo de bits que não são necessários para a decodificação correta dos valores de amostra das imagens de saída, mas podem ser utilizados para diversas outras finalidades, tais como temporização de saída de imagens, exibição, assim como detecção e ocultação de perdas. Por exemplo, as unidades NAL SEI podem incluir mensagens de temporização de imagens que são utilizadas por um decodificador de vídeo quando decodifica um fluxo de bits. As mensagens de temporização de imagens podem incluir informações que indicam quando um decodificador de vídeo deve começar a decodificar uma unidade NAL VCL. Os codificadores podem incluir qualquer número de unidades NAL SEI em uma unidade de acesso, e cada unidade NAL SEI pode conter uma ou mais mensagens SEI. O padrão HEVC de rascunho inclui a sintaxe e a semântica para várias mensagens SEI, mas o processamento das mensagens SEI não é especificado, uma vez que elas não afetam o processo de decodificação normativo. Uma razão para ter as mensagens SEI no padrão HEVC de rascunho é a de permitir que dados suplementares sejam interpretados de maneira idêntica em sistemas diferentes que utilizam a HEVC. Especificações e sistemas que utilizam a HEVC podem exigir codificadores para gerar determinadas mensagens SEI ou podem definir processamento específico de tipos específicos de mensagens SEI recebidas. A Tabela 1 enumera as mensagens SEI especificadas na HEVC e descreve resumidamente suas finalidades.

Mensagem Sei	Finalidade
Período de Armazenamento.	Retardos iniciais para funcionamento de decodificador de referência hipotético (HRD) .
Temporização de imagens.	Tempo de saída de imagens e tempo de remoção de imagens/sub-imagens para funcionamento de HRD, assim como informações relacionadas com estrutura de

	imagens.
Retângulo de PAN-varredura.	Exibição a uma razão de aspecto de imagem (PAR) diferente da PAR das imagens transmitidas.
Carga útil de enchimento.	Ajuste da taxa de bits para satisfazer restrições específicas.
Dados de usuário registrados dados de usuário não registrados.	Mensagens SEI a serem especificadas por entidades externas.
Ponto de Recuperação.	Informações adicionais para acesso aleatório limpo. Renovação de decodificação gradual
Informações sobre cenas.	Informações sobre alterações e transições de cenas.
Instantâneo de quadro inteiro.	Indicação para rotular as imagem decodificada conexa como um instantâneo de imagem parada de um conteúdo de vídeo.
Segmento de refinamento progressivo.	Indica que determinadas imagens consecutivas representam um refinamento progressivo da qualidade de uma imagem e não de uma cena em movimento.
Característica de grão de película.	Permite que decodificadores sintetizem o grão de película.
Preferência de exibição de filtro de desbloqueio.	Recomenda se ou não as imagens exibidas devem passar por desbloqueio pelo processo de filtragem por desbloqueio na malha.
Palpite pós-filtro.	Apresenta coeficientes pós-filtro sugeridos com informações de correlação para desenho pós-filtro.
Informações de mapeamento de tons.	Remapeamento em outro espaço colorido que o utilizado ou suposto na codificação.
Disposição de empacotamento de quadros.	Acondicionamento de vídeo estereoscópico em um fluxo de bits HEVC.
Orientação de exibição.	Especifica movimento rápido e/ou rotação que deve ser aplicada às imagens de saída quando elas são exibidas.
Estrutura de descrição de imagens	Descreve a estrutura temporal e de inter-predição do fluxo de bits.
Hash de imagens decodificadas	Soma de verificação da imagem decodificada, que pode ser utilizada para

	detecção de erros.
Conjuntos de parâmetros ativos.	Fornece informações sobre VPS-SPS ativo, etc.
Informações sobre unidades de decodificação.	Tempo de Remoção de sub-imagens para funcionamento do HRD, assim como índice de unidade de decodificação.
Índice zero de nível temporal.	Fornece valores de índice zero ao nível temporal.
Aninhamento escalonável.	Provê um mecanismo para aninhar mensagem SEI para associação com pontos e camadas operacionais diferentes.
Informações de renovação regionais.	Fornece informações sobre região renovada e não renovada para renovação de decodificação gradual.

TABELA 1: Vista Panorâmica de Mensagens SEI

[0056] O acesso aleatório refere-se à decodificação de um fluxo de bits de vídeo que começa de uma imagem decodificada que não é a primeira imagem codificada no fluxo de bits. O acesso aleatório a um fluxo de bits é necessário em muitos aplicativos de vídeo, tais como execução de broadcast de fluxo contínuo, como, por exemplo, para que os usuários comutem entre canais diferentes, saltem para partes específicas do vídeo ou comutem para um fluxo de bits diferente para adaptação de fluxo (para taxa de bits, taxa de quadros ou capacidade de escalonamento de resolução espacial, por exemplo). O acesso aleatório é habilitado tendo-se uma estrutura de codificação que inclui imagens de Ponto de Acesso Aleatório (RAP) ou unidades de acesso, muitas vezes em intervalos regulares, para uma sequência de vídeo. Imagens de Renovação de Decodificador Instantâneas, imagens de Acesso Aleatório Limpas CRA e imagens de Acesso de Link Quebrado (BLA) são tipos de imagens RAP definidos no WD7 HEVC. Cada uma das imagens IDR, imagens CRA e imagens BLA inclui apenas fatias I. Entretanto, cada uma das imagens IDR, imagens CRA e imagens BLA difere com base em restrições de referência definidas.

[0057] As imagens IDR são especificadas na AVC e definidas de acordo com o WD7 HEVC. Enquanto as imagens IDR podem ser utilizadas para acesso aleatório, as imagens IDR são restringidas no sentido de que imagens que se seguem a uma imagem IDR em ordem de decodificação não podem utilizar imagens decodificadas antes da imagem IDR como referência. No exemplo mostrado nas Figuras 1 e 2, conforme descrito acima, a Pic₆ na sequência de vídeo 100 pode ser uma imagem IDR. Devido às restrições associadas a imagens IDR, os fluxos de bits que recorrem a imagens IDR para acesso aleatório podem reduzir significativamente a eficácia de codificação.

[0058] Para aperfeiçoar a eficácia de codificação, o conceito de imagens CRA foi introduzido na HEVC. De acordo com o WD7 HEVC, uma imagem CRA, como uma imagem IDR, inclui apenas fatias I. Entretanto é permitido que imagens que se seguem a uma imagem CRA em ordem de decodificação, mas precedem as imagens CRA em ordem de saída, utilizem imagens decodificadas antes da imagem CRA como referência. As imagens que se seguem a uma imagem CRA em ordem de decodificação, mas precedem a imagem CRA em ordem de saída são referidas como imagens dianteiras associadas à imagem CRA (ou imagens dianteiras da imagem CRA) As imagens dianteiras de uma imagem CRA são decodificáveis corretamente se a decodificação começar de uma imagem IDR ou CRA antes da imagem CRA atual. Entretanto, as imagens dianteiras de uma imagem CRA podem não ser decodificáveis corretamente quando ocorre o acesso aleatório da imagem CRA. Entretanto, as imagens dianteiras de uma imagem CRA podem não ser corretamente decodificáveis quando o acesso aleatório da imagem CRA ocorrer. Com referência ao exemplo mostrado nas Figuras 1 e 2, a Pic₉ pode ser uma imagem CRA

e a Pic₈ pode ser uma imagem dianteira da Pic₉. A Pic₈ é decodificável corretamente se o GOP₂ for acessado na Pic₉, mas pode não ser decodificável corretamente se o GOP₂ for acessado como Pic₉. Isto é devido ao fato de a Pic₇ poder não estar disponível se o GOP₂ for acessado como Pic₉. Para evitar propagação de erros de imagens de referência que podem não estar disponíveis dependendo de onde a decodificação começa, de acordo com o WD7 HEVC todas as imagens que se seguem a uma imagem CRA tanto em ordem de decodificação quanto em ordem de saída são restringidas a não utilizar qualquer imagem que precede a imagem CRA ou na ordem de decodificação ou na ordem de saída, (que incluem as imagens dianteiras) como referência. Além disto, as imagens dianteiras são tipicamente descartadas durante a decodificação de acesso aleatório.

[0059] Emenda de fluxos de bits refere-se a concatenação de dois ou mais fluxos de bits ou partes deles. Por exemplo, um primeiro fluxo de bits pode ser anexado por um segundo fluxo de bits, possivelmente com algumas modificações ou em um ou em ambos os fluxos de bits de modo a se gerar um fluxo de bits emendado. A primeira imagem codificada no segundo fluxo de bits é também referida como o ponto de emenda. Portanto, as imagens após o ponto de emenda no fluxo de bits emendado eram originadas do segundo fluxo de bits, enquanto as imagens que precedem o ponto de emenda no fluxo de bits emendado eram originadas do primeiro fluxo de bits. A emenda de fluxo de bits é tipicamente efetuada por elementos de emenda de fluxo de bits. Os elementos de emenda de fluxo de bits são frequentemente leves e muito menos inteligentes que os codificadores de vídeo. Por exemplo, um elemento de emenda de fluxo de bits pode não ser equipado com capacidades de

decodificação e codificação por entropia. A capacidade de escalonamento temporal é um aplicativo que pode utilizar emendas de fluxo de bits. A capacidade de escalonamento temporal pode referir-se à decodificação de uma sequência de vídeo a uma ou mais taxas de quadros. Por exemplo, uma sequência de vídeo pode ser passível de decodificação a 30 quadros por segundo (fps) ou 60 fps com base nas capacidades do sistema. De modo a se obter capacidade de escalonamento temporal, um sequência de vídeo pode incluir uma série de camadas temporais, onde cada camada temporal é uma sequência de vídeo codificada a uma taxa de quadros. A camada temporal com a taxa de quadros mais elevada pode ser referida como a camada temporal mais elevada. Uma série de camadas temporais podem ser emendadas umas nas outras de modo a se gerar a sequência de vídeo à taxa de quadros mais elevada, como, por exemplo, uma sequência de vídeo codificada com 30 fps é emendada com uma sequência de vídeo codificada que permite 60 fps.

[0060] A comutação entre fluxos de bits pode ser utilizada em ambientes de fluxo contínuo adaptativos. Uma operação de comutação entre fluxos de bits em determinada imagem no fluxo de bits para o qual se comuta é efetivamente uma operação de emenda de fluxo de bits na qual o ponto de emenda é o ponto de comutação entre fluxos de bits, isto é, a primeira imagem do fluxo de bits para o qual se comuta. Deve-se observar que a comutação entre fluxos de bits é usualmente efetuada em dois fluxos com a mesma estrutura de codificação. Ou seja, os dois fluxos têm a mesma estrutura de predição e a mesma locação de imagens IDR, imagens CRA, imagens P e imagens D, etc.

[0061] O conceito de imagem de acesso de link quebrado (BLA) foi também introduzido no WD7 HEVC após a introdução de imagens CRA e é baseado no conceito de imagens CRA. Uma

imagem BLA se origina tipicamente da emenda de fluxos de bits na posição de uma imagem CRA e, no fluxo de bits emendado, a imagem CRA de ponto de emenda é alterada para uma imagem BLA. A diferença mais essencial entre imagens BLA e imagens CRA é a seguinte: para uma imagem CRA as imagens dianteiras conexas são decodificáveis corretamente se a decodificação começar a partir de uma imagem RAP antes de uma imagem CRA em ordem de decodificação, e pode não ser decodificável corretamente quando o acesso aleatório começar da imagem CRA; para uma imagem BLA, as imagens dianteiras conexas podem não ser decodificáveis corretamente em todos os casos, mesmo quando a decodificação começar de uma imagem RAP antes da imagem BLA em ordem de decodificação. Deve-se observar que, para uma imagem CRA ou BLA específica, algumas das imagens dianteiras conexas são decodificáveis corretamente mesmo quando a imagem CRA ou BLA for a primeira imagem no fluxo de bits. Estas imagens dianteiras são referidas como imagens dianteiras decodificáveis (DLPs), e outras imagens dianteiras são referidas como imagens dianteiras não decodificáveis (NLPs). As NLPs são também referidas como etiquetadas para imagens de descarte (TFD) no WD9 HEVC. Deve-se observar que todas as imagens dianteiras associadas a uma imagem IDR são imagens DLP. A Tabela 2 é uma tabela incluída no WD7 HEVC que especifica as unidades NAL definidas de acordo com o WD7 HEVC. Conforme mostrado na Tabela 2, os tipos de unidades NAL no WD7 HEVC incluem imagem CRA, imagem BLA, imagem IDR, tipos de unidades NAL VPS, SPS, PPS e APS, que correspondem às imagens e conjunto de parâmetros descritos acima.

[0061] O conceito de imagem de acesso de link quebrado (BLA) foi também introduzido no WD7 HEVC após a introdução de imagens CRA e é baseado no conceito de imagens CRA. Uma

imagem BLA se origina tipicamente da emenda de fluxos de bits na posição de uma imagem CRA e, no fluxo de bits emendado, a imagem CRA de ponto de emenda é alterada para uma imagem BLA. A diferença mais essencial entre imagens BLA e imagens CRA é a seguinte: para uma imagem CRA as imagens dianteiras conexas são decodificáveis corretamente se a decodificação começar a partir de uma imagem RAP antes de uma imagem CRA em ordem de decodificação, e pode não ser decodificável corretamente quando o acesso aleatório começar da imagem CRA; para uma imagem BLA, as imagens dianteiras conexas podem não ser decodificáveis corretamente em todos os casos, mesmo quando a decodificação começar de uma imagem RAP antes da imagem BLA em ordem de decodificação. Deve-se observar que, para uma imagem CRA ou BLA específica, algumas das imagens dianteiras conexas são decodificáveis corretamente mesmo quando a imagem CRA ou BLA for a primeira imagem no fluxo de bits. Estas imagens dianteiras são referidas como imagens dianteiras decodificáveis (DLPs), e outras imagens dianteiras são referidas como imagens dianteiras não decodificáveis (NLPs). As NLPs são também referidas como etiquetadas para imagens de descarte (TFD) no WD9 HEVC. Deve-se observar que todas as imagens dianteiras associadas a uma imagem IDR são imagens DLP. A Tabela 2 é uma tabela incluída no WD7 HEVC que especifica as unidades NAL definidas de acordo com o WD7 HEVC. Conforme mostrado na Tabela 2, os tipos de unidades NAL no WD7 HEVC incluem imagem CRA, imagem BLA, imagem IDR, tipos de unidades NAL VPS, SPS, PPS e APS, que correspondem às imagens e conjunto de parâmetros descritos acima.

nal_unit_type	Conteúdo de unidade NAL e estrutura de sintaxe RDSP	Classe de tipo de unidade

		NAL
0	Não especificado.	Não VCL.
1	Fatia codificada de uma imagem não RAP, não TFD e não TLA slice layer rbsp()	VCL.
2	Fatia codificada de uma imagem TFD slice layer rbsp()	VCL.
3	Fatia codificada de uma imagem TLA não TFD slice layer rbsp()	VCL.
4,5	Fatia codificada de uma imagem CRA slice layer rbsp()	VCL.
6,7	Fatia codificada de uma imagem BLA slice layer rbsp()	VCL.
8	Fatia codificada de uma imagem IDR slice layer rbsp()	VCL.
9... 24	Reservado	n/a
25	Conjunto de parâmetros de vídeo video parameter set rbsp()	Não VCL.
26	Conjunto de parâmetros de sequência seq parameter set rbsp()	Não VCL.
27	Conjunto de parâmetros de imagem pic parameter set rbsp()	Não VCL.
28	Conjunto de parâmetros de adaptação aps rbsp()	Não VCL.
29	Delimitador de unidade de acesso access unit delimiter rbsp()	Não VCL.
30	Dados de enchimento filler data rbsp()	Não VCL.
31	Informações de aperfeiçoamento suplementares (SEI) sei rbsp()	Não VCL.
32... 47	Reservado.	n/a
48... 63	Não especificado.	Não VCL.

TABELA 2: Códigos de tipo de unidade NAL e classes de tipo de unidade NAL do WD7 HEVC

[0062] De modo a simplificar as alocações de unidades NAL, S. Kanumuri, G. Sullivan "Refinamento de Suporte de Ponto de Acesso Aleatório", 10°. Encontro, Estocolmo, SE, julho de 2012, Doc. JCTVC-J0344 (daqui por diante "Kanumuri") que é incorporado em sua totalidade à guisa de referência, propõe (1) uma restrição a imagens IDR de modo

que não haja imagens dianteiras associadas a qualquer imagem IDR (isto é, nenhuma imagem pode seguir-se a uma imagem IDR em ordem de decodificação e preceder uma imagem IDR em ordem de saída), e (2) tipos de Unidade NAL de alocação modificada 4 a 7 definidos de acordo com a Tabela 2 acima para imagem RAP da seguinte maneira:

Tipo de unidade NAL	Descrição	Tipos de SAP possíveis
4	Imagem CRA	1,2,3
5	Imagem BLA	1,2,3
6	Imagem BLA sem imagens TFD conexas	1,2
7	Imagem BLA sem imagens dianteiras	1

TABELA 3: Tipos de unidade NAL propostos de acordo com Kanumuri

[0063] Na Tabela 3, os tipos SAP referem-se a tipos de Ponto de Acesso de Fluxo definidos no ISO/IEC 14496-12, 4°. Ed., "Tecnologia da Informação - Codificação de Objetos Áudio Visuais - Parte 12: Formato de Arquivo de Meios Base ISO", w12640, 100°. Encontro MPEG, Genebra, abril de 2012, que é incorporado em sua totalidade à guisa de referência. Conforme descrito acima, as imagens IDR e as imagens BLA/CRA são funcionalmente diferentes para comutação entre fluxos de bits, embora elas sejam funcionalmente as mesmas para acesso aleatório (aplicativos de busca, por exemplo). Para comutação entre fluxos de bits em imagens IDR, um sistema de codificação de vídeo pode saber ou supor que a apresentação possa ser contínua sem aumento na intensidade da corrente elétrica (falta de imagens não apresentadas, por exemplo). Isto é porque as imagens que se seguem a uma imagem IDR em ordem de decodificação não podem utilizar imagens decodificadas antes da imagem IDR como referência (isto é, as imagens dianteiras associadas a uma imagem IDR

são DLP). Entretanto, para comutação entre fluxos de bits em imagens BLA, alguma decodificação superposta de uma ou mais imagens de ambos os fluxos pode ser necessária para assegurar que a apresentação seja contínua. Esta decodificação superposta pode não ser atualmente possível para decodificadores em conformidade com o WD7 HEVC sem capacidade adicional. Sem capacidade adicional pode não haver quaisquer imagens nas posições de imagem TDF conexas a serem apresentadas uma vez que elas podem ter sido descartadas. Isto pode resultar em uma apresentação que não é necessariamente contínua. Além disto, mesmo se a imagem BLA for uma imagem BLA sem imagens TFD conexas, o problema é o mesmo uma vez que imagens TFD que estavam presentes no fluxo de bits original podem ser descartadas. Além disto, se não tiver havido imagens TFD no fluxo de bits original, então a imagem CRA (posteriormente alterada para ser uma imagem BLA devido a emenda/comutação de fluxos de bits, etc.) pode ter sido codificada como uma imagem IDR. Portanto, a não marcação de imagens IDR com imagens dianteiras como imagens IDR (isto é, não permitir que imagens IDR tenham imagens dianteiras) conforme proposto por Kanumuri, torna as imagens IDR menos amigáveis para sistemas para comutação entre fluxos de bits.

[0064] Do ponto de vista dos sistemas de fluxo contínuo, como, por exemplo, o fluxo contínuo dinâmico através do http (DASH) é benéfico ser capaz de identificar facilmente qual imagem é uma imagem RAP e, se a decodificação começar de uma imagem RAP, qual é o tempo de apresentação inicial (um valor de total de ordens de imagem inicial (POC), por exemplo). Portanto, os desenhos de alocação existentes de tipos de unidades NAL para diferentes imagens RAP assim como para imagens DLP e TFD podem ser também aperfeiçoados para serem mais amigáveis a sistemas de fluxo contínuo. De

acordo com os desenhos existentes, para cada imagem RAP os sistemas têm que verificar se há imagens DLP conexas de modo a se saber se há o tempo de apresentação da imagem RAP propriamente dita é o tempo de apresentação mais antigo quando a decodificação começa a partir da imagem RAP. Além disto, o sistema tem que verificar e comparar os tempos de apresentação de todas as imagens DLP para saber o valor no tempo de apresentação mais antigo.

[0065] Os padrões de codificação de vídeo incluem uma especificação de modelo de armazenamento de vídeo. Na AVC e HEVC o modelo de armazenamento é referido como decodificador de referência hipotético (HRD), que inclui um modelo de armazenamento tanto de um buffer de imagens codificadas (CPB) quanto um armazenador de imagens decodificadas (DPB). De acordo com o WD7 HEVC, o HRD é definido como um modelo de decodificador hipotético que especifica restrições à variabilidade de fluxos de unidades NAL conformadoras ou fluxos de bits conformadores que um processo de codificação pode produzir. Assim, na AVC e na HEVC a conformação aos fluxos de bits e a conformação ao decodificador são especificadas como partes da especificação do HRD. De acordo com o WD7 HEVC, um CPB é um armazenador de primeiro-dentro-primeiro-fora que contém unidades de acesso em ordem de decodificação e um DPB é um armazenador que guarda imagens decodificadas para referência. Os comportamentos do CPB e do DPB são matematicamente especificados de acordo com o HRD. O HRD impõe diretamente restrições à temporização, tamanhos de armazenador e taxa de bits, e impõe indiretamente restrições às características e estatísticas de fluxos de bits. Um conjunto completo de parâmetros de HRD inclui cinco parâmetros básicos: retardo de remoção de CPB inicial, tamanho de CPB, taxa de bits, retardo de saída de

TPB inicial e tamanho de DPB. De acordo com o WD7 HEVC, os parâmetros de HRD podem ser incluídos em parâmetros de informação de capacidade e utilização de vídeo (VUI) e os parâmetros VUI podem ser incluídos em um SPS. Deve-se observar que, embora o HRD seja referido como decodificador, o HRD é tipicamente necessário no lado de codificador para garantir a conformação do fluxo de bits, e tipicamente não necessário no lado de decodificador. O WD7 HEVC especifica dois tipos de fluxos de bits para conformação do HRD, a saber, o Tipo I e o Tipo II. O WD7 HEVC especifica também dois tipos de conformação de decodificador, a saber, conformação de decodificador de temporização de saída e conformação de decodificador de ordem de saída.

[0066] Nos modelos de HRD AVC e HEVC, a decodificação ou remoção de CPB é baseado em unidades de acesso e supõe-se que a decodificação de imagens seja instantânea. O tempo necessário para decodificar uma imagem nos aplicativos do mundo real não pode ser igual a zero. Assim, em aplicações práticas, se um decodificador conformador seguir rigorosamente os tempos de decodificação sinalizados, como, por exemplo, nas mensagens SEI de temporização de imagens, para iniciar a decodificação de unidades de acesso, então o tempo mais antigo possível em que uma imagem decodificada específica pode ser transmitida é igual ao tempo de decodificação dessa imagem específica mais o tempo necessário para decodificar essa imagem específica.

[0067] Um comportamento de CPB baseado em sub-imagens semelhante ao comportamento de CPB descrito em Ye-Kui Wang *et alii*, "Funcionamento de CPB baseado em Sub-Imagens", 9°. Encontro, Genebra, CH, maio de 2012, JCTVC-I0588 (daqui por diante "Wang") foi incluído no WD7 HEVC. O CPB baseado em sub-imagens Wang permite que a remoção de CPBs seja

efetuada ou ao nível de unidade de acesso (AU) ou ao nível de sub-imagem. Permitir a remoção de CPB ou ao nível de AU ou ao nível de sub-imagem ajuda a obter retardo de CODEC reduzido de qualquer maneira inter-operante. Quando a remoção de CPB ocorre ao nível da unidade de acesso, uma unidade de acesso é removida do CPB cada vez que uma operação de remoção ocorre. Quando a remoção do CPB ocorre ao nível de sub-imagem, uma unidade de decodificação (DU) que contém uma ou mais fatias é removida do CPB cada vez que ocorre uma operação de remoção.

[0068] Informações de temporização de remoção de CPB ao nível de sub-imagem podem ser sinalizadas além de informações de temporização de remoção de CPB ao nível de AU. Quando informações de temporização de remoção de CPB estão presentes para remoção tanto ao nível de AU quanto ao nível de sub-imagem, um decodificador pode escolher acionar o CPB ou ao nível de AU ou ao nível de sub-imagem. Deve-se observar que de modo que a mensagem SEI de temporização da imagem atual e que o mecanismo para permitir remoção de CPB de HRD tanto ao nível de AU tanto ao nível de DU ao mesmo tempo para se obter retardo de sub-imagem, DUs devem ser enviadas antes que a AU inteira seja codificada, e as mensagens SEI ao nível de AU ainda não podem ser enviadas antes que a AU inteira seja codificada.

[0069] De acordo com o WD7 HEVC, as informações de temporização podem incluir informações que definem a distância temporal entre os tempos de saída de HRB de duas imagens consecutivas. O WD7 HEVC define os elementos de sintaxe de informação de temporização seguintes:

time_scale é o número de unidades de tempo que passam em um segundo. Por exemplo, um sistema de coordenadas de tempo que mede o tempo utilizando um relógio de 27 MHz tem uma time_scale de 27.000.000, a time_scale será maior que 0.

num_units_in_tick é o número de unidades de tempo de um relógio que funciona **time_scale** que corresponde a um incremento (chamado um pulso de relógio), de um contador de pulsos de relógio. **num_units_in_tick** será maior que 0.

[0070] Assim, com base nos valores **time_scale** e **num_units_in_tick**, a chamada variável de pulso de relógio, t_C , pode ser derivada de maneira seguinte:

$$t_C = \text{num_units_in_tick} \div \text{time_scale} \quad (1)$$

[0071] De acordo com o WD7 HEVC, a variável de pulso de relógio pode ser utilizada para restringir os tempos de saída de HRD. Ou seja, em alguns casos pode ser necessário que a diferença entre os tempos de apresentação de duas imagens contínuas na ordem de saída (isto é, a primeira e a segunda imagem) seja igual ao pulso de relógio. O WD7 HEVC inclui o elemento de sintaxe **fixed_pic_rate_flag**, que indica se a diferença entre os tempos de apresentação de duas imagens contínuas em ordem de saída é igual ao pulso de relógio. O elemento de sintaxe **fixed_pic_rate_flag** pode ser incluído em um conjunto de parâmetros VUI que podem ser incluídos em um SPS. No WD7 HEVC, quando o elemento de sintaxe **fixed_pic_rate_flag** é igual a um, a distância temporal entre os tempos de saída de HRD de quaisquer duas imagens consecutivas em ordem de saída é restringida de modo a ser igual ao pulso de relógio determinado sujeito ou a uma ou à outra das seguintes condições seguintes serem verdadeiras: (1) a segunda imagem está na mesma sequência de vídeo codificada que a primeira imagem; ou (2) a segunda imagem está em uma sequência de vídeo codificada diferente da primeira imagem e **fixed_pic_rate_flag** é igual a 1 na sequência de vídeo codificada que contém a segunda imagem e o valor de **num_units_in_tick** dividido por **time_scale** é o mesmo para ambas as sequências de vídeo codificadas. Quando

o elemento de sintaxe **fixed_pic_rate_flag** é igual a zero, nenhuma de tais restrições se aplica à distância temporal entre os tempos de saída de HRD de quaisquer duas imagens consecutivas (isto é, primeira e segunda imagem) em ordem de saída. Deve-se observar que, quando **fixed_pic_rate_flag** não está presente, ele é inferido como sendo igual a 0. Deve-se observar que, de acordo com o WD7 HEVC quando **fixed_pic_rate_flag** é igual a 1 a adaptação de fluxos baseada na capacidade de escalonamento temporal exigiria a alteração do valor ou de **time_scale** ou de **num_units_in_tick** no caso de algumas camadas temporais mais elevadas serem descartadas. Deve-se observar que o WD7 HEVC provê os seguintes elementos semânticos para **fixed_pic_rate_flag**:

Quando **fixed_pic_rate_flag** é igual a 1 para uma sequência de vídeo codificada que contém a imagem n , o valor computado para $\Delta t_{o,dbp}(n)$ especificado na Equação C-13 será igual a t_C , conforme especificado na equação C-1 (utilizando-se o valor de t_C para a sequência de vídeo codificada que contém a imagem n , quando uma ou mais das condições seguintes forem verdadeiras para a imagem n_n seguinte, que é especificada para uso na Equação C-13):

- A imagem n_n está na mesma sequência de vídeo codificada que a imagem n .
- A imagem n_n está em uma sequência de vídeo codificada diferente e **fixed_pic_rate_flag** é igual a 1 na sequência de vídeo codificada que contém a imagem n_n e o valor de **num_units_in_tick** dividido por **time_scale** e o mesmo para ambas as sequências de vídeo codificadas.

[0072] Onde a Equação C-1 corresponde a equação (1) e a Equação C-13 é definida no WD7 HEVC de maneira seguinte:

$$\Delta t_{o,dbp}(n) = t_{o,dbp}(n_n) - t_{o,dbp}(n) \quad (2)$$

[0073] Em vista das características de temporização e acesso aleatório acima mencionadas associadas ao WD7 HEVC, esta revelação descreve técnicas que podem ser utilizadas para reduzir retardo em aplicativos de vídeo, tais como aplicativos de conversação, e proporcionar aperfeiçoamentos no acesso aleatório a uma sequência de vídeo codificada. Em um exemplo, esta revelação descreve técnicas para alocar tipos de unidades NAL. Em outro exemplo, esta revelação descreve, para nível de sub-imagem ou nível de unidade de decodificação, o comportamento de um HRD. Em outro exemplo, esta revelação descreve técnicas para referência de IDs de conjunto de parâmetros. Em ainda outro exemplo, esta revelação descreve técnicas para prover elementos semânticos aperfeiçoados para o elemento de sintaxe **fixed_pic_rate_flag**. Deve-se observar que qualquer uma e todas as combinações destas técnicas e outras técnicas aqui descritas podem ser incorporadas a um sistema de codificação e decodificação de vídeo.

[0074] A Figura 3 é um diagrama de blocos que mostra um sistema de codificação e decodificação de vídeo exemplar 10 que pode utilizar as técnicas aqui descritas. Em particular, o sistema de codificação e decodificação de vídeo pode utilizar as técnicas aqui descritas relacionadas com (1) a alocação de tipos de unidades NAL, (2) comportamento de HRD ao nível de sub-imagem ou ao nível de unidade de decodificação, (3) a referência de IDs de conjunto de parâmetros, (4) elementos semânticos aperfeiçoados para **fixed_pic_rate_flag** ou qualquer uma ou todas as combinações destas técnicas. O sistema de codificação e decodificação de vídeo 10 é um exemplo de sistema de vídeo que pode ser utilizado para qualquer um dos aplicativos de vídeo seguinte: repetição local, fluxo

contínuo, aplicativos de broadcast, multicast ou de conversação. O aparelho de origem 12 e o aparelho de destino 14 são exemplos de aparelhos de codificação nos quais o aparelho de origem 12 gera dados de vídeo codificados para transmissão para o aparelho de destino 14. Em alguns exemplos, o aparelho de origem 12 e o aparelho de destino 14 podem funcionar de maneira substancialmente simétrica de modo que cada um do aparelho de origem 12 e do aparelho de destino 14 inclua componentes de codificação e decodificação de vídeo. Consequentemente o sistema 10 pode ser configurado para suportar transmissão de vídeo unidirecional ou bidirecional entre o aparelho de origem 12 e o aparelho de destino 14.

[0075] Embora as técnicas aqui descritas sejam descritas em conjunto com o aparelho de origem 12 e o aparelho de destino 14, as técnicas podem ser executadas por qualquer aparelho de codificação e/ou decodificação de vídeo digital. As técnicas desta revelação podem ser também executadas por um pré-processador de vídeo. Além disto, embora as técnicas desta revelação sejam geralmente descritas como sendo executadas por um aparelho de codificação de vídeo e um aparelho de decodificação de vídeo, as técnicas podem ser também executadas por um codificador/decodificador de vídeo, tipicamente referido "CODEC". Assim, cada um do codificador de vídeo 20 e do decodificador de vídeo 30 da Figura 3 pode ser incluído em um ou mais codificadores ou decodificadores, ou um ou outro dos quais pode ser integrado como parte de um codificador/decodificador (CODEC) combinado em um respectivo aparelho. Além disto, um aparelho que inclui o codificador de vídeo 20 e o decodificador de vídeo 30 pode compreender um circuito integrado, um microprocessador e um aparelho de comunicação sem fio tal como um telefone

celular. Embora não mostrado na Figura 3, sob alguns aspectos, o codificador de vídeo 20 e o decodificador de vídeo 30 podem ser, cada um, integrados com um codificador e decodificador de áudio e podem incluir unidades MUX-DEMUX apropriadas ou outro hardware e software, para processar a codificação tanto de áudio quanto de vídeo em um fluxo de dados comum ou em um fluxo de dados separado. Se aplicável, as unidades MUX-DEMUX podem conformar-se ao protocolo de multiplexador ITU-H.223 ou a outros protocolos tais como o protocolo de datagrama de usuário (UDP).

[0076] Conforme mostrado na Figura 3, o sistema 10 inclui um aparelho de origem 12 que provê dados de vídeo codificados a serem decodificados posteriormente por um aparelho de destino 14. Em particular o aparelho de origem 12 fornece dados de vídeo codificados ao aparelho de destino 14 por meio de um meio passível de leitura por computador 16. O aparelho de destino 14 pode receber os dados de vídeo codificados a serem decodificados por meio do meio passível de leitura por computador 16. O aparelho de origem 12 e o aparelho de destino 14 podem compreender qualquer um de uma ampla faixa de aparelhos, que inclui computadores de mesa, computadores notebook (isto é, laptop, computadores tablet, conversores set-top-box, aparelhos telefônicos tais como os chamados telefones "inteligentes", os chamados dispositivos "inteligentes", televisões, câmeras, aparelhos de exibição, tocadores de meios digitais, consoles para jogos de vídeo, aparelhos de fluxo contínuo de vídeo ou semelhantes. Em alguns casos, o aparelho de origem 12 e o aparelho de destino 14 podem ser equipados para comunicação sem fio.

[0077] O meio passível de leitura por computador 16 pode compreender qualquer tipo de meio ou aparelho capaz de mover os dados de vídeo codificados do aparelho de origem

12 até o aparelho de destino 14. O meio passível de leitura por computador 16 pode incluir meios transitórios, tais como um broadcast sem fio ou uma transmissão em rede cabeada ou meios de armazenamento (isto é, meios de armazenamento não transitórios) tais como disco rígido, unidade flash, um disco compacto, um disco de vídeo digital, um disco Blu-ray, ou outros meios passíveis de leitura por computador. Em alguns exemplos, um servidor de rede (não mostrado) pode receber dados de vídeo codificados do aparelho de origem 12 e enviar os dados de vídeo codificados ao aparelho de destino 14, por meio de uma transmissão em rede. Da mesma maneira, um aparelho de computação de uma instalação de produção de meios, tal como uma instalação de estampagem de discos, pode receber dados de vídeo codificados do aparelho de origem 12 e produzir um disco que contém os dados de vídeo codificados.

[0078] Em um exemplo, o meio passível de leitura por computador 16 pode compreender um meio de comunicação para permitir que o aparelho de origem 12 transmita dados de vídeo codificados diretamente para o aparelho de destino 14 em tempo real. Os dados de vídeo codificados podem ser modulados de acordo com um padrão de comunicação, tal como um protocolo de comunicação sem fio, e transmitidos para o aparelho de destino 14. O meio de comunicação pode compreender qualquer meio de comunicação sem fio ou cabeado, tal como um espectro de radiofrequência ou uma ou mais linhas de transmissão físicas. O meio de comunicação pode fazer parte de uma rede baseada em pacotes, tal como uma rede de área local, uma rede de área estendida ou uma rede global como a Internet. O meio de comunicação pode incluir roteadores, comutadores, estações base ou qualquer outro equipamento que possa ser útil para facilitar as

comunicações do aparelho de origem 12 com o aparelho de destino 14.

[0079] O aparelho de armazenamento pode incluir qualquer um de diversos meios de armazenamento de dados distribuídos ou localmente acessados, tais como uma unidade rígida, discos Blu-ray, DVDs, CD-ROMs, memória flash, memória volátil ou não volátil ou quaisquer outros meios de armazenamento digital adequados para armazenar dados de vídeo codificados. Em outro exemplo, o aparelho de armazenamento pode corresponder a um servidor de arquivos ou outro aparelho de armazenamento intermediário que possa armazenar o vídeo codificado gerado pelo aparelho de origem 12. O aparelho de destino 14 pode acessar dados de vídeo armazenados do aparelho de armazenamento por meio de fluxo contínuo ou download. O servidor de arquivos pode ser qualquer tipo de servidor capaz de armazenar dados de vídeo codificados e transmitir esses dados de vídeo codificados para o aparelho de destino 14. Servidores de arquivos exemplares incluem um servidor da Web (para um site da Web, por exemplo), um servidor FTP, aparelhos de armazenamento anexados à rede (NAS) ou uma unidade de disco local. O aparelho de destino 14 pode acessar os dados de vídeo codificados através de qualquer conexão com dados padrão, inclusive uma conexão com a Internet. Isto pode incluir um canal sem fio (uma conexão Wi-Fi, por exemplo), uma conexão cabeada (como, por exemplo, DSL, modem a cabo, etc.) ou uma combinação de ambos que seja adequado para acessar dados de vídeo codificados armazenados em um servidor de arquivos. A transmissão de dados de vídeo codificados do aparelho de armazenamento pode ser uma transmissão de fluxo contínuo, uma transmissão de download ou uma combinação delas.

[0080] As técnicas desta revelação não estão necessariamente limitadas a aplicativos ou parâmetros sem

fio. As técnicas podem ser aplicadas à codificação de vídeo em suporte a qualquer um de diversos aplicativos multimídia, como, por exemplo, broadcast de televisão através do ar, transmissões de televisão a cabo, transmissões de televisão por satélite, transmissões de vídeo em fluxo contínuo da Internet, tais como fluxo contínuo adaptativo dinâmico através do http (DASH), vídeo digital que é codificado em um meio de armazenamento de dados, decodificação de vídeo digital armazenado em um meio de armazenamento de dados, ou outros aplicativos.

[0081] No exemplo da Figura 3 o aparelho de origem 12 inclui uma fonte de vídeo 18, uma unidade de estrutura de codificação 19, um codificador de vídeo 20, uma unidade de encapsulamento 21 e uma interface de saída 22. O aparelho de destino 14 inclui uma interface de entrada 28, uma unidade de desencapsulamento 29, um decodificador de vídeo 30 e um aparelho de exibição 32. Em outro exemplo, o aparelho de origem 12 e o aparelho de destino 14 podem incluir outros componentes ou disposições. Por exemplo, o aparelho de origem 12 pode receber dados de vídeo de uma fonte de vídeo externa 18, tal como uma câmera externa. Da mesma maneira, o aparelho de destino 14 pode formar interface com um aparelho de exibição externo, em vez de incluir um aparelho de exibição integrado. Os componentes do aparelho de origem 12 e do aparelho de destino 14 podem ser, cada um, implementados como qualquer um de diversos circuitos adequados, tais como um ou mais micro-processadores, processadores de sinais digitais (DSPs), circuitos integrados específicos de aplicativo (ASICs), arranjos de portas programáveis no campo (FPGAs), lógica discreta, software, hardware, firmware ou quaisquer combinação deles. Quando as técnicas aqui descritas são implementadas parcialmente em software, um aparelho pode

armazenar instruções para o software em um meio passível de leitura por computador não transitório adequado e executar as instruções em hardware utilizando um ou mais processadores para executar as técnicas.

[0082] A fonte de vídeo 18 do aparelho de origem 12 pode incluir um aparelho de captação de vídeo tal como uma câmera de vídeo, um arquivo de vídeo que contém vídeo previamente captado e/ou uma interface de alimentação de vídeo para receber vídeo de um provedor de conteúdo de vídeo. Como outra alternativa, a fonte de vídeo 18 pode gerar dados baseados em gráficos de computador como o vídeo de origem ou uma combinação de vídeo ao vivo, vídeo arquivado e vídeo gerado por computador. Em alguns casos, se a fonte de vídeo 18 for uma câmera de vídeo, o aparelho de origem 12 e aparelho de destino 14 podem formar os chamados telefones com câmera ou telefones com vídeo. Conforme mencionado acima, contudo, as técnicas descritas nesta revelação podem ser aplicadas à codificação de vídeo em geral e podem ser aplicadas a aplicativos sem fio e/ou cabeados. Em cada caso, o vídeo captado, pré-captado ou gerado por computador pode ser recebido pelo codificador de vídeo 20. A interface de saída 22 pode ser configurada para transmitir dados de vídeo codificados, tais como uma sequência de vídeo codificada, em um meio passível de leitura por computador 16. Em alguns exemplos, uma sequência de vídeo codificada pode ser transmitida da interface de saída 22 para um aparelho de armazenamento. A interface de entrada 28 do aparelho de destino 14 recebe os dados de vídeo codificados do meio passível de leitura por computador 16. O aparelho de exibição 32 exibe os dados de vídeo decodificados ao usuário e pode compreender qualquer um de diversos aparelhos de exibição, tais como um tubo de raios catódicos (CRT), uma tela de cristal líquido (LCD),

uma tela de plasma, uma tela de diodo emissor de luz orgânico (OLED) ou outro tipo de aparelho de exibição.

[0083] A unidade de estrutura de codificação 19, o codificador de vídeo 20, a unidade de encapsulamento 21, a unidade de desencapsulamento 29 e o decodificador de vídeo 30 podem funcionar de acordo com padrão de codificação de vídeo, tal como a HEVC vindoura descrita acima e podem conformar-se geralmente ao Modelo de Teste HEVC (HM). Alternativamente, o codificador de vídeo 20 e o decodificador de vídeo 30 podem funcionar de acordo com outros padrões patenteados ou industriais, tais como o padrão ITU-T H.264, alternativamente referido como MPEG-4 Parte 10, Codificação Avançada de Vídeo (AVC) ou extensões de tais padrões. A unidade de estrutura de codificação 19, o codificador de vídeo 20, a unidade de encapsulamento 21, a unidade de desencapsulamento 29 e o decodificador de vídeo 30 podem funcionar também de acordo com uma versão modificada de um padrão de codificação de vídeo em que, a versão modificada do padrão de codificação de vídeo é modificada de modo a incluir qualquer uma e todas as combinações das técnicas aqui descritas.

[0084] O codificador de vídeo 20 pode dividir um quadro ou imagem de vídeo em uma série de blocos de vídeo de tamanho igual, tais como uma CU, descrita no WD7 HEVC. Uma CU inclui um nó de codificação e unidades de predição (PUs) e unidades de transformada (TUs) associadas ao nó de codificação. O tamanho da CU corresponde ao tamanho do nó de codificação e deve ser de conformação quadrada. O tamanho da CU pode variar de 8x8 pixels até o tamanho do bloco de árvore com um máximo de 64x64 pixels ou mais. Cada CU pode conter uma ou mais PUs e uma ou mais TUS. Os dados de sintaxe associados a uma CU podem descrever, por exemplo, o particionamento da CU em uma ou mais PUs. Os

modos de particionamento podem diferir entre se a CU é saltada ou codificada no modo direto, codificada no modo de intra-predição ou no modo de inter-predição. As PUs podem ser particionadas para serem de conformação não quadrada. Os dados de sintaxe associados a uma CU podem descrever também, por exemplo, o particionamento da CU em uma ou mais TUs de acordo com uma transformação quad-tree. Uma TU pode ser de conformação quadrada ou não quadrada (retangular, por exemplo).

[0085] O padrão HEVC proporciona transformações de acordo com TUs, que podem ser diferentes para CUs diferentes. As TUs são tipicamente dimensionadas com base no tamanho das PUs dentro de uma dada CU definida para uma LCU particionada, embora este possa não ser sempre o caso. As TUs são tipicamente do mesmo tamanho ou menores que as PUs. Em alguns exemplos, amostras residuais que correspondem a uma CU podem ser sub-divididas em unidades menores utilizando-se uma estrutura de transformação quad-tree, conhecida como "transformação quad-tree residual" (RQT). Os nós-folha da RQT podem ser referidos como unidades de transformada (TUs). Os valores de diferença de pixel associados às TUs podem ser transformados de modo a se produzirem coeficientes de transformada, que podem ser quantificados.

[0086] Uma CU-folha pode incluir uma ou mais unidades de predição (PUs). Em geral, uma PU representa uma área espacial que corresponde a toda ou a uma parte da CU correspondente, e pode incluir dados para recuperar uma amostra de referência para a PU. Além do mais, uma PU inclui dados relacionados com predição. Por exemplo, quando a PU é codificada de maneira intra-modal, dados para a PU podem ser incluídos em uma transformação quad-tree residual (RQT), que pode incluir dados que descrevem um modo de

intra-predição para uma TU que corresponde à PU. Como outro exemplo, quando a PU é codificada de maneira inter-modal, a PU pode incluir dados que definem um ou mais vetores de movimento para a PU. Os dados que definem o vetor de movimento para uma PU podem descrever, um componente horizontal do vetor de movimento, um componente vertical do vetor de movimento, uma resolução para o vetor de movimento (precisão de um quarto de pixel ou precisão de um oitavo de pixel, por exemplo), uma imagem de referência que o vetor de movimento indica e/ou uma lista de imagens de referência (Lista 0, Lista 1 ou Lista C, por exemplo) para o vetor de movimento.

[0087] Uma CU-folha que tem uma ou mais PUs pode incluir também uma ou mais unidades de transformada (TUs). As unidades de transformada podem ser especificadas utilizando-se uma RQT (também referida como estrutura de transformação quad-tree de TU), conforme discutido acima. Por exemplo, um indicador de divisão pode indicar se uma CU-folha é dividida em quatro unidades de transformada. Em seguida, cada unidade de transformada pode ser dividida também em outras sub-TUs. Quando uma TU não é dividida adicionalmente, ela pode ser referida como uma TU-folha. Geralmente, para intra-codificação todas as TUs-folha pertencentes a uma CU-folha compartilham o mesmo modo de intra-predição. Ou seja, o mesmo modo de intra-predição é geralmente aplicado para calcular valores preditos para todas as TUs de uma CU-folha. Para intra-codificação, um codificador de vídeo pode calcular um valor residual para cada TU-folha utilizando o modo de intra-predição, como a diferença entre a parte da CU que corresponde à TU e o bloco original. Uma TU não está necessariamente limitada ao tamanho de uma PU. Assim, as TUs podem ser maiores ou menores que uma PU. Para intra-codificação, uma PU pode ser

co-localizada com uma TU-folha correspondente para a mesma CU. Em alguns exemplos, o tamanho máximo de uma TU-folha pode corresponder ao tamanho da CU-folha correspondente.

[0088] Além do mais, as TUs das CUs-folha podem ser também associadas a estrutura de dados de transformação quad-tree, referidas como transformações quad-tree residuais (RQTs). Ou seja, uma CU-folha pode incluir uma transformação quad-tree que indica como a CU-folha é particionada em TUs. O nó-raiz de uma transformação quad-tree de TU corresponde geralmente a uma CU-folha, enquanto o nó-raiz de uma transformação quad-tree de CU corresponde geralmente a um bloco de árvore (ou LCU). As TUs da RQT que não são divididas são referidas como TUs-folha. Em geral, esta revelação utiliza os termos CU e TU para refere-se a uma CU-folha e TU-folha, respectivamente, a menos que observado de outra maneira. Esta revelação utiliza o termo bloco para referir-se a qualquer uma de uma CU, PU ou TU, no contexto da HEVC, ou estrutura de dados semelhantes no contexto de outros padrões (macro-blocos e sub-blocos delas no H.264/AVC.).

[0089] Como exemplo, o HM suporta predição em diversos tamanhos de PU. Supondo-se que o tamanho de CU específica seja $2N \times 2N$, o HM suporta intra-predição em tamanhos de PU de $2N \times 2N$ ou $N \times N$ e inter-predição em tamanhos de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$ ou $N \times N$. O HM suporta também particionamento assimétrico para inter-predição em tamanhos de PU de $2N \times nU \times 2N \times nD$, $nL \times n$ e $nR \times 2N$. No particionamento assimétrico, uma direção de uma CU não é particionada, enquanto a outra direção é particionada em 25% e 75%. A parte da CU que corresponde à partição de 25% é indicada por um n seguido de uma indicação de "Para Cima", "Para Baixo", "Esquerda" ou "Direita". Assim, por exemplo, " $2N \times nU$ " refere-se a uma CU $2N \times 2N$ que é particionada em

sentido horizontal com uma PU $2N \times 0,5N$ no topo e uma PU $2N \times 1,5N$ na base.

[0090] Nesta revelação, " $N \times N$ " e " N por N " podem ser utilizados de maneira intercambiável para referir a dimensões de pixel de um bloco de vídeo em termos das dimensões vertical e horizontal, como, por exemplo, 16×16 pixels ou 16 por 16 pixels. Em geral, um bloco de 16×16 terá 16 pixels na direção vertical ($y = 16$) e 16 pixels na direção horizontal ($x = 16$). Da mesma maneira, um bloco de $N \times N$ tem geralmente N pixels na direção vertical e N pixels na direção horizontal, onde N representa um valor de número inteiro não negativo. Os pixels em um bloco podem ser dispostos em fileiras e colunas. Além do mais, não é necessário que os blocos tenham o mesmo número de pixels na direção horizontal como na direção vertical. Por exemplo, os blocos podem compreender $N \times M$ pixels, o M não é necessariamente igual a N .

[0091] Em seguida a codificação intra-preditiva ou inter-preditiva que utiliza as PUs de uma CU, o codificador de vídeo 20 pode calcular dados residuais para as TUs da CU. As PUs podem compreender dados de sintaxe que descrevem um método ou modo para gerar dados de pixels preditivos no domínio espacial (também referido como domínio de pixel) e as TUs podem compreender coeficientes no domínio de transformada em seguida à aplicação de uma transformada, como, por exemplo, uma transformada de co-seno discreta (DCT), uma transformada de número inteiro, uma transformada de wavelet ou uma transformada conceptualmente semelhante a dados de vídeo residuais. Os dados residuais podem corresponder a diferenças de pixel entre os pixels da imagem não codificada e os valores de predição que correspondem às PUs. O codificador de vídeo 20 pode formar as TUs incluindo os dados residuais para a CU e em seguida

transformar as TU de modo a produzir coeficientes de transformada para a CU.

[0092] Em seguida a quaisquer transformadas para produzir coeficientes de transformada, o codificador de vídeo 20 pode efetuar a quantificação dos coeficientes de transformada. A quantificação refere-se geralmente a um processo no qual coeficientes de transformada são quantificados de modo a se reduzir possivelmente a quantidade de dados utilizados para representar os coeficientes, obtendo-se compactação adicional. O processo de quantificação pode reduzir a profundidade de bits associada a alguns ou todos os coeficientes. Por exemplo, um valor de n bits pode ser arredondado para menos até um valor de m durante a quantificação, onde n é maior que m .

[0093] Em seguida à quantificação, o codificador de vídeo pode varrer os coeficientes de transformada, produzindo um vetor unidimensional a partir da matriz bidimensional que inclui os coeficientes de transformada quantificados. A varredura pode ser projetada para colocar coeficientes de energia mais elevada (e, portanto, de frequência mais baixa) na frente do arranjo e para colocar coeficientes de energia mais baixa (e, portanto, de frequência mais elevada) na parte posterior do arranjo. Em alguns exemplos, o codificador de vídeo 20 pode utilizar uma ordem de varredura pré-definida para varrer os coeficientes de transformada quantificados de modo a produzir um vetor serializado que pode ser codificado por entropia. Em outro exemplo, o codificador de vídeo 20 pode efetuar uma varredura adaptativa. Depois de varrer os coeficientes de transformada quantificados para formar um vetor unidimensional o codificador de vídeo 20 pode codificar por entropia o vetor unidimensional, como, por exemplo, de acordo com de acordo com a codificação de

comprimento variável adaptativa ao contexto (CAVLC), a codificação aritmética binária adaptativa ao contexto (CABAC), a codificação aritmética binária adaptativa ao contexto baseada em sintaxe (SEBAC), a codificação por Entropia com Particionamento de Intervalos de Probabilidade (PIPE) ou outra metodologia de codificação por entropia. O codificador de vídeo 20 pode codificar por entropia também os elementos de sintaxe associados aos dados de vídeo codificados para utilização pelo decodificador de vídeo 30 na decodificação dos dados de vídeo.

[0094] Para efetuar CABAC, o codificador de vídeo 20 pode atribuir um contexto dentro de um modelo de contexto a um símbolo a ser transmitido. O contexto pode referir-se, por exemplo, a se os valores vizinhos do símbolo são não zero ou não. Para efetuar CAVLC, o codificador de vídeo 20 pode selecionar um código de comprimento variável para um símbolo a ser transmitido. Palavras de código na VLC podem ser construídas de modo que códigos relativamente mais curtos correspondam a símbolos mais prováveis, enquanto códigos mais compridos correspondem a símbolos menos prováveis. Desta maneira, a utilização da VLC pode levar a uma economia de bits em comparação, por exemplo, com a utilização de palavras de código de igual comprimento para cada símbolo a ser transmitido. A determinação de probabilidade pode ser baseada no contexto atribuído ao símbolo.

[0095] Conforme descrito acima, uma sequência de vídeo pode ser codificada de acordo com uma estrutura de codificação de vídeo determinada, onde a estrutura de codificação define a alocação de tipos de imagem (imagens RAP e não RAP, por exemplo) utilizado para codificar uma sequência de vídeo. Por exemplo, uma sequência de vídeo pode ser codificada com imagens RAP incluídas a intervalos

pré-determinados de modo a se facilitar o acesso aleatório de uma sequência de vídeo. Tal estrutura de codificação pode ser útil para aplicativos de broadcast. Além disto, uma sequência de vídeo pode ser codificada de acordo com uma estrutura de codificação que reduz ao mínimo o retardo para aplicativos de baixo retardo. A unidade de estrutura de codificação 19 pode ser configurada para determinar uma estrutura de codificação a ser utilizada pelo codificador de vídeo 20 para codificar uma sequência de vídeo recebida da fonte de vídeo 18. Em um exemplo, a unidade de estrutura de codificação 19 pode armazenar estruturas de codificação pré-definidas que correspondem a respectivos aplicativos de vídeo. A unidade de estrutura de codificação 19 pode ser configurada para transmitir informações que indicam uma estrutura de codificação específica para cada um do codificador de vídeo 20 e da unidade de encapsulamento 21. O codificador de vídeo 20 recebe uma sequência de vídeo da fonte de vídeo 18 e informações de estrutura de codificação da unidade de estrutura de codificação e gera dados de vídeo codificados. A unidade de encapsulamento 21 recebe dados de vídeo codificados do codificador de vídeo 20 e informações que indicam uma estrutura de codificação específica e gera uma sequência de vídeo codificada que inclui unidades de acesso. A unidade de desencapsulamento 29 pode ser configurada para receber a sequência de vídeo codificada e efetuar parse nas unidades de acesso e nas unidades NAL. O decodificador de vídeo 30 pode ser configurado para receber unidades NAL e reconstruir dados de vídeo com base nas informações incluídas nas unidades NAL recebidas.

[0096] Deve-se observar que a unidade de estrutura de codificação e/ou codificador de vídeo 20 podem ser configurados para gerar elementos de sintaxe incluídos em

um conjunto de parâmetros. Em alguns exemplos, a unidade de estrutura de codificação 19 pode ser configurada para gerar elementos de sintaxe incluídos em conjunto de parâmetros de alto nível, tais como um SPS, e o codificador de vídeo 20 pode ser configurado para efetuar codificação de vídeo com base nos elementos de sintaxe recebidos da unidade de estrutura de codificação, assim como elementos de sintaxe codificados por entropia de saída como parte dos dados de vídeo codificados.

[0097] De acordo com as técnicas desta revelação, a alocação de tipos de unidades NAL pode ser feita de maneira que um aparelho, tal como o aparelho de destino 14, possa identificar facilmente uma imagem RAP e informações de temporização conexas. Em um exemplo, imagens IDR sem imagens dianteiras conexas têm um tipo de unidades NAL distinto de imagens IDR que podem ter imagens dianteiras conexas. Por exemplo, imagens IDR sem imagens dianteiras conexas têm um tipo de unidade NAL M, enquanto imagens IDR que podem ter imagens dianteiras conexas têm um tipo de unidade NAL N, onde M não é igual a N, conforme mostrado na Tabela 4. Deve-se observar que, no exemplo mostrado na Tabela 4, as imagens dianteiras associadas a uma imagem IDR podem ser imagens DLP. Em um exemplo, os tipos de unidades NAL mostrados na Tabela 4 podem ser incorporados aos códigos de tipo de unidade NAL do WD7 HEVC e às classes de tipo de unidade NAL mostradas na Tabela 2. Por exemplo, os valores de tipo de unidade NAL invertidos na Tabela 2 podem ser utilizados para os tipos de unidades NAL M e N na Tabela 4.

M	Fatia codificada de uma imagem IDR sem imagens dianteiras conexas slice_layer_rbsp()	VCL
N	Fatia codificada de uma imagem IDR que pode ter imagens dianteiras conexas	VCL

	slice_layer_rbsp()	
--	--------------------	--

TABELA 4: Tipos de unidade NAL IDR distintos

[0098] Em outro exemplo, as imagens CRA sem imagens dianteiras conexas têm um tipo de unidade NAL distinto diferente das imagens CRA que podem ter imagens dianteiras conexas. Além disto, as imagens CRA sem imagens TFD conexas têm uma unidade NAL distinta diferente das imagens CRA que podem ter imagens TFD conexas. Por conseguinte, três tipos de unidade NAL diferentes podem ser utilizados para tipos diferentes de imagens CRA, conforme mostrado na Tabela 5. Em um exemplo, os tipos de unidades NAL mostrados na Tabela 5 podem ser incorporados aos códigos de tipos de unidade NAL do WD7 HEVC e as classes de tipo de unidade NAL mostradas na Tabela 2. Por exemplo, os valores de tipo de unidade NAL invertidos na Tabela 1 podem ser utilizados para os tipos de unidade NAL X, Y e Z na Tabela 5

X	Fatia codificada de uma imagem CRA sem imagens dianteiras conexas slice_layer_rbsp()	VCL
Y	Fatia codificada de uma imagem CRA que não tem TFD associadas (mas pode ter DLP conexas) slice_layer_rbsp()	VCL
Z	Fatia codificada de uma imagem CRA que pode ter imagens TFD conexas slice_layer_rbsp()	VCL

TABELA 5: Tipos de unidade NAL CRA distintos

[0099] Em outro exemplo, as imagens BLA sem imagens dianteiras conexas podem ter um tipo de unidade NAL distinto diferente das imagens BLA que podem ter imagens dianteiras conexas. Além disto, as imagens BLA sem imagens TFD conexas podem ter uma unidade NAL distinta diferente das imagens BLA que podem ter imagens TFD conexas. Por conseguinte, três tipos de unidades NAL diferentes podem

ser utilizados para tipos diferentes de BLA, conforme mostrado na Tabela 6. Em um exemplo, os tipos de unidade NAL mostrados na Tabela 6 podem ser incorporados aos códigos de tipo de unidade NAL do WD7 HEVC e às classes de tipo de unidade NAL mostradas na Tabela 2. Por exemplo, os valores de tipo de unidade NAL invertidos na Tabela 2 podem ser utilizados para os tipos de unidade NAL A, B e C na Tabela 6.

A	Fatia codificada de uma imagem BLA sem imagens dianteiras conexas slice layer rbsp()	VCL
B	Fatia codificada de uma imagem BLA que não tem imagens TFD conexas (mas pode ter imagens DLP conexas) slice layer rbsp()	VCL
C	Fatia codificada de uma imagem BLA que pode ter imagens TFD conexas slice layer rbsp()	VCL

TABELA 6: Tipos de unidade NAL BLA distintos

[0100] Qualquer um e todas as combinações dos tipos de unidade NAL descritos com relação às Tabelas 4-6 podem ser utilizadas para a alocação de tipos de unidade NAL. Em um exemplo, todos os tipos de unidade NAL descritos com relação às Tabelas 4-6 podem ser utilizados para a alocação de tipos de unidade NAL. A Tabela 7 mostra um exemplo onde todos os tipos de unidade NAL mostrados nas Tabelas 4-6 são utilizados para a alocação de tipos de unidade NAL. Conforme mostrado na Tabela 7, os tipos de unidade NAL incluem tipos de unidade NAL de imagem CRA, imagem BLA e imagem IDR descritos com relação às Tabelas 4-6 assim como os tipos de unidade NAL VPS, SPS, PPS e APS descritos acima. A Tabela 7 pode ser contrastada com a Tabela 2 acima no sentido de que a alocação de tipos de unidade NAL apresentados na Tabela 7 inclui vários tipos de unidade NAL para imagens IDR, CRA e BLA, ao passo que a alocação de tipos de unidade NAL apresentados na Tabela 1 inclui um

único tipo de unidade NAL para cada uma das imagens IDR, CRA e BLA.

nal_unit_type	Conteúdo de unidade NAL e estrutura de sintaxe RPSP	Classe de tipo de unidade NAL
0	Não especificado	não VCL
1	Fatia codificada de uma imagem não RAP, não TFD, não DLP e não TLA slice layer rbsp()	VCL
2	Fatia codificada de uma imagem TLA slice layer rbsp()	VCL
3	Fatia codificada de uma imagem TFD slice layer rbsp()	VCL
4	Fatia codificada de uma imagem DLP slice layer rbsp()	VCL
5	Fatia codificada de uma imagem CRA sem imagens dianteiras conexas slice layer rbsp()	VCL
6	Fatia codificada de uma imagem CRA que não tem imagens TFD conexas (mas pode ter imagens DLP conexas) slice layer rbsp()	VCL
7	Fatia codificada de uma imagem CRA que pode ter imagens TFD conexas slice layer rbsp()	VCL
8	Fatia codificada de uma imagem BLA sem imagens dianteiras conexas	VCL
48... 63	Não especificado	não VCL

TABELA 7: Códigos de tipos de unidade NAL e classes de tipo de unidades NAL

[0101] A unidade de encapsulamento pode ser configurada para receber dados de vídeo codificados do codificador de vídeo 20 e informações que indicam um espaço de codificação específica e gerar uma sequência de vídeo codificada que inclui unidades de acesso com base na alocação do tipos de unidades NAL mostrados em qualquer uma e todas as combinações de alocações de unidades NAL mostradas nas Tabelas 2-7. Além disto, a unidade de desencapsulamento 29 pode ser configurada para receber a sequência de vídeo codificada e efetuar parse nas unidades de acesso e nas

unidades NAL, onde as unidades NAL são alocadas com base em qualquer uma e todas as combinações de alocações de unidades NAL mostradas nas Tabelas 2-7.

[0102] Conforme descrito acima, de acordo com WD7 HEVC, de modo que a mensagem SEI de temporização da imagem atual e o mecanismo para permitir a remoção tanto ao nível de CPB de HRD tanto ao nível de AU quanto ao nível de DU ao mesmo tempo obtenham um retardo de sub-imagem de USs devem ser enviadas antes que AU inteira seja codificada, e as mensagens SEI ao nível de AU não podem ser ainda enviadas antes que a AU inteira seja codificada. De acordo com as técnicas desta revelação a unidade de encapsulamento e a unidade de desencapsulamento podem ser configuradas de modo que o comportamento do HRD ao nível de sub-imagem ou ao nível de unidade de decodificação comparado com o WD7 HEVC.

[0103] Por exemplo, a unidade de encapsulamento 21 pode ser configurada de modo que as mensagens SEI ao nível de AU sejam enviadas depois que a AU inteira seja codificada. Tal mensagem SEI ao nível de AU pode ser incluída em uma unidade NAL SEI com um tipo de unidade NAL distinto. Uma diferença entre tal unidade NAL SEI e as definições existentes de unidades NAL SEI, definidas no WD7 HEVC, por exemplo, é que este tipo de unidade NAL SEI distinto pode ser autorizado a suceder a última unidade NAL VCL na mesma AU em ordem de decodificação e pode ser restringido de modo que não preceda a primeira unidade NAL VCL na mesma AU em ordem de decodificação. Unidades NAL SEI e mensagens SEI convencionais podem ser respectivamente referidas como unidades NAL SEI de prefixo e mensagens SEI de prefixo enquanto a unidade NAL SEI e a mensagem SEI distintas aqui descritas podem ser respectivamente referidas como unidade NAL SEI de sufixo e mensagem SEI de sufixo.

[0104] Além de ser configurada para gerar uma sequência de vídeo codificada com base em qualquer uma e todas as combinações de alocações de unidades NAL mostradas nas Tabelas 2-7, a unidade de encapsulamento 21 pode ser configurada para gerar uma sequência de vídeo codificada que inclui unidades NAL SEI de prefixo e sufixo. Da mesma maneira, a unidade de desencapsulamento 29 pode ser configurada para receber uma sequência de vídeo codificada e efetuar parse nas unidades de acesso e nas unidades NAL onde as unidades NAL incluem tipos de unidades NAL SEI de prefixo e sufixo. Ou seja, a unidade de desencapsulamento 29 pode ser configurada para extrair as unidades NAL SEI de sufixo das unidades de acesso. A Tabela 8 mostra um exemplo no qual todos os tipos de NAL mostrados nas Tabelas 4-6 são utilizados na alocação de tipos de unidades NAL, assim como de unidades NAL SEI de prefixo e sufixo.

nal_unit_type	Conteúdo de unidade NAL e estrutura de sintaxe RPSP	Classe de tipos de unidade NAL
0	Não especificado	Não VCL
1	Fatia codificada de uma imagem não RAP, não TFD, não DLP e não TLA slice_layer_rbsp()	VCL
2	Fatia codificada de uma imagem TLA slice_layer_rbsp()	VCL
3	Fatia codificada de uma imagem TFD slice_layer_rbsp()	VCL
4	Fatia codificada de uma imagem DLP slice_layer_rbsp()	VCL
5	Fatia codificada de uma imagem CRA sem imagens dianteiras conexas slice_layer_rbsp()	VCL
6	Fatia codificada de uma imagem CRA que não tem imagens TFD conexas (mas pode ter imagens	VCL

	DLP conexas) slice_layer_rbsp()	
7	Fatia codificada de uma imagem CRA que pode ter imagens TFD conexas slice_layer_rbsp()	VCL
8	Fatia codificada de uma imagem DLA sem imagens dianteiras conexas slice_layer_rbsp()	VCL
9	Fatia codificada de uma imagem BLA que não tem imagens TFD conexas (mas pode ter imagens DLP conexas) slice_layer_rbsp()	VCL
10	Fatia codificada de uma imagem BLA que pode ter imagens TFD conexas slice_layer_rbsp()	VCL
11	Fatia codificada de uma imagem IDR sem imagens dianteiras conexas slice_layer_rbsp()	VCL
12	Fatia codificada de uma imagem IDR que pode ter imagens dianteiras conexas slice_layer_rbsp()	VCL
13... 24	Reservado.	n/a
25	Conjunto de parâmetros de vídeo video_parameter_set_rbsp()	Não VCL
26	Conjunto de parâmetros de sequência seq_parameter_set_rbsp()	Não VCL
27	Conjunto de parâmetros de imagem pic_parameter_set_rbsp()	Não VCL
28	Conjunto de parâmetros de adaptação aps_rbsp()	Não VCL
29	Delimitador de unidades de acesso access_unit_delimiter_rbsp()	Não VCL
30	Dados de enchimento filler_data_rbsp()	Não VCL
31	Informações de aperfeiçoamento suplementares de prefixo (SEI) sei_rbsp()	Não VCL
32	Informações de aperfeiçoamento suplementares (SEI) de sufixo	Não VCL

	sei_rbsp()	
33... 47	Reservado.	n/a
48... 63	Não especificado.	Não VCL

TABELA 8: Códigos de Tipo de Unidade NAL e classes de Tipo de Unidade NAL

[0105] Conforme descrito acima, além de unidades NAL SEI, os tipos de unidade NAL não VCL incluem unidades NAL, VPS, SPS, PPS e APS. De acordo com os tipos de conjunto de parâmetros definidos no WD7 HEVC cada SPS refere-se a um ID de VPS, cada PPS refere-se a um ID de SPS, e cada cabeçalho de fatia refere-se a um ID de PPS e possivelmente um ID de APS. O codificador de vídeo 20 e a unidade de estrutura de codificação 19 podem ser configurados para gerar conjuntos de parâmetros de acordo com os conjuntos de parâmetros definidos no WD7 HEVC. Além disto, o codificador de vídeo 20 e a unidade de estrutura de codificação 19 podem ser configurados para gerar conjuntos de parâmetros em que o ID de VPS e o ID de SPS (com o ID de VPS precedendo o ID de SPS, por exemplo) podem ser opcionalmente sinalizados nos cabeçalhos de fatia. Em um exemplo no qual o ID de VPS e o ID de SPS são sinalizados em um cabeçalho de fatia, nenhum ID de VPS será localizado no SPS e nenhum ID de SPS será localizado no PPS. Além disto, em um exemplo o ID de VPS e o ID de SPS podem estar presentes nos cabeçalhos de fatia de cada imagem RAP e cada imagem pode estar associada a uma mensagem SEI de ponto de recuperação. Além disto, em outros exemplos, o ID de VPS e o ID de SPS podem estar presentes no cabeçalho de fatia para outras imagens.

[0106] A Figura 4 é um diagrama de blocos que mostra uma unidade de encapsulamento exemplar que pode implementar as técnicas descritas nesta revelação. No exemplo mostrado na Figura 4, a unidade de encapsulamento 21 inclui um construtor de unidades NAL VCL 402, um construtor de unidades NAL não VCL 404, um construtor de unidades de

acesso 406 e uma interface de saída de fluxos de bits 408. A unidade de encapsulamento 21 recebe dados de vídeo codificados e sintaxe de alto nível e transmite um fluxo de bits de vídeo codificado. Os dados de vídeo codificados podem incluir dados de vídeo residuais e dados de sintaxe associados a uma fatia. Os dados de sintaxe de alto nível podem incluir, por exemplo, elementos de sintaxe incluídos em um conjunto de parâmetros, mensagens SEI ou outros elementos de sintaxe definidos por um padrão de codificação de vídeo, tal como o padrão HEVC vindouro. Um fluxo de bits de vídeo codificado pode incluir uma ou mais sequências de vídeo codificadas e pode conformar-se geralmente a um padrão de codificação de vídeo, tal como o padrão HEVC vindouro. Conforme descrito acima, as unidades NAL DCL incluem uma fatia de dados de vídeo. O construtor de unidades NAL VCL 402 pode ser configurado para receber fatias de dados de vídeo codificados e gerar unidades NAL VCL com base no tipo de imagem que inclui uma fatia. O construtor de unidades NAL VCL 402 pode ser configurado para gerar unidades NAL VCL de acordo com qualquer uma e todas as combinações das alocações de NAL descritas acima com relação às Tabelas 2-8. O construtor de unidades NAL VCL 402 pode ser configurado para incluir um cabeçalho na unidade NAL VCL onde o cabeçalho identifica o tipo de unidade NAL VCL.

[0107] Por exemplo, o construtor de unidades NAL VCL 402 pode ser configurado para receber uma fatia de dados de vídeo incluída em uma imagem IDR e (1) se a imagem IDR não tiver imagens dianteiras conexas, encapsular a fatia de dados de vídeo em uma unidade NAL com um tipo que indica que a imagem IDR não tem imagens dianteiras ou (2) se a imagem IDR tiver imagens dianteiras conexas, encapsular a fatia de dados de vídeo em uma unidade NAL com um tipo que

indica que a imagem IDR tem imagens dianteiras. O construtor de unidades NAL VCL 402 pode ser configurado para receber uma fatia de dados de vídeo incluída em uma imagem CRA e (1) se a imagem CRA não tiver imagens dianteiras conexas, encapsular a fatia de dados de vídeo em uma unidade NAL com um tipo que indica se a imagem CRA tem imagens dianteiras. Além disto, se as imagens dianteiras associadas à imagem CRA forem imagens TFD, o construtor de unidades NAL VCL 402 pode ser configurado para encapsular uma fatia de dados de vídeo em uma unidade NAL com um tipo que indica que as imagens dianteiras associadas à imagem CRA são TFD.

[0108] Além disso, se as imagens dianteiras associadas à imagem CRA não forem imagens TFD, o construtor de unidades NAL VCL 402 pode ser configurado para encapsular a fatia de dados de vídeo em uma unidade NAL com um tipo que indica que as imagens dianteiras associadas à imagem CRA não são TFD. Além disto, o construtor de unidades NAL VCL 402 pode ser configurado para receber uma fatia de dados de vídeo incluída em uma imagem BLA e (1) se a imagem BLA não tiver imagens dianteiras conexas, encapsular a fatia de dados de vídeo em uma unidade NAL com um tipo que indica que a imagem BLA não tem imagens dianteiras, ou (2) se a imagem BLA tiver imagens dianteiras conexas, encapsular a fatia de dados de vídeo em uma unidade NAL com um tipo que indica que a imagem BLA tem imagens dianteiras. Além disto, se as imagens dianteiras associadas à imagem BLA forem imagens TFD, o construtor de unidades NAL 402 pode ser configurado para encapsular uma fatia de dados de vídeo em uma unidade NAL com um tipo que indica que as imagens dianteiras associadas à imagem BLA são TFD. Além disto, se as imagens dianteiras associadas à imagem BLA não forem imagens TFD, o construtor de unidades NAL VCL pode ser configurado para

encapsular a fatia de dados de vídeo em uma unidade NAL com um tipo que indica que as imagens dianteiras associadas à imagem BLA não são TFD.

[0109] A Figura 5 é um fluxograma que mostra um exemplo de geração de unidades NAL VCL de acordo com as técnicas desta revelação. Embora o exemplo de geração de unidades NAL VCL mostradas na Figura 5 seja descrito como sendo executado pelo construtor de unidades NAL 402, qualquer combinação de aparelho de origem 12, codificador de vídeo 20, unidade de encapsulamento 21 e combinações de componentes deles podem executar o exemplo de geração de unidades NAL VCL mostradas na Figura 5. Conforme mostrado na Figura 5, o construtor de unidades NAL VCL recebe uma fatia de dados de vídeo (502). A fatia de dados de vídeo pode consistir em dados de vídeo codificados de acordo com qualquer uma das técnicas de codificação aqui descritas. A fatia de dados de vídeo pode ser incluída em um dos tipos de imagem aqui descritos. O construtor de unidades NAL VCL 402 determina se a fatia de dados de vídeo é incluída em uma imagem IDR ou CRA (504).

[0110] Se a fatia de dados de vídeo for incluída em uma imagem IDR (ramificação "IDR" de 504), o construtor de unidades NAL VCL 402 determina se a imagem IDR tem imagens dianteiras conexas (506). Se a imagem IDR não tiver imagens dianteiras conexas (ramificação "NÃO" de 506), o construtor de unidades NAL VCL 402 gera uma unidade NAL VCL que indica que a imagem IDR não tem imagens dianteiras conexas (508). Se a imagem IDR tiver imagens dianteiras conexas (ramificação "SIM" de 506), o construtor de unidades NAL VCL 402 gera uma unidade NAL VCL que indica que a imagem IDR tem imagens dianteiras conexas (510).

[0111] Se a fatia de dados de vídeo for incluída em uma imagem CRA, o construtor de unidades NAL VCL 402 determina

se a imagem CRA tem imagens dianteiras conexas (512). Se a imagem CRA não tiver imagens dianteiras conexas (ramificação "NÃO" de 512), o construtor de unidades NAL VCL gera uma unidade NAL VCL que indica que a imagem CRA não tem imagens dianteiras conexas (514). Se a imagem CRA tiver imagens dianteiras conexas (ramificação "SIM" de 512) o construtor de unidades NAL VCL 402 determina se as imagens dianteiras conexas são imagens TFD (516).

[0112] Se as imagens dianteiras conexas da imagem CRA forem imagens TFD (ramificação "SIM" de 516) o construtor de unidades NAL VCL 402 gera uma unidade NAL VCL que indica que as imagens dianteiras conexas das imagens CRA são imagens TFD (518). Se as imagens dianteiras conexas da imagem BLA não forem imagens TFD (ramificação "NÃO" de 516), o construtor de unidades NAL VCL 402 gera uma unidade NAL VCL que indica que as imagens dianteiras conexas não são imagens TFD (520).

[0113] O construtor de unidades NAL VCL 402 pode gerar unidades NAL encapsulando dados em fatia em uma unidade NAL e incluindo um valor de tipo de unidade NAL em um cabeçalho de unidade NAL. Cada valor de tipo de unidade NAL pode corresponder a um respectivo tipo de unidade NAL. Em um exemplo, os valores de tipo de unidade NAL podem ser definidos de acordo com a Tabela 7. As unidades NAL geradas podem ser transmitidas pelo construtor de unidades NAL VCL 402 para o construtor de unidades de acesso 406 para inclusão em uma unidade de acesso (522).

[0114] Desta maneira, a unidade de encapsulamento 21 representa um exemplo de aparelho para gerar um fluxo de bits que inclui dados de vídeo, o aparelho incluindo um processador configurado para determinar se uma imagem de ponto de acesso aleatório (RAP) é de um tipo que pode ter imagens dianteiras conexas e se a imagem RAP compreende uma

imagem de renovação de decodificador instantânea (IDR) ou uma imagem de acesso aleatório limpa (CRA) encapsular uma fatia da imagem RAP em uma unidade de camada de abstração de rede (NAL), em que a unidade NAL inclui um valor de tipo de unidade NAL que indica se a imagem RAP é de um tipo que pode ter imagens dianteiras conexas, gerar um fluxo de bits que inclui a unidade NAL.

[0115] Da mesma maneira, o método da Figura 5 representa um exemplo de método para gerar um fluxo de bits que inclui dados de vídeo, o método incluindo determinar se uma imagem de ponto de acesso aleatório (RAP) é de um tipo que pode ter imagens dianteiras conexas e se a imagem RAP compreende uma imagem de renovação de decodificador instantânea (IDR) ou uma imagem de acesso aleatório limpa (CRA), encapsular uma fatia da imagem RAP em uma unidade de camada de abstração de rede (NAL), em que a unidade NAL inclui um valor de tipo de unidade NAL que indica se a imagem RAP é de um tipo que pode ter imagens dianteiras conexas e gerar um fluxo de bits que inclui a unidade NAL.

[0116] Novamente com referência à Figura 4, o construtor de unidades NAL não VCL 404 pode ser configurado para receber elementos de sintaxe de alto nível, tais como elementos de sintaxe incluídos nos conjuntos de parâmetros e em mensagens SEI, conforme descrito acima, e gerar unidade NAL não VCL com base em qualquer uma e em todas as combinações das alocações de unidades NAL descritas acima com relação às Tabelas 2-8. O construtor de unidades NAL não VCL 404 pode ser configurado para gerar unidades NAL não VCL encapsulando dados de sintaxe em uma unidade NAL e incluindo um valor de tipo de unidade NAL em um cabeçalho de unidade NAL. Por exemplo, o construtor de NAL não VCL pode ser configurado para receber elementos de sintaxe incluídos em um conjunto de parâmetros e incluir um valor

de tipo de unidade NAL que indica o tipo de conjunto de parâmetros no cabeçalho de unidade NAL.

[0117] Além disso, o construtor de unidades NAL não VCL 404 pode ser configurado para receber mensagens SEI ao nível de AU e gerar unidade NAL de mensagem SEI. Em um exemplo, o construtor de unidades NAL não VCL 404 pode ser configurado para gerar dois tipos de unidades NAL de mensagem SEI, onde um primeiro de tipo de unidade NAL SEI indica que tal unidade NAL SEI pode suceder a última unidade NAL VCL em uma unidade de acesso em ordem de decodificação e o segundo tipo de unidade NAL SEI indica que tal unidade NAL SEI pode não suceder a última unidade NAL VCL em uma unidade de acesso em ordem de decodificação. Além disto, o primeiro tipo de unidade NAL SEI pode ser restringido de modo que não seja autorizado a preceder a primeira unidade NAL VCL na mesma unidade de acesso em ordem de decodificação. O primeiro tipo de unidade NAL pode ser referido como unidade NAL SEI de sufixo e o segundo tipo de unidade NAL pode ser referido como unidade NAL SEI de prefixo. O construtor de unidades NAL não VCL 404 transmite unidades NAL não VCL para o construtor de unidades de acesso 406.

[0118] O construtor de unidades de acesso 406 pode ser configurado para receber unidades NAL VCL e unidades NAL não VCL e gerar unidades de acesso. O construtor de unidades de acesso 406 pode receber qualquer tipo de unidade NAL definido nas Tabelas 2-8. O construtor de unidades de acesso VCL 406 pode ser configurado para gerar unidades de acesso com base em qualquer um e em todas as combinações de tipos de unidade NAL aqui descritas. Conforme descrito acima, de acordo com o WD7 HEVC, uma unidade de acesso é um conjunto de unidades NAL que são consecutivas e em ordem de decodificação e contêm uma

imagem codificada. Assim, o construtor de unidades de acesso 406 pode ser configurado para receber uma série de unidades NAL e dispor a série de unidades NAL de acordo com uma ordem de decodificação. Além disto, o construtor de unidades de acesso 406 pode ser configurado para dispor uma unidade NAL SEI de sufixo, conforme descrito acima, de modo que ela suceda a última unidade NAL VCL em uma unidade de acesso e/ou não preceda a primeira unidade NAL VCL na mesma unidade de acesso.

[0119] A Figura 6 é um fluxograma que mostra um exemplo de geração de unidades NAL não VCL de acordo com as técnicas desta revelação. Embora o exemplo de geração de unidades NAL não VCL mostradas na Figura 6 seja descrito como sendo executado pelo construtor de unidades NAL não VCL 404 e pelo construtor de unidades de acesso 406, qualquer combinação do aparelho de origem 12 do codificador de vídeo 20, da unidade de encapsulamento 21 e combinações de componentes deles podem executar o exemplo de geração de unidades NAL não VCL mostrado na Figura 6.

[0120] Conforme mostrado na Figura 6, o construtor de unidades NAL não VCL 404 recebe uma mensagem SEI (602). A mensagem SEI pode ser qualquer tipo de mensagens SEI descrito acima com relação à Tabela 1. O construtor de unidades NAL não VCL 404 determina se a mensagem SEI é uma mensagem SEI de prefixo ou uma mensagem SEI de sufixo (604).

[0121] Se a mensagem SEI for uma mensagem SEI de sufixo (ramificação "SUFIXO" de 604) o construtor de unidades NAL não VCL 404 gera um valor de tipo para uma unidade NAL SEI que indica que a unidade NAL SEI é uma mensagem SEI de sufixo (606). Se a mensagem SEI for uma mensagem SEI de prefixo (ramificação "PREFIXO" de 604), o construtor de unidades NAL não VCL 404 gera um valor de tipo para uma

unidade NAL SEI que indica que a unidade NAL SEI é uma mensagem SEI convencional (608).

[0122] O construtor de unidades de acesso 406 recebe as unidades NAL geradas, que podem incluir qualquer combinação dos tipos de unidades NAL descritos acima com relação às Tabelas 2-8 (610). O construtor de unidades de acesso 406 gera unidades de acesso incluídas nas unidades NAL recebidas (612). Se a unidade de acesso gerada incluir uma unidade NAL SEI de sufixo, as unidades NAL da unidade de acesso podem ser dispostas de modo que a NAL SEI de sufixo não preceda a primeira unidade NAL VCL na mesma unidade de acesso, mas possa suceder a última unidade NAL VCL em uma unidade de acesso em ordem de decodificação.

[0123] Desta maneira, a unidade de encapsulamento 21 representa um exemplo de processador configurado para determinar se uma mensagem de informações de aperfeiçoamento suplementares (SEI) é uma mensagem SEI de prefixo ou uma mensagem SEI de sufixo, em que a mensagem SEI inclui dados relacionados com dados de vídeo codificados, encapsular a mensagem SEI em uma unidade NAL SEI, em que a unidade NAL SEI inclui um valor de tipo de unidade NAL que indica se a unidade NAL SEI é uma unidade NAL SEI de prefixo ou uma unidade NAL SEI de sufixo e se a mensagem SEI é uma mensagem SEI de prefixo ou uma mensagem SEI de sufixo, e gerar um fluxo de bits que inclui pelo menos a unidade NAL SEI.

[0124] Da mesma maneira, o método da Figura 6 representa um exemplo de método para gerar um fluxo de bits que inclui dados de vídeo, o método incluindo determinar se uma mensagem de informações de aperfeiçoamento suplementares (SEI) é uma mensagem SEI de prefixo ou uma mensagem SEI de sufixo, em que a mensagem SEI inclui dados relacionados com dados de vídeo codificados, encapsular a mensagem SEI em

uma unidade NAL SEI, em que a unidade NAL SEI inclui um valor de tipo de unidade NAL que indica se a unidade NAL SEI é uma unidade NAL SEI de prefixo ou uma unidade NAL SEI de sufixo e se a mensagem SEI é uma mensagem SEI de prefixo ou uma mensagem SEI de sufixo, e gerar um fluxo de bits que inclui pelo menos a unidade NAL SEI.

[0125] Novamente com referência à Figura 4, a interface de saída de fluxos de bits 408 pode ser configurada para receber unidades de acesso e gerar uma sequência de vídeo codificada. A interface de saída de fluxos de bits 408 pode ser também configurada para transmitir uma sequência de vídeo codificada como parte de um fluxo de bits de vídeo codificado, onde um fluxo de bits de vídeo codificado inclui uma ou mais sequências de vídeo codificadas com base em qualquer um e todas as combinações de tipos de unidades NAL aqui descritos. Conforme descrito acima, de acordo com o WD7 HEVC, uma sequência de vídeo codificada é um conjunto de unidades de acesso que são consecutivas em ordem de decodificação. Assim, a interface de saída de fluxos de bits 408 pode ser configurada para receber uma série de unidades de acesso e dispor a série de unidades de acesso de acordo com uma ordem de decodificação.

[0126] Conforme descrito acima, a unidade de estrutura de codificação 19 e/ou o codificador de vídeo 20 podem ser configurados para gerar elementos de sintaxe incluídos em um conjunto de parâmetros que inclui o elemento de sintaxe **fixed_pic_rate_flag** que pode ser incluído em um conjunto de parâmetros VUI, que pode ser incluído em um SPS, conforme apresentado no WD7 HEVC. Além disto, a unidade de estrutura de codificação 19 e/ou o codificador de vídeo 20 podem ser configurados para gerar o elemento de sintaxe **fixed_pic_rate_flag**, onde o elemento de sintaxe **fixed_pic_rate_flag** inclui elementos semânticos que são

modificados a partir dos apresentados no WD7 HEVC. Por exemplo, de acordo com os elementos semânticos atuais de **fixed_pic_rate_flag** no WD7 HEVC quando **fixed_pic_rate_flag** é igual a 1, é necessário que a diferença entre os tempos de apresentação de duas imagens contínuas em ordem de saída seja igual ao pulso de relógio. Entretanto, isto exigiria a alteração do valor ou de **time_scale** ou de **num_units_in_tick** quando algumas camadas temporais mais elevadas são descartadas para adaptação de fluxos com base na capacidade de escalonamento temporal.

[0127] Em um exemplo, em vez de ser necessário que o delta (isto é, a diferença entre os tempos de apresentação entre duas imagens contínuas em ordem de saída) seja exatamente igual ao pulso de relógio pode ser necessário que o delta seja um número inteiro de pulso de relógio. Desta maneira, a unidade de estrutura de codificação 19 e/ou codificador de vídeo 20 podem ser configurados para gerar o elemento de sintaxe **fixed_pic_rate_flag** de modo que, quando **fixed_pic_rate_flag** é igual a 1, seja necessário que a diferença entre os tempos de apresentação de duas imagens contínuas em ordem de saída seja igual a um número inteiro do pulso de relógio.

[0128] Em outro exemplo, pode ser necessário que a unidade de estrutura de codificação 19 e/ou o codificador de vídeo 20 sinalizem um **fixed_pic_rate_flag** para cada camada temporal. Além disto, neste exemplo, se **fixed_pic_rate_flag** para uma camada temporal específica for igual a 1, isto é, a representação de camada temporal tiver uma taxa de imagens constante, um valor N pode ser sinalizado, e o delta (entre os tempos de apresentação de duas imagens contínuas em ordem de saída) para a representação de camada temporal pode ser igual a N pulsos de relógio.

[0129] Em outro exemplo, a unidade de estrutura de codificação 19 e/ou codificador de vídeo 20 podem ser configurados para sinalizar opcionalmente um **fixed_pic_rate_flag** para cada camada temporal. Neste exemplo se **fixed_pic_rate_flag** para uma camada temporal estiver presente e for igual a 1, isto é, a representação de camada temporal tiver uma taxa de imagens constante, um valor N pode ser sinalizado e o delta (entre os tempos de apresentação de duas imagens contínuas em ordem de saída) para a representação de camada temporal é igual a N pulsos de relógio. No caso de **fixed_pic_rate_flag** ser opcionalmente sinalizado para cada camada temporal, supondo-se que o **fixed_pic_rate_flag** seja sinalizado para camada temporal mais elevada e o valor seja igual a 1 então, para cada camada temporal específica que não tenha **fixed_pic_rate_flag** sinalizado, o valor de **fixed_pic_rate_flag** pode ser derivado como sendo igual ao **fixed_pic_rate_flag** sinalizado para camada temporal mais elevada, e o valor de N é derivado como sendo igual a $2^{\text{max_Tid}} - \text{currTid}$, em que max_Tid é igual ao valor de temporal_id mais elevado, e currTid é igual ao temporal_id da camada temporal específica.

[0130] A Figura 7 é um fluxograma que mostra um exemplo de sinalização de um valor delta de tempo de apresentação. Embora o exemplo de sinalização de um valor delta de tempo de apresentação mostrado na Figura 7 seja descrito como sendo executado pela unidade de encapsulamento 21, qualquer combinação do aparelho de origem 12, do codificador de vídeo 20, da unidade de encapsulamento 21 e combinações de componentes deles podem executar o exemplo de sinalização do valor delta de tempo de apresentação mostrado na Figura 7.

[0131] Conforme mostrado no exemplo da Figura 7, a unidade de encapsulamento 21 gera um indicador que indica se o delta entre o tempo de apresentação (um valor de POC, por exemplo) de uma primeira imagem e o tempo de apresentação de uma segunda imagem é um número inteiro de um valor de pulso de relógio (702). Em outras palavras, a unidade de encapsulamento 21 pode gerar dados que indicam se a diferença (o delta, por exemplo) entre os tempos de apresentação de uma primeira imagem e uma segunda imagem é um múltiplo de número inteiro do valor de pulso de relógio. O indicador descrito na Figura 7 representa um exemplo de tais dados gerados. Em alguns casos, a unidade de encapsulamento 21 pode receber um valor para o indicador da unidade de estrutura de codificação 19 ou do codificador de vídeo 20. O indicador pode ser qualquer um dos elementos de sintaxe **fixed_pic_rate_flag** descritos acima.

[0132] Em um exemplo, a unidade de encapsulamento 21 determina se um valor para o indicador pode indicar que o delta é um número inteiro do valor de pulso de relógio (704). Quando o indicador indica que o delta é um valor de número inteiro do pulso de relógio (ramificação "SIM" de 704), a unidade de encapsulamento 21 pode gerar um valor de número inteiro N (706) que representa o múltiplo de número inteiro do valor de pulso de relógio. O valor de número inteiro N pode ser utilizado por um aparelho de decodificação, tal como o aparelho de destino 14, para determinar o valor delta, onde o delta é múltiplo de número inteiro do valor de pulso de relógio. Em um exemplo, o valor de número inteiro N pode ser um valor de zero a 2047 e pode indicar um valor de 1 menos o número inteiro de relógios a qual o delta é igual. A unidade de encapsulamento 21 pode transmitir então o indicador e o

valor de número inteiro N como parte de um fluxo de bits (708).

[0133] Por outro lado, quando a unidade de encapsulamento 21 determina que o indicador indica que o valor delta não é um múltiplo de número inteiro do pulso de relógio (ramificação "NÃO" de 704) a unidade de encapsulamento 21 pode simplesmente transmitir o indicador (710).

[0134] Desta maneira, o aparelho de origem 12 representa um exemplo de processador configurado para gerar dados que indicam se a diferença entre o tempo de apresentação de uma primeira imagem e o tempo de apresentação de uma segunda imagem é um múltiplo de número inteiro de um valor de pulso de relógio, e, quando os dados indicarem que a diferença é o múltiplo de número inteiro do valor de pulso de relógio, gerar dados que representam o múltiplo de número inteiro.

[0135] Da mesma maneira, o método da Figura 7 representa um exemplo de método para gerar um fluxo de bits que inclui dados de vídeo, o método incluindo gerar dados que indicam se a diferença entre o tempo de apresentação de uma primeira imagem e o tempo de apresentação de uma segunda imagem é um múltiplo de número inteiro de um valor de pulso de relógio e, quando os dados indicam que a diferença é o múltiplo de número inteiro do valor de pulso de relógio, gerar dados que representam o múltiplo de número inteiro.

[0136] Conforme descrito acima, a unidade de encapsulamento 21 recebe dados de vídeo codificados. A Figura 8 é um diagrama de blocos que mostra um exemplo de codificador de vídeo 20 que pode gerar dados de vídeo codificados. Conforme mostrado na Figura 8, o codificador de vídeo 20 recebe dados de vídeo e dados de sintaxe de alto nível. O codificador de vídeo funciona tipicamente em blocos de vídeo dentro de fatias de vídeo individuais de

modo a codificar os dados de vídeo. Um bloco de vídeo pode corresponder a um nó de codificação dentro de uma CU. Os blocos de vídeo podem ter tamanhos fixos ou variáveis e podem diferir em tamanho de acordo com o padrão de codificação especificado. O codificador de vídeo 20 pode gerar também dados de sintaxe, tais como dados de sintaxe baseados em blocos, dados de sintaxe baseados em quadros e dados de sintaxe baseados em GOP, como, por exemplo, em um cabeçalho de quadro, um cabeçalho de bloco, um cabeçalho de fatia ou um cabeçalho de GOP. Os dados de sintaxe de GOP podem descrever o número de quadros no respectivo GOP e os dados de sintaxe de quadro podem indicar o modo de codificação/predição utilizado para codificar o quadro correspondente.

[0137] No exemplo da Figura 8, o codificador de vídeo 20 inclui uma unidade de seleção de modo 40, uma memória de imagens de referência 64, um somador 50, uma unidade de processamento de transformadas 52, uma unidade de quantificação 54 e uma unidade de codificação por entropia 56. A unidade de seleção de modo 40, por sua vez, inclui uma unidade de compensação de movimento 44, uma unidade de estimação de movimento 42, uma unidade de intra-predição 46 e uma unidade de partição 48. Para reconstrução de blocos de vídeo, o codificador de vídeo 20 inclui também uma unidade de quantificação inversa 58, uma unidade de transformada inversa 60 e um somador 62. Um filtro de desbloqueio (não mostrado na Figura 8) pode ser também incluído para filtrar fronteiras entre blocos de modo a remover artefatos de blocagem de vídeo reconstruído. Se desejável, o filtro de desbloqueio filtraria tipicamente a saída do somador 62. Filtros adicionais (em malha ou pós-malha) podem ser também utilizados além do filtro de desbloqueio. Tais filtros não são mostrados por razões de

concisão, mas, se desejado, podem filtrar a saída do somador 50 (como um filtro em malha).

[0138] Durante o processo de codificação, o codificador de vídeo 20 recebe um quadro ou fatia de vídeo a ser codificada. O quadro ou fatia pode ser dividida em vários blocos de vídeo. A unidade de estimação de movimento 42 e a unidade de compensação de movimento 44 efetuam codificação inter-preditiva do bloco de vídeo recebido com relação a um ou mais blocos em um ou mais quadros de referência, de modo a se obter predição temporal. A unidade de intra-predição 46 pode efetuar alternativamente codificação intra-preditiva do bloco de vídeo recebido com relação a um ou mais blocos vizinhos no mesmo quadro ou fatia do bloco a ser codificado, de modo a se obter predição espacial. O codificador de vídeo 20 pode executar várias passagens de codificação, como, por exemplo, para selecionar um modo de codificação apropriado para cada bloco de dados de vídeo.

[0139] Além do mais, a unidade de partição 48 pode particionar blocos de dados de vídeo em sub-blocos, com base na avaliação de esquema de particionamento anteriores em passagens de codificação anteriores. Por exemplo, a unidade de partição 48 pode particionar inicialmente um quadro ou fatia em LCUs, e particionar cada uma das LCUs em sub-CUs com base na análise de distorção de taxa (otimização de distorção de taxa, por exemplo). A unidade de seleção de modo 40 pode produzir também uma estrutura de dados de transformação quad-tree que indica o particionamento de uma LCU em sub-CUs. As CUs de nó-folha da transformação quad-tree podem incluir uma ou mais PUs e uma ou mais TUs.

[0140] A unidade de seleção de modo 40 pode selecionar um dos modos de codificação, intra- ou inter- como, por exemplo, com base em resultados de erro e envia o bloco

intra- ou inter-codificado resultante ao somador 50 para gerar dados de bloco residuais e ao somador 62 para reconstruir o bloco codificado para utilização como um quadro de referência. A unidade de seleção de modo 40 também provê elementos de sintaxe, tais como vetores de movimento, indicadores de intra-modo, informações de partição e outras informações de sintaxe que tais, para a unidade de codificação por entropia 56.

[0141] A unidade de estimação de movimento 42 e unidade de compensação de movimento 44 podem ser altamente integradas, mas são mostradas separadamente para fins conceptuais. A estimação de movimento efetuada pela unidade de estimação de movimento 42, é o processo de gerar vetores de movimento, que estimam o movimento para blocos de vídeo. Um vetor de movimento, por exemplo, pode indicar o deslocamento de uma PU de um bloco de vídeo dentro do quadro ou imagem de vídeo atual com relação a um bloco preditivo dentro de um quadro de referência (ou outra unidade codificada) com relação ao bloco atual que é codificado dentro do quadro atual (ou outra unidade codificada). Um bloco preditivo é um bloco que se verifica corresponder intimamente ao bloco a ser codificado, em termos de diferença de pixel, o que pode ser determinado pela soma da diferença absoluta (SAD), pela soma da diferença ao quadrado (SSD) ou outras métricas de diferença. Em alguns exemplos, o codificador de vídeo 20 pode calcular valores para posições de sub-pixel de número inteiro de imagens de referência armazenadas na memória de imagens de referência 64. Por exemplo, o codificador de vídeo 20 pode interpolar valores de posições de um quarto de pixel, posições de um oitavo de pixel ou outras posições de pixel fracionárias da imagem de referência. Portanto, a unidade de estimação de movimento 42 pode efetuar uma busca

de movimento com relação às posições de pixel completas e posições de pixel fracionárias e transmitir um vetor de movimento com precisão de pixel fracionária.

[0142] A unidade de estimação de movimento 42 calcula um vetor de movimento para uma PU de um bloco de vídeo em uma fatia inter-codificada comparando a posição da PU com a posição de um bloco preditivo de uma imagem de referência. A imagem de referência pode ser selecionada a partir de uma primeira lista de imagens de referência (Lista 0) ou de uma segunda lista de imagens de referência (Lista 1), cada uma das quais identifica uma ou mais imagens de referência armazenadas na memória de imagens de referência 64. A unidade de estimação de movimento 42 envia o vetor de movimento calculado à unidade de codificação por entropia 56 e à unidade de compensação de movimento 44.

[0143] A compensação de movimento efetuada pela unidade de compensação de movimento 44, pode envolver buscar ou gerar o bloco preditivo com base no vetor de movimento determinado pela unidade de estimação de movimento 42. Mais uma vez, a unidade de estimação de movimento 42 e a unidade de compensação de movimento 44 podem ser funcionalmente integradas, em alguns exemplos. Ao receber o vetor de movimento para a PU do bloco de vídeo atual, a unidade de compensação de movimento 44 pode localizar o bloco preditivo que o vetor de movimento indica em uma das listas de imagens de referência. O somador 50 forma um bloco de vídeo residual subtraindo valores de pixel do bloco preditivo dos valores de pixel do bloco de vídeo atual que é codificado, formando valores de diferença de diferença de pixel conforme discutido em seguida. Em geral, a unidade de estimação de movimento 42 efetua estimação de movimento com relação a componentes luma e a unidade de compensação de movimento 44 utiliza vetores de movimento calculados com

base nos componentes luma tanto para componentes croma quanto componentes luma. A unidade de seleção de modo 40 pode gerar também elementos de sintaxe associados aos blocos de vídeo e à fatia de vídeo para utilização pelo decodificador de vídeo 30 na decodificação dos blocos de vídeo da fatia de vídeo.

[0144] A unidade de intra-predição 46 pode intra-predizer o bloco atual, como alternativa para a inter-predição efetuada pela unidade de estimação de movimento 42 e pela unidade de compensação de movimento 44, conforme descrito acima. Em particular, a unidade de intra-predição 46 pode determinar um modo de intra-predição a ser utilizado para codificar o bloco atual. Em alguns exemplos, a unidade de intra-predição 46 pode codificar o bloco atual utilizando diversos modos de intra-predição, como, por exemplo, durante passagens de codificação separadas, e a unidade de intra-predição 46 (ou a unidade de seleção de modo 40, em alguns exemplos) pode selecionar o modo de intra-predição apropriado a ser utilizado a partir dos modos testados.

[0145] Por exemplo, a unidade de intra-predição 46 pode calcular valores de distorção de taxa utilizando uma análise de distorção de taxa para os diversos modos de intra-predição testados, e selecionar o modo de intra-predição que tem as melhores características de distorção de taxa entre os modos testados. A análise de distorção de taxa determina geralmente o grau de distorção (ou erro) entre um bloco codificado e um bloco não codificado original que foi codificado de modo a se produzir o bloco codificado, assim como uma taxa de bits (isto é, o número de bits) utilizada para produzir o bloco codificado. A unidade de intra-predição 46 pode calcular razões a partir das distorções e taxas para os diversos blocos codificados

de modo a determinar qual o modo de intra-predição apresenta o melhor valor de distorção de taxa para o bloco. [0146] Depois de selecionar o modo de intra-predição para um bloco, a unidade de intra-predição 46 pode fornecer informações que indicam o modo de intra-predição selecionado para o bloco à unidade de codificação por entropia 56. A unidade de codificação por entropia 56 pode codificar as informações que indicam o modo de intra-predição selecionado. O codificador de vídeo 20 pode incluir no fluxo de bits transmitido dados de configuração, que podem incluir uma série de tabelas de índices de modo de intra-predição e uma série de tabelas de índices de modo de intra-predição modificadas (também referidas como tabelas de mapeamento em palavras de código), definições de contexto de codificação para diversos blocos e indicações do modo de intra-predição mais provável, uma tabela de índices de modos de intra-predição modificada a ser utilizada para cada um dos contextos.

[0147] O codificador de vídeo 20 forma um bloco de vídeo residual subtraindo os dados de predição da unidade de seleção de modo 40 do bloco de vídeo original que é codificado. O somador 50 representa o componente ou componentes que executam esta operação de subtração. A unidade de processamento de transformadas 52 aplica uma transformada, tal como uma transformada de co-seno discreta (DCT) ou uma transformada conceptualmente semelhante ao bloco residual, produzindo um bloco de vídeo que compreende valores de coeficientes de transformada residuais. A unidade de processamento de transformadas 52 pode executar outras transformadas que são conceptualmente semelhantes à DCT, podem ser também utilizadas transformadas de wavelet, transformadas de número inteiro, transformadas de sub-banda ou outros tipos de transformadas. Seja como for, a unidade

de processamento de transformadas 52 aplica a transformada ao bloco residual, produzindo um bloco de coeficientes de transformada residuais. A transformada pode converter as informações residuais de um domínio de valor de pixel em um domínio de transformada, tal como um domínio de frequência. A unidade de processamento de transformadas 52 pode enviar os coeficientes de transformada resultantes à unidade de quantificação 54. A unidade de quantificação 54 quantifica os coeficientes de transformada de modo a reduzir ainda mais a taxa de bits. O processo de quantificação pode reduzir a profundidade de bits associada a alguns ou todos os coeficientes. O grau de quantificação pode ser modificado pelo ajuste de um parâmetro de quantificação. Em alguns exemplos, a unidade de quantificação 54 pode efetuar então uma varredura da matriz que inclui os coeficientes de transformada quantificados. Alternativamente, a unidade de codificação por entropia 56 pode efetuar a varredura.

[0148] Em seguida à quantificação, a unidade de codificação por entropia 56 codifica por entropia os coeficientes de transformada quantificados. Por exemplo, a unidade de codificação por entropia 56 pode efetuar codificação de comprimento variável adaptativa ao contexto (CAVLC), codificação aritmética binária adaptativa ao contexto (CABAC), codificação aritmética binária adaptativa ao contexto baseada em sintaxe (SBAC), codificação por entropia com particionamento de intervalos de probabilidade (PIPE) ou outra técnica de codificação por entropia. No caso da codificação por entropia baseada no contexto, o contexto pode ser baseado em blocos vizinhos. Em seguida à codificação por entropia pela unidade de codificação por entropia 56, o fluxo de bits codificado pode ser transmitido para outro aparelho (o decodificador de vídeo

30, por exemplo) ou arquivado para transmissão ou recuperação posterior.

[0149] A unidade de quantificação inversa 58 e a unidade de transformada inversa 60 aplicam quantificação inversa e transformação inversa, e respectivamente, para reconstruir o bloco residual no domínio de pixel, como, por exemplo, para utilização posterior como bloco de referência. A unidade de compensação de movimento 44 pode calcular um bloco de referência adicionando o bloco residual a um bloco preditivo de um dos quadros da memória de imagens de referência 64. A unidade de compensação de movimento 44 pode aplicar também um ou mais filtros de interpolação ao bloco residual reconstruído de modo a calcular valores de sub-pixel de número inteiro para utilização na estimação de movimento. O somador 62 adiciona o bloco residual reconstruído ao bloco de predição compensado em movimento produzido pela unidade de compensação de movimento 44 de modo a se produzir um bloco de vídeo reconstruído para armazenamento na memória de imagens de referência 64. O bloco de vídeo reconstruído pode ser utilizado pela unidade de estimação de movimento 42 e pela unidade de compensação de movimento 44 como um bloco de referência para inter-codificar um bloco no quadro de vídeo subsequente.

[0150] Conforme descrito acima, a unidade de desencapsulamento 29 pode ser configurada para receber a sequência de vídeo codificada e efetuar parse nas unidades de acesso e nas unidades NAL, onde as unidades NAL são alocadas com base em qualquer uma e em todas as combinações de alocações de unidades NAL mostradas nas Tabelas 2-7. Além disto, a unidade de desencapsulamento 29 e o decodificador de vídeo 30 podem reconstruir dados de vídeo com base nas alocações de tipos de unidades NAL. Em um exemplo, a unidade de desencapsulamento 29 pode ser

configurada para receber uma unidade NAL, em que a unidade NAL inclui um valor de tipo de NAL e determina se a unidade NAL encapsula uma fatia codificada de dados de vídeo incluída em uma imagem RAP associada a uma imagem dianteira com base no valor de tipo de NAL, e o decodificador de vídeo 30 pode ser configurado para reconstruir dados de vídeo com base em se a unidade NAL encapsula uma fatia codificada de dados de vídeo incluída em uma imagem RAP com uma imagem dianteira conexa. Em outro exemplo, a unidade de desencapsulamento 29 pode ser configurada para receber uma unidade NAL, em que a unidade NAL inclui um valor de tipo de NAL, e determinar se a unidade NAL encapsula uma mensagem SEI ao nível de AU com base no valor de tipo de NAL, e o decodificador de vídeo 30 pode ser configurado para reconstruir dados de vídeo com base em se a unidade NAL encapsula uma mensagem SEI ao nível de AU. Em alguns casos, reconstruir dados de vídeo pode incluir gerar um fluxo de bits emendado, conforme descrito acima, e o decodificador de vídeo 30 pode determinar os tempos de apresentação de imagens no fluxo de vídeo emendado com base na determinação do tipo de unidade NAL.

[0151] Também conforme descrito acima, um aparelho de origem, tal como o aparelho de origem 12, pode ser configurado para sinalizar o delta entre o tempo de apresentação de uma primeira imagem e o tempo de apresentação de uma segunda imagem, onde a sinalização utiliza qualquer um dos elementos de sintaxe **fixed_pic_rate_flag** descritos acima. Assim, o aparelho de destino 14, a unidade de desencapsulamento 29 e o decodificador de vídeo 30 podem ser configurados para determinar os tempos de apresentação de uma primeira imagem e uma segunda imagem e apresentar as imagens em conformidade com isto.

[0152] A Figura 9 é um fluxograma que mostra um método exemplar para determinar um valor delta de tempo de apresentação. Embora o exemplo de sinalização de um valor delta de tempo de apresentação mostrado na Figura 9 seja descrito como sendo executado pela unidade de desencapsulamento 29, qualquer combinação do aparelho de destino 14, do decodificador de vídeo 30, da unidade de desencapsulamento 29 e combinações de componentes deles podem executar o exemplo da determinação de um valor delta de tempo de apresentação mostrado na Figura 9. Conforme mostrado na Figura 9, a unidade de desencapsulamento 29 obtém uma primeira imagem (902). A primeira imagem pode ser uma imagem codificada que corresponde a uma unidade de acesso. A unidade de desencapsulamento 29 obtém uma segunda imagem (904). A segunda imagem pode ser uma imagem codificada que corresponde a uma unidade de acesso. A segunda imagem pode ser incluída na mesma camada temporal da primeira imagem. Além disto, as primeira e segunda imagens podem ser incluídas em uma camada temporal mais elevada de dados de vídeo.

[0153] A unidade de desencapsulamento 29 pode obter então um valor de número inteiro N (906). Isto é supor que a unidade de desencapsulamento 29 tinha obtido anteriormente dados, tais como um valor para um indicador, que indica isso. O valor de número inteiro N pode ser incluído em um conjunto de parâmetros VUI, que pode ser incluído em um SPS. A unidade de desencapsulamento 29 determina um valor de pulso de relógio (908). A unidade de desencapsulamento 29 pode determinar o valor de pulso de relógio com base nos elementos de sintaxe **time_scale** e **num_units_in_tick** de acordo com a equação (1) descrita acima.

[0154] A unidade de desencapsulamento 29 pode determinar então um delta entre o tempo de apresentação da primeira imagem e o tempo de apresentação da segunda imagem (910). O delta pode ser igual a um número inteiro do valor de pulso de relógio baseado no valor de número inteiro N. Por exemplo, o delta pode ser igual a pulso de relógio ($N + 1$).

[0155] A unidade de desencapsulamento e o decodificador de vídeo 30 podem apresentar então a primeira imagem e a segunda imagem de acordo com o delta determinado (912). Em um exemplo, a unidade de desencapsulamento 29 pode sinalizar o valor de delta para o decodificador de vídeo 30 e o decodificador de vídeo 30 pode executar um processo de decodificação com base no valor de delta. Desta maneira, o aparelho de destino 14 representa o exemplo de aparelho que inclui um processador configurado para determinar um valor de diferença entre o tempo de apresentação de uma primeira imagem e o tempo de apresentação de uma segunda imagem, em que o valor de diferença é igual a um valor de número inteiro multiplicado por um valor de pulso de relógio e apresentar a primeira imagem e a segunda imagem de acordo com o valor de diferença determinado.

[0156] Da mesma maneira, o método da Figura 9 representa um exemplo de método que inclui determinar um valor de diferença entre o tempo de apresentação de uma primeira imagem e o tempo de apresentação de uma segunda imagem, em que o valor de diferença é igual a um valor de número inteiro multiplicado por um valor de pulso de relógio, e apresentar a primeira imagem e a segunda imagem de acordo com valor de diferença determinado.

[0157] A Figura 10 é um diagrama de blocos que mostra um exemplo de decodificador de vídeo 30 que pode implementar técnicas para (1) receber dados que incluem tipos de

unidades NAL, (2) processar o comportamento de HRD ao nível de sub-imagem ou ao nível de decodificação recebido (3) processar dados que incluem referência a IDs de conjunto de parâmetros, (4) processar dados recebidos que incluem elementos semânticos aperfeiçoados **fixed_pic_rst_flag** ou qualquer uma e todas as combinações deste. No exemplo da Figura 10 o decodificador de vídeo 30 inclui uma unidade de decodificação por entropia 70, uma unidade de compensação de movimento 72, uma unidade de intra-predição 74, uma unidade de quantificação inversa 76, uma unidade de transformação inversa 78, uma memória de imagens de referência 82 e um somador 80. O decodificador de vídeo 30 pode, em alguns exemplos, executar uma passagem de decodificação geralmente correspondente à passagem de codificação descrita com relação ao codificador de vídeo 20 (Figura 2). A unidade de compensação de movimento 72 pode gerar dados de predição com base nos vetor de movimento recebidos da unidade de decodificação por entropia 70, enquanto a unidade de intra-predição 74 pode gerar dados de predição com base nos indicadores de modo de intra-predição recebidos da unidade de decodificação por entropia 70.

[0158] Durante o processo de decodificação o decodificador de vídeo 30 recebe um fluxo de bits de vídeo codificado que representa blocos de vídeo de uma fatia de vídeo codificada e elementos de sintaxe conexos do codificador de vídeo 20. A unidade de decodificação por entropia 70 do decodificador de vídeo 30 decodifica por entropia o fluxo de bits de modo a gerar coeficientes quantificados, vetores de movimento ou indicadores de modo de intra-predição e outros elementos de sintaxe. A unidade de decodificação por entropia 70 emite os vetores de movimento para e outros elementos de sintaxe para a unidade de compensação de movimento 72. O decodificador de vídeo 30

pode receber os elementos de sintaxe ao nível de fatia de vídeo ou ao nível de bloco de vídeo.

[0159] Quando a fatia de vídeo é codificada como uma fatia intra-codificada (I) a unidade de intra-predição 74 pode gerar dados de predição para um bloco de vídeo da fatia de vídeo atual com base no modo de inter-predição sinalizado e em dados de blocos decodificados anteriormente do quadro ou imagem atual. Quando o quadro de vídeo é codificado como uma fatia inter-codificada (isto é, B, P ou GPB) a unidade de compensação de movimento 72 produz blocos preditivos para um bloco de vídeo da fatia de vídeo atual com base nos vetores de movimento e em outros elementos de sintaxe recebidos da unidade de decodificação por entropia 70. Os blocos preditivos podem ser produzidos a partir de uma das imagens de referência dentro de uma das listas de imagens de referência. O decodificador de vídeo 30 pode construir as Listas de quadros de referência, Lista 0 e Lista 1, utilizando técnicas de construção pré-definidas baseadas em imagens de referência armazenadas na memória de imagens de referência 82. A unidade de compensação de movimento 72 determina informações de predição para um bloco de vídeo da fatia de vídeo atual efetuando parse nos vetores de movimento e em outros elementos de sintaxe, e utiliza as informações de predição para produzir os blocos preditivos para o bloco de vídeo atual que é decodificado. Por exemplo, a unidade de compensação de movimento 72 utiliza alguns dos elementos de sintaxe recebidos para determinar um modo de predição (intra- ou inter-predição, por exemplo) utilizado para codificar os blocos de vídeo da fatia de vídeo, um tipo de fatia de inter-predição (fatia B, fatia P ou fatia GPB, por exemplo), informações de construção para uma ou mais das Listas de imagens de referência para a fatia, vetores de movimento para cada

bloco de vídeo inter-codificado da fatia, a condição de inter-predição para cada bloco de vídeo inter-codificado da fatia e outras informações para decodificar os blocos de vídeo na fatia de vídeo atual.

[0160] A unidade de compensação de movimento 72 pode efetuar também interpolação com base em filtros de interpolação. A unidade de compensação de movimento 72 pode utilizar os filtros de interpolação utilizados pelo codificador de vídeo 20 durante a codificação dos blocos de vídeo para calcular valores interpolados para sub-pixels de número inteiro de blocos de referência. Neste caso, a unidade de compensação de movimento 72 pode determinar os filtros de interpolação utilizados pelo codificador de vídeo 20 a partir dos elementos de sintaxe recebidos e utilizar os filtros de interpolação para produzir blocos preditivos.

[0161] A unidade de quantificação inversa 76 quantifica por inversão, isto é, desquantifica os coeficientes de transformada quantificados apresentados no fluxo de bits e decodificados pela unidade de decodificação por entropia 70. O processo de quantificação inversa pode incluir a utilização de um parâmetro de quantificação QPY calculado pelo decodificador de vídeo 30 para cada bloco de vídeo na fatia de vídeo de modo a se determinar o grau de quantificação e, da mesma maneira, o grau de quantificação inversa que deve ser aplicado. A unidade de transformada inversa 78 aplica uma transformada inversa, como, por exemplo, uma DCT inversa, uma transformada de número inteiro inversa ou um processo de transformada inversa conceitualmente semelhante, aos coeficientes de transformada de modo a se produzirem blocos residuais no domínio de pixel.

[0163] Depois que a unidade de compensação de movimento 72 gera o bloco preditivo para o bloco de vídeo atual com base nos vetores de movimento e em outros elementos de sintaxe, o decodificador de vídeo 30 forma um bloco de vídeo decodificado somando os blocos residuais da unidade de transformada inversa 78 com os blocos preditivos correspondentes gerados pela unidade de compensação de movimento 72. O somador 80 representa o componente ou componentes que executam esta operação de soma. Se desejado, um filtro de desbloqueio pode ser também aplicado para filtrar os blocos decodificados de modo a se removerem os artefatos de bloqueio. Outros filtros de malha (ou na malha de codificação ou depois da malha de codificação) podem ser também utilizados para suavizar as transições entre pixels ou senão aperfeiçoar a qualidade de vídeo. Os blocos de vídeo decodificados em um dado quadro ou imagem são em seguida armazenados na memória de imagens de referência 82, que armazena imagens de referência utilizadas para compensação de movimento subsequente. A memória de imagens de referência 82 armazena também vídeo decodificado para apresentação posterior em um aparelho de exibição, tal como o aparelho de exibição 32 da Figura 3.

[0164] Deve-se reconhecer que, dependendo do exemplo, determinados atos ou eventos de qualquer uma das técnicas aqui descritas podem ser executados em uma sequência diferente, podem ser adicionados, fundidos ou deixados de fora completamente (nem todos os atos ou eventos descritos são necessários para a prática das técnicas, por exemplo). Além do mais, em determinados exemplos, atos ou eventos podem ser executados concomitantemente, como, por exemplo, através de processamento de vários fluxos, processamento com interrupções ou vários processadores, e não sequencialmente.

[0165] Em um ou mais exemplos, As funções descritas podem ser implementadas em hardware, software, firmware ou qualquer combinação deles. Se implementadas em software, as funções podem ser armazenadas ou transmitidas através de uma ou mais instruções ou código em um meio passível de leitura por computador e executadas por uma unidade de processamento baseada em hardware. Os meios passíveis de leitura por computador podem incluir meios de armazenamento passíveis de leitura de computador que correspondem a um meio tangível, tal como um meio de armazenamento de dados ou meio de comunicação que inclui um meio que facilita a transferência de um programa de computador de um lugar para outro, de acordo com um protocolo de comunicação. Desta maneira, os meios passíveis de leitura por computador podem corresponder a (1) meios de armazenamento passíveis de leitura por computador tangíveis que são não transitórios ou (2) um meio de comunicação tal como um sinal ou onda portadora. Os meios de armazenamento de dados podem ser quaisquer meios disponíveis que possam ser acessados por um ou mais computadores ou um ou mais processadores para recuperar instruções, código e/ou estrutura de dados para implementação das técnicas descritas nesta revelação. Um produto de programa de computador pode incluir um meio passível de leitura por computador.

[0166] A título de exemplo, e não de limitação, tal meio de armazenamento passível de leitura por computador pode compreender RAM, ROM, EEPROM, CD-ROM ou outro armazenamento em disco óptico, armazenamento em disco magnético ou outros aparelhos de armazenamento magnético, memória flash ou qualquer outro meio que possa ser utilizado para armazenar código de programa desejado sob a forma de instruções ou estruturas de dados e que possa ser acessado por um computador. Além disto, qualquer conexão é

denominada apropriadamente de meio passível de leitura por computador. Por exemplo, se as instruções forem transmitidas de um site da Web, servidor ou outra fonte remota utilizando-se um cabo coaxial, cabo de fibra óptica, par trançado, linha de assinante digital (DSL) ou tecnologias sem fio tais como infravermelho, rádio e microonda, então o cabo coaxial, o cabo de fibra óptica, o par trançado, a DSL ou tecnologias sem fio tais como infravermelho, rádio e microonda são incluídos na definição de meio. Deve ficar entendido, contudo, que os meios de armazenamento passíveis de leitura por computador e os meios de armazenamento de dados não incluem conexões, ondas portadoras, sinais ou outros meios transitórios, mas são em vez disso direcionados a meios de armazenamento tangíveis não transitórios. O termo disco (*disk*) e disco (*disc*), conforme aqui utilizado, inclui disco compacto (CD), disco de laser, disco óptico, disco versátil digital (DVD), disco flexível e disco Blu-ray, em que usualmente discos (*disks*) reproduzem dados magneticamente, enquanto discos (*discs*) reproduzem dados opticamente com lasers. Combinações deles devem ser também incluídas dentro do alcance dos meios passíveis de leitura por computador.

[0167] As instruções podem ser executadas por um ou mais processadores, tais como um ou mais processadores de sinais digitais (DSP), micro-processadores de propósito geral, circuitos integrados específicos de aplicativo (ASICs), arranjos lógicos programáveis no campo (FPGAs) ou outros circuitos lógicos integrados ou discretos equivalentes. Por conseguinte, o termo "processador", conforme aqui utilizado, pode referir-se a qualquer uma das estruturas precedentes ou qualquer outra estrutura adequada para implementação das técnicas aqui descritas. Além disto, sob alguns aspectos, a funcionalidade aqui descrita pode ser

apresentada dentro de hardware dedicado e/ou em módulos de software configurados para codificação e decodificação ou incorporada a um CODEC combinado. Além disto, as técnicas podem ser completamente implementadas em um ou mais circuitos ou elemento lógicos.

[0168] As técnicas desta revelação podem ser implementadas em uma ampla variedade de aparelhos ou equipamentos, que incluem um aparelho telefônico sem fio, um circuito integrado (IC) ou um conjunto de ICs (isto é, um conjunto de chips). Diversos componentes, módulos ou unidades são descritos nesta revelação para enfatizar aspectos funcionais de aparelhos configurados para executar as técnicas reveladas, mas não exigem necessariamente a execução por unidades de hardware diferente. Em vez disso, conforme descrito acima, diversas unidades podem ser combinadas em uma unidade de hardware de CODEC ou providas por uma coleção de unidades de hardware inter-operantes, que incluem um ou mais processadores, conforme descrito acima em conjunto com software e/ou firmware adequado.

[0169] Foram descritos diversos exemplos. Estes e outros exemplos estão dentro do alcance das reivindicações seguintes.

REIVINDICAÇÕES

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

com base em uma determinação de que uma camada temporal, incluindo uma primeira imagem e uma segunda imagem, possui uma taxa de imagem constante, decodificar informações de temporização para os dados de vídeo, as informações de temporização incluindo um valor inteiro para os dados de vídeo;

determinar um valor de diferença entre um tempo de apresentação da primeira imagem e um tempo de apresentação da segunda imagem, tal que o valor de diferença seja igual ao valor inteiro multiplicado por um valor de pulso de relógio, em que a primeira imagem e a segunda imagem são consecutivas na ordem de saída; e

apresentar a primeira imagem em um primeiro tempo e apresentar a segunda imagem em um segundo tempo, tal que a diferença entre o segundo tempo e o primeiro tempo é o valor de diferença determinado.

2. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que compreende adicionalmente exibir, por um dispositivo de exibição, os dados de vídeo apresentados.

3. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que compreende adicionalmente determinar que a camada temporal incluindo a primeira imagem e a segunda imagem tem a taxa de imagens constante, em que as informações de temporização incluem dados definindo o valor inteiro, em que decodificar as informações de temporização para os dados de vídeo inclui

determinar o valor inteiro, e em que determinar o valor inteiro compreende, com base na determinação de que a camada temporal tem a taxa de imagem constante, decodificar os dados definindo o valor inteiro.

4. Método, de acordo com a reivindicação 3, **caracterizado** pelo fato de que determinar que a camada temporal tem a taxa de imagens constante compreende determinar que um `fixed_pic_rate_flag` tem um valor indicando que a camada temporal tem a taxa de imagens constante.

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

determinar, para cada camada temporal tendo uma taxa de imagens constante respectiva, um valor inteiro sinalizado respectivo; e

apresentar imagens de cada camada temporal tendo uma taxa de imagem constante respectiva de acordo com os respectivos valores inteiros multiplicados pelo valor de pulso de relógio.

6. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que decodificar as informações de temporização para os dados de vídeo inclui determinar o valor de pulso de relógio, em que determinar o valor de pulso de relógio inclui determinar o valor de pulso de relógio, tal que o valor de pulso de relógio é baseado, pelo menos em parte, em um valor de escala de tempo, em que o valor de escala de tempo é baseado em uma frequência de relógio.

7. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que a camada temporal inclui uma camada temporal mais elevada, compreendendo adicionalmente:

determinar um segundo valor de diferença entre o tempo de apresentação de uma terceira imagem e um tempo de apresentação de uma quarta imagem em uma camada temporal mais baixa tendo um valor de identificação de camada temporal que é inferior a um valor de identificação temporal que é associado à camada temporal mais elevada, um segundo valor inteiro associado com a camada temporal mais baixa e o valor de pulso de relógio; e

apresentar imagens da camada temporal mais baixa com base no segundo valor de diferença.

8. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que o tempo de apresentação para a primeira imagem compreende um primeiro valor de contagem de ordem de imagens, POC, e em que o tempo de apresentação para a segunda imagem compreende um segundo valor de POC.

9. Aparelho para apresentar dados de vídeo, **caracterizado** pelo fato de que compreende:

meios para decodificar informações de temporização para os dados de vídeo em que as informações de temporização são determinadas com base em uma determinação que uma camada temporal, incluindo uma primeira imagem e uma segunda imagem, possui uma taxa de imagem constante, as informações de temporização incluindo um valor inteiro para os dados de vídeo;

meios para determinar um valor de diferença entre um tempo de apresentação da primeira imagem e um tempo de apresentação da segunda imagem, tal que o valor de

diferença é igual a um valor inteiro multiplicado por um valor de pulso de relógio, em que a primeira imagem e a segunda imagem são consecutivas na ordem de saída; e

meios para apresentar a primeira imagem em um primeiro tempo e apresentar a segunda imagem em um segundo tempo, tal que a diferença entre o segundo tempo e o primeiro tempo é o valor de diferença determinado.

10. Aparelho, de acordo com a reivindicação 9, **caracterizado** pelo fato de que compreende adicionalmente meios para determinar que a camada temporal incluindo a primeira imagem e a segunda imagem tem a taxa de imagens constante, em que as informações de temporização incluem dados definindo o valor inteiro, em que os meios para decodificar as informações de temporização para os dados de vídeo incluem meios para determinar o valor inteiro, e em que os meios para determinar o valor inteiro compreendem meios para decodificar, com base na determinação de que a camada temporal tem a taxa de imagens constante, os dados definindo o valor inteiro.

11. Aparelho, de acordo com a reivindicação 10, **caracterizado** pelo fato de que os meios para determinar que a camada temporal tem a taxa de imagens constante compreendem meios para determinar que um `fixed_pic_rate_flag` tem um valor indicando que a camada temporal tem a taxa de imagens constante.

12. Aparelho, de acordo com a reivindicação 10, **caracterizado** pelo fato de que compreende adicionalmente:

meios para determinar, para cada camada temporal que tendo uma taxa de imagens constante respectiva, um valor inteiro sinalizado respectivo; e

meios para apresentar imagens de cada camada temporal tendo taxas de imagens constantes respectivas de acordo com valores inteiro respectivos multiplicados pelo valor de pulso de relógio.

13. Aparelho, de acordo com a reivindicação 9, **caracterizado** pelo fato de que a camada temporal inclui uma camada temporal mais elevada, compreendendo adicionalmente:

meios para determinar um segundo valor de diferença entre um tempo de apresentação de uma terceira imagem e um tempo de apresentação de uma quarta imagem em uma camada temporal mais baixa tendo um valor de identificação de camada temporal que é inferior a um valor de identificação temporal que é associado à camada temporal mais elevada, um segundo valor inteiro associado com a camada temporal mais baixa e o valor de pulso de relógio; e

meios para apresentar imagens da camada temporal mais baixa com base no segundo valor de diferença.

14. Aparelho, de acordo com a reivindicação 9, **caracterizado** pelo fato de que o tempo de apresentação para a primeira imagem compreende um primeiro valor de contagem de ordem de imagens, POC, e em que o tempo de apresentação para a segunda imagem compreende um segundo valor de POC.

15. Memória **caracterizada** pelo fato de que compreende instruções armazenadas na mesma, as instruções sendo executadas por um computador para realizar o método conforme definido em qualquer uma das reivindicações 1 a 8.

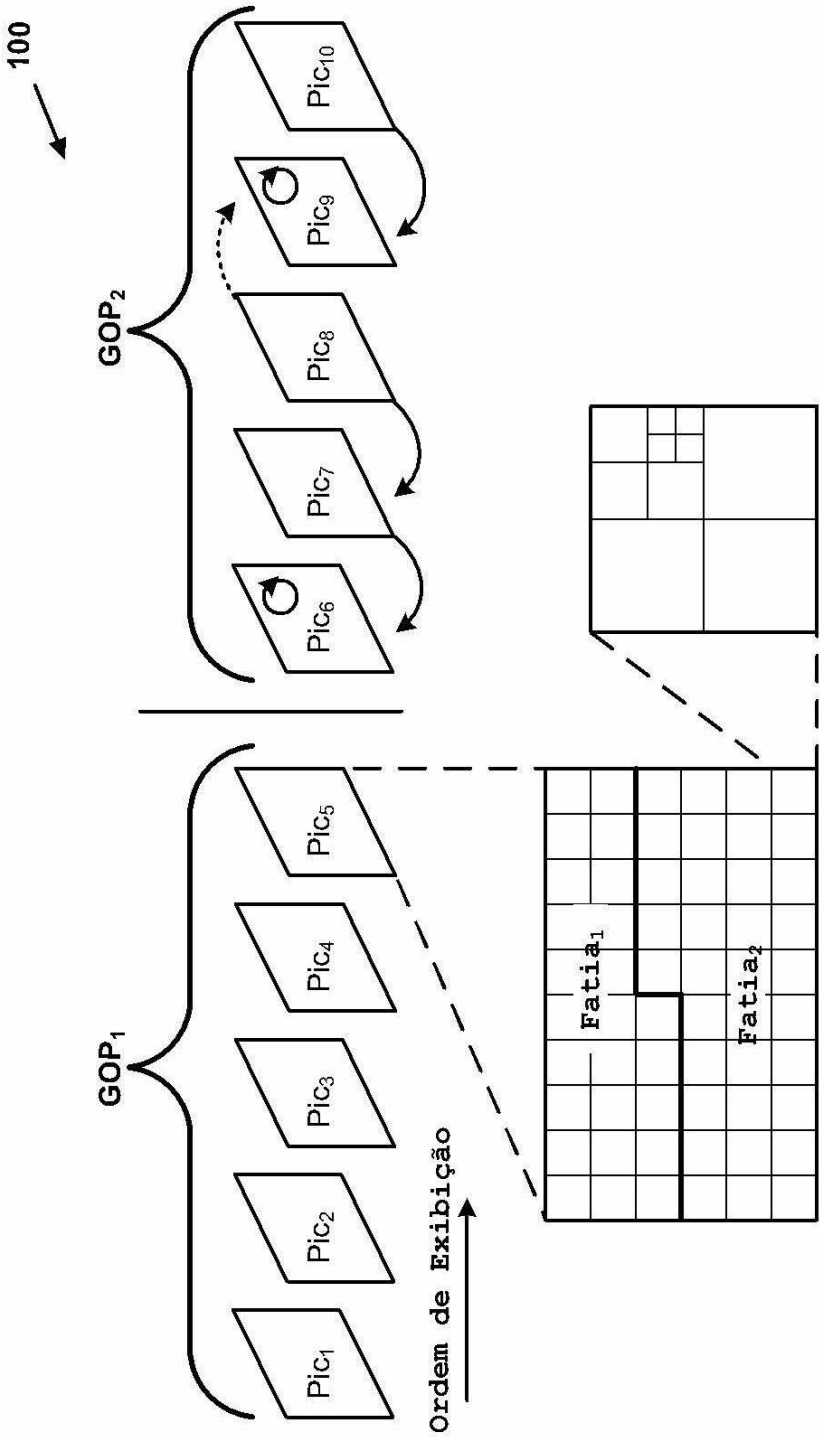


FIG. 1

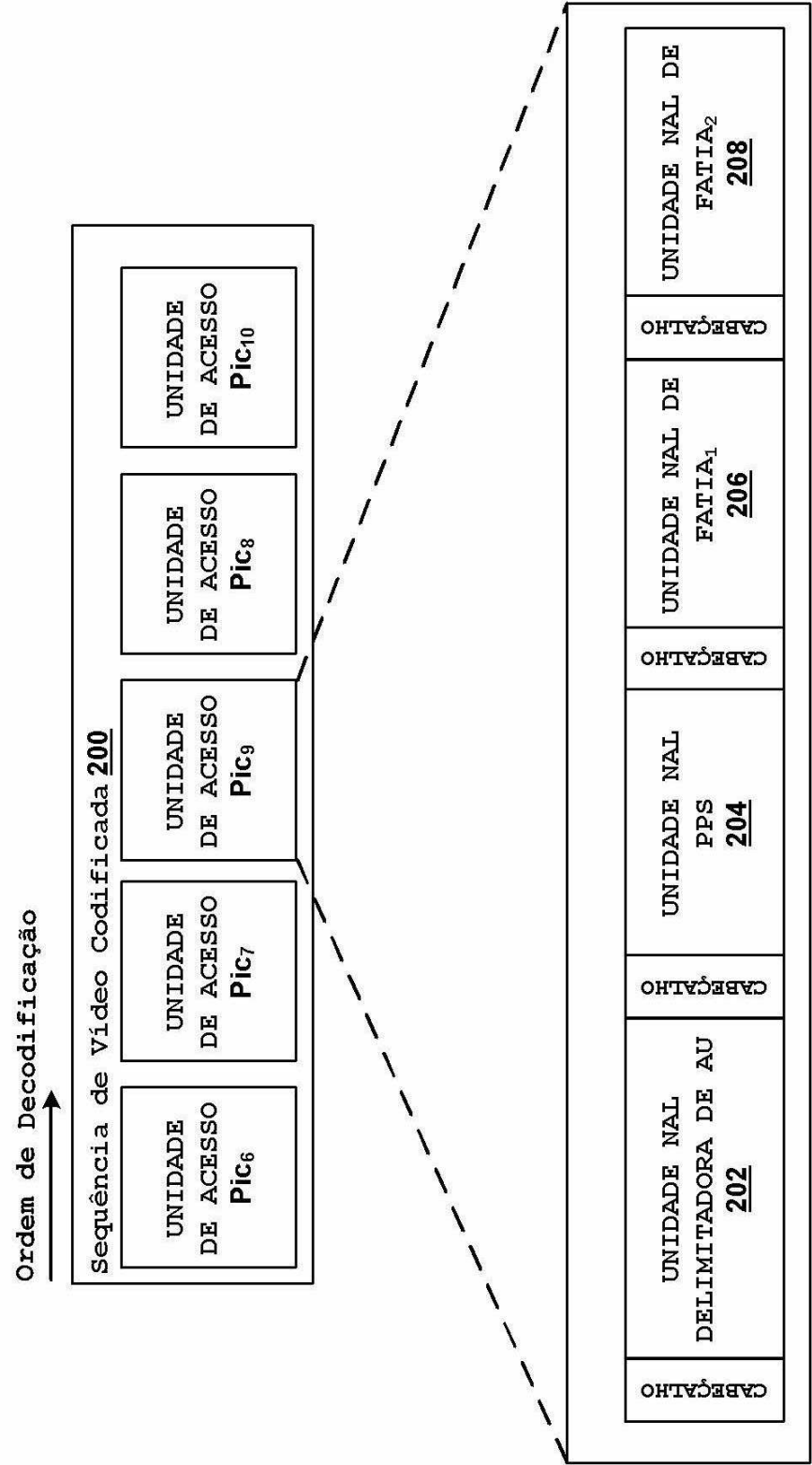
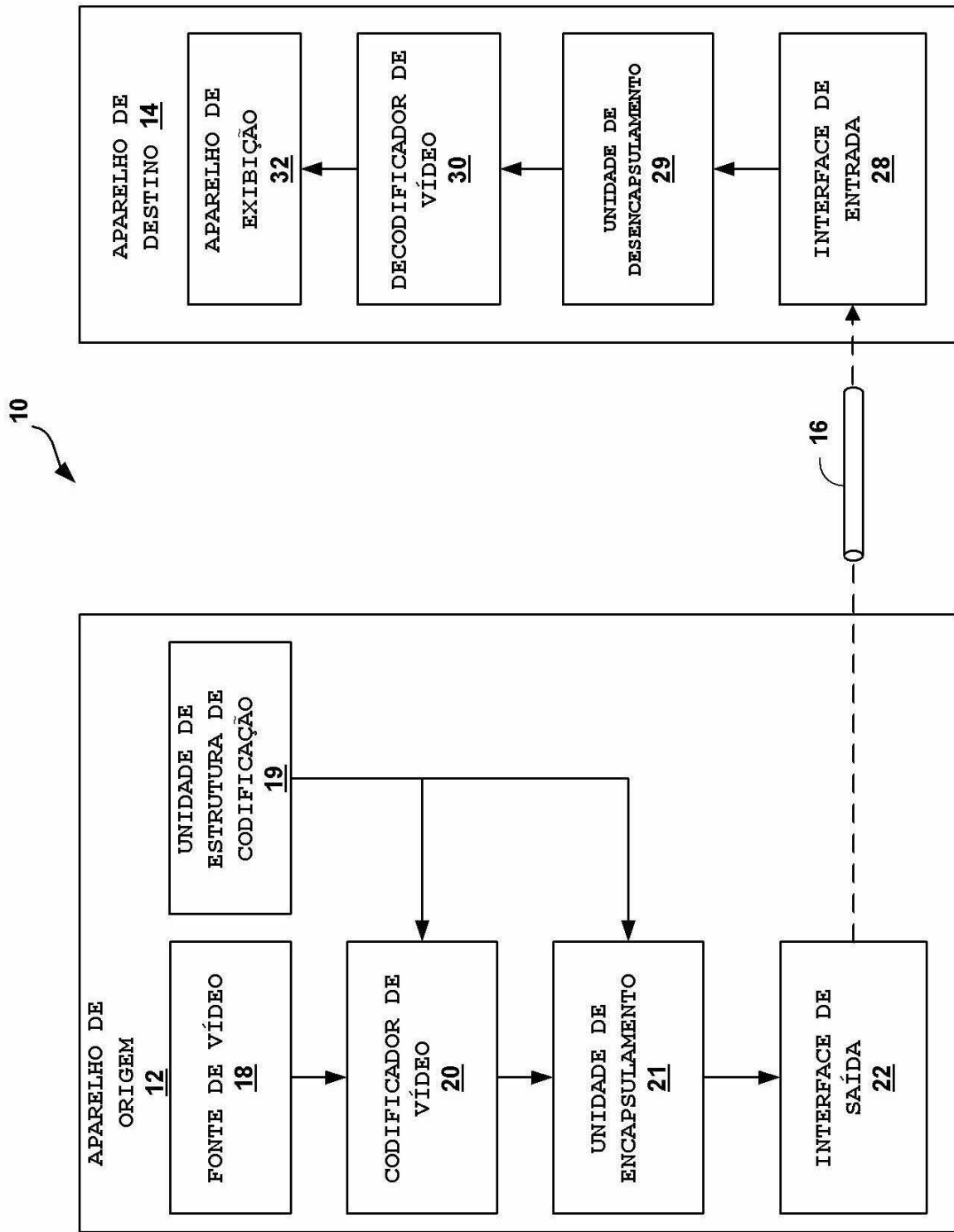


FIG. 2



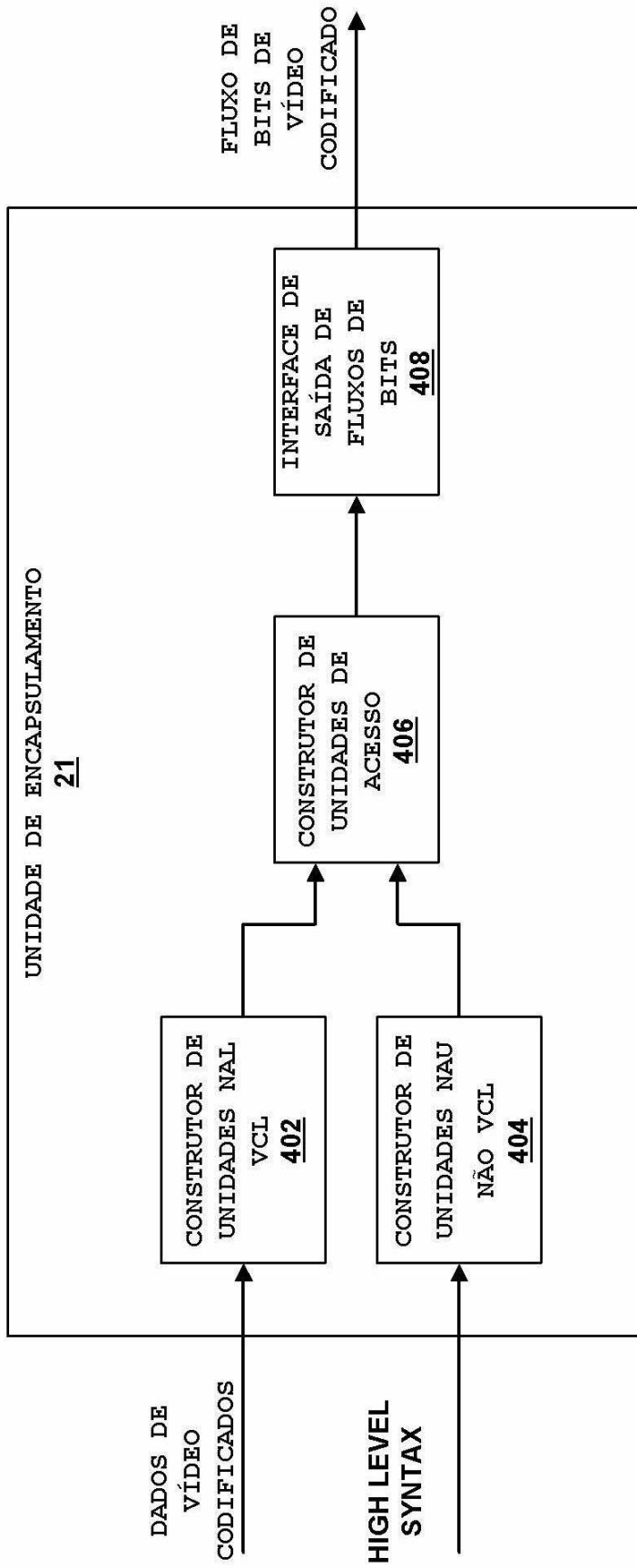


FIG. 4

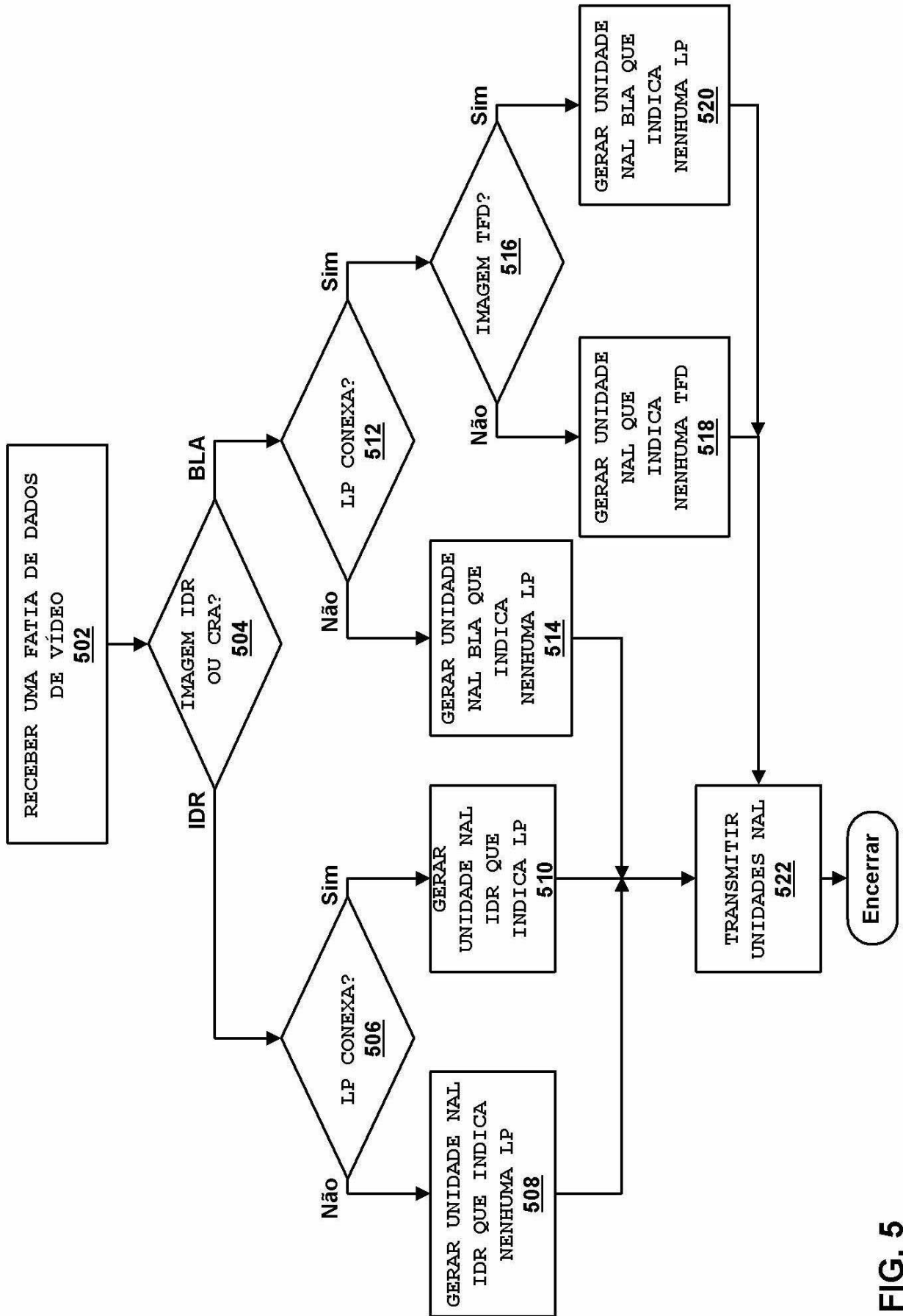


FIG. 5

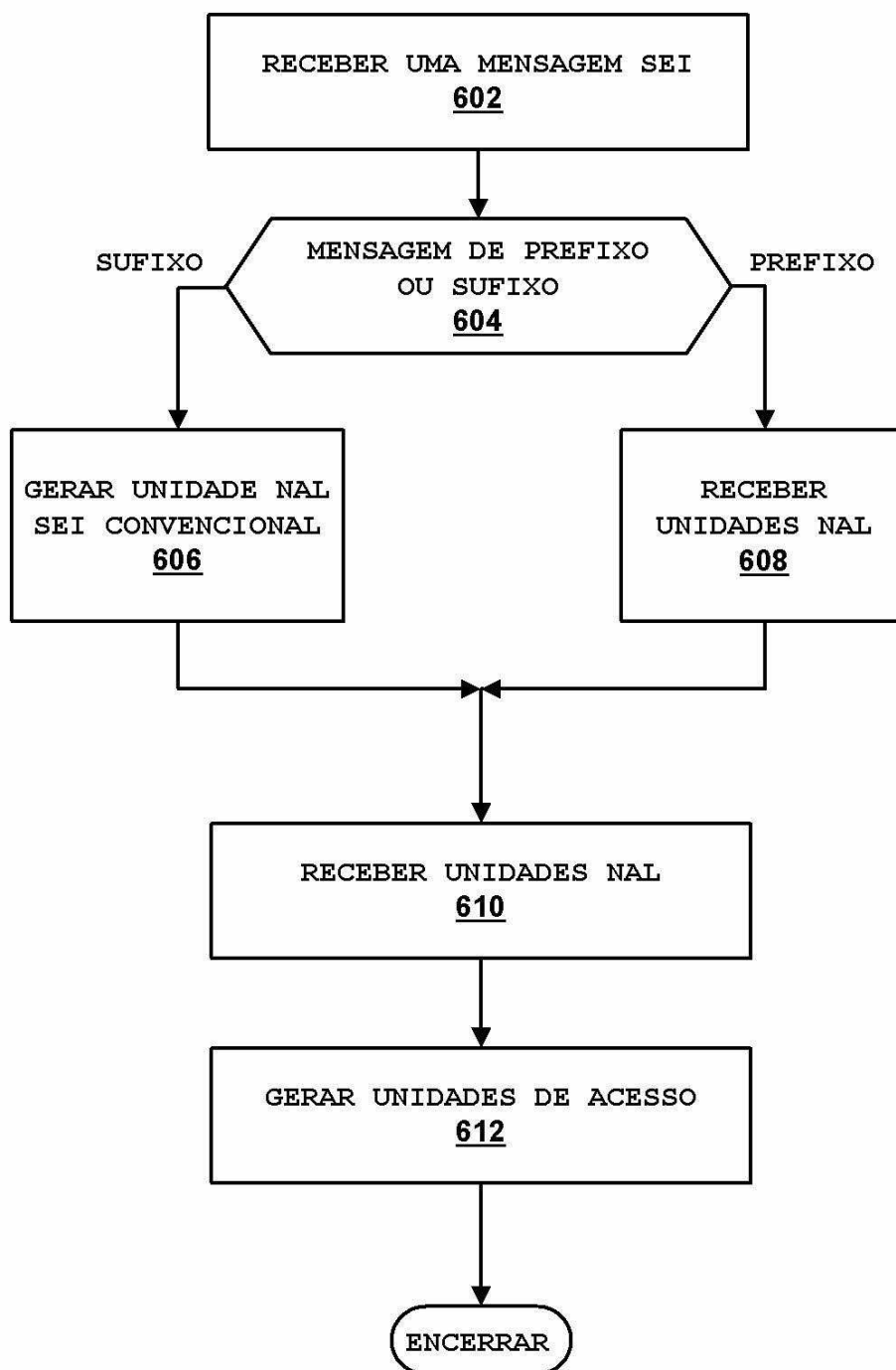


FIG. 6

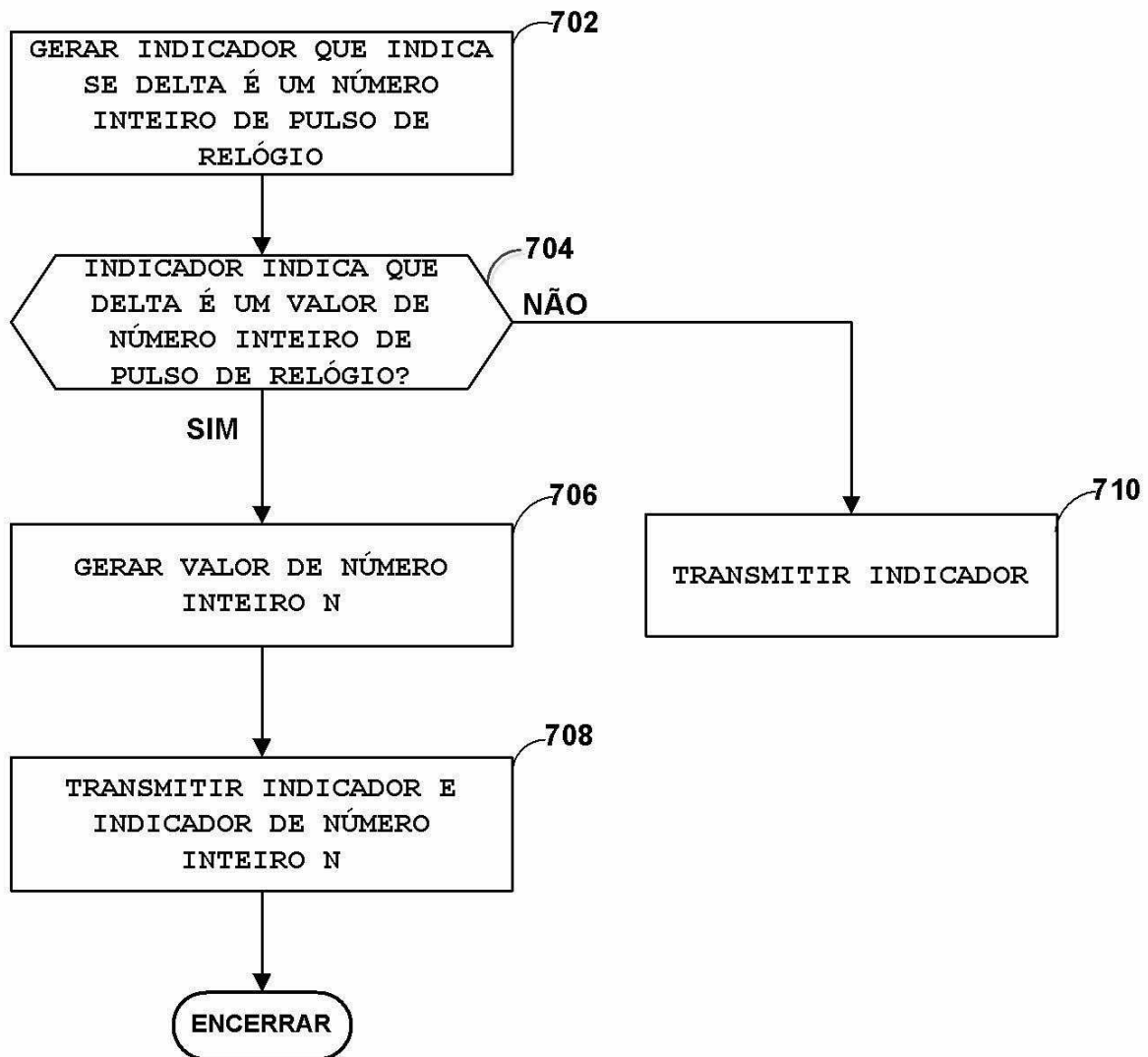


FIG. 7

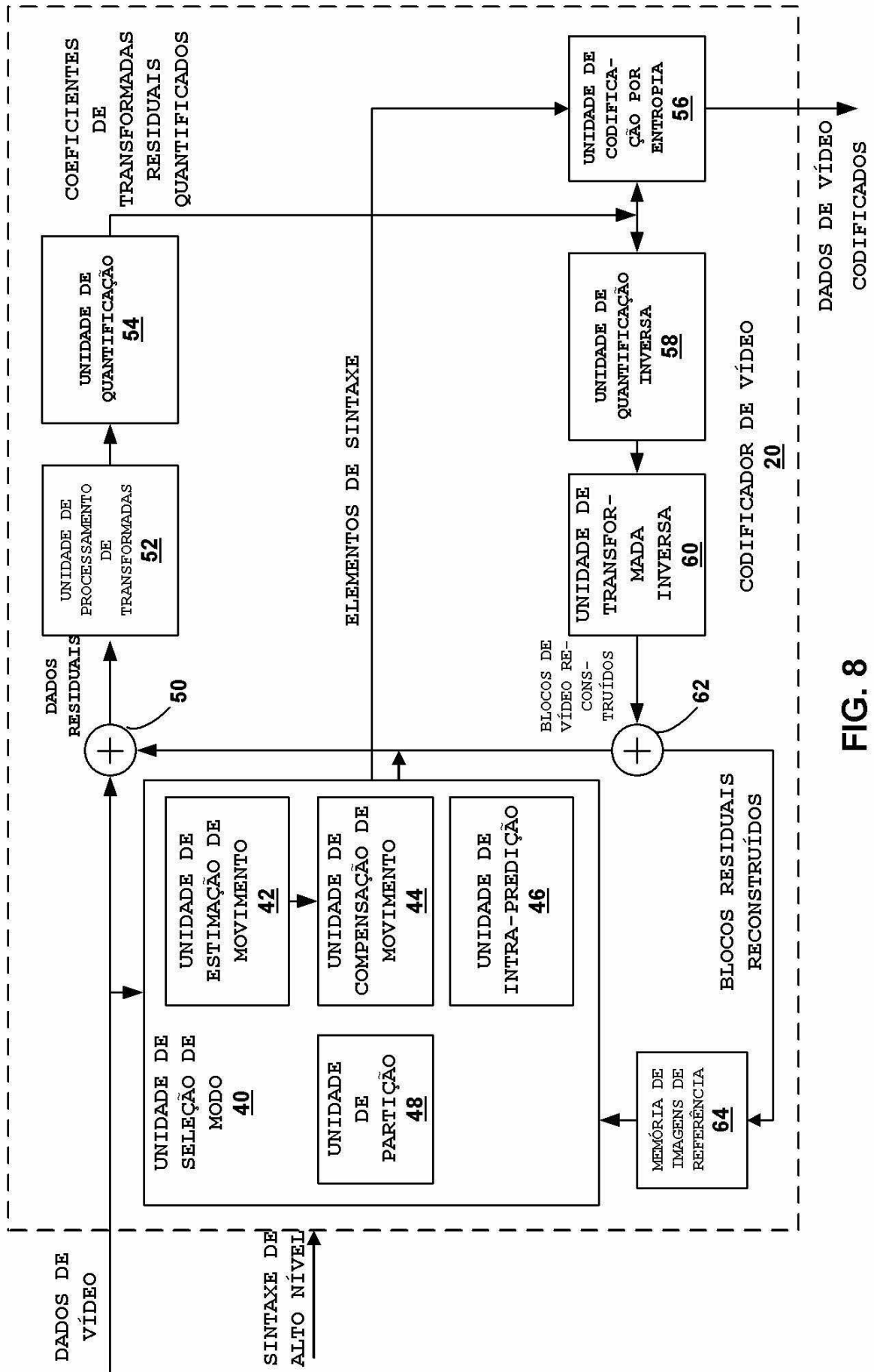
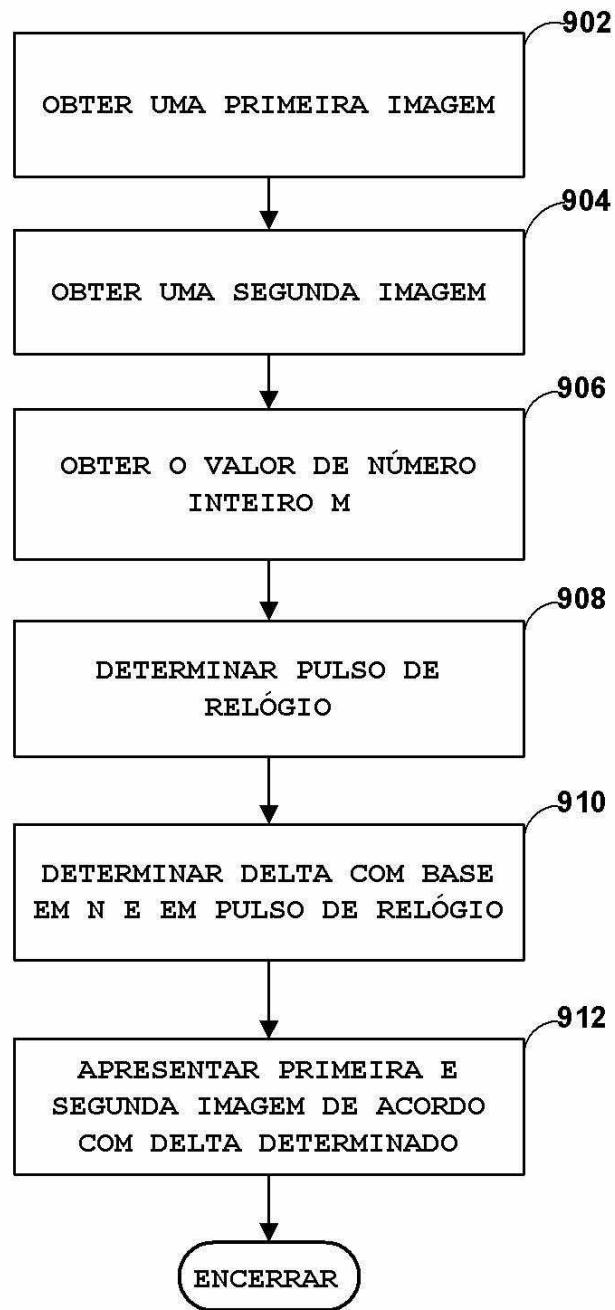


FIG. 8

**FIG. 9**

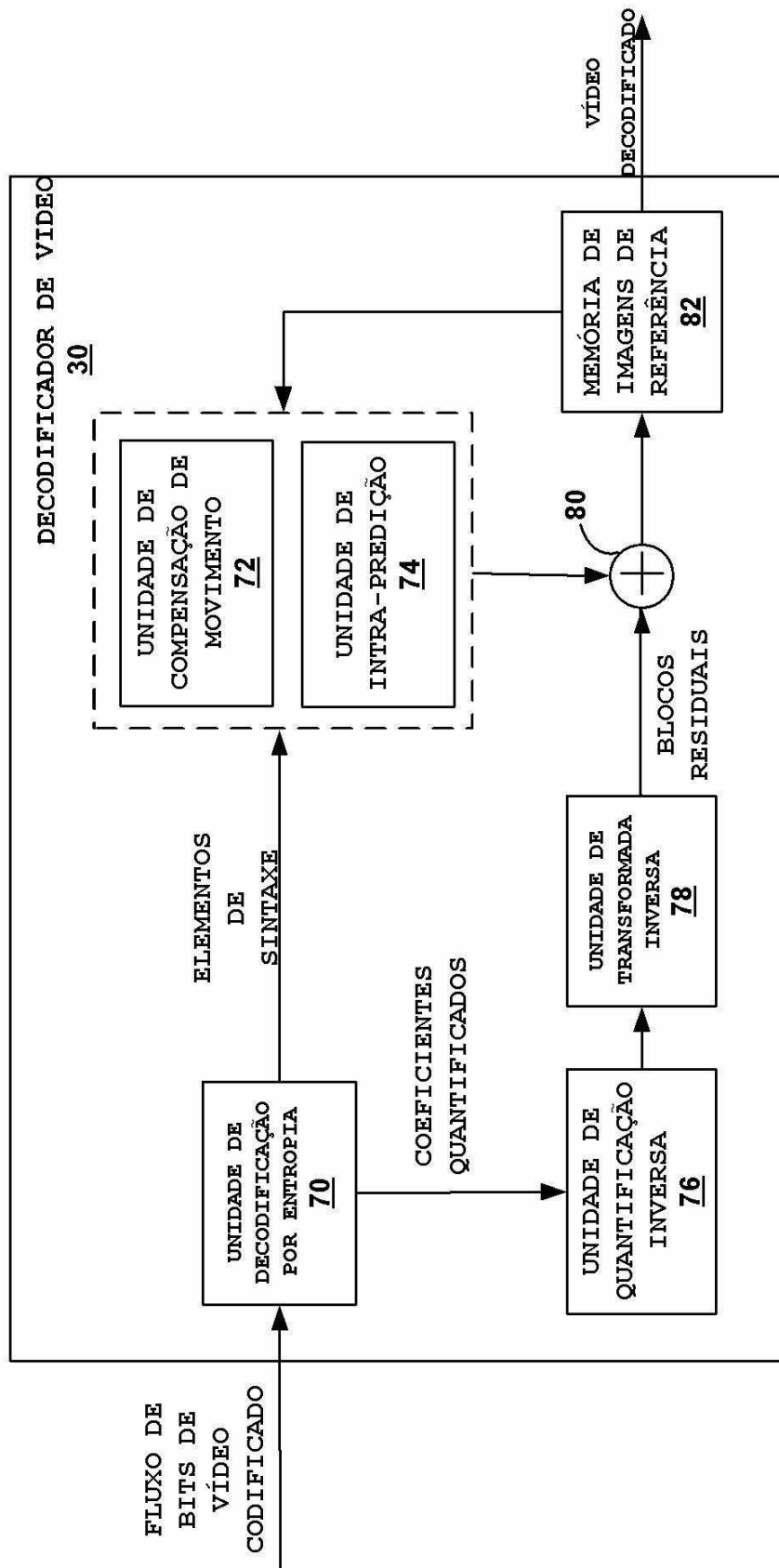


FIG. 10