



República Federativa do Brasil

Ministério do Desenvolvimento, Indústria,
Comércio e Serviços

Instituto Nacional da Propriedade Industrial



(11) BR 112016008241-9 B1

(22) Data do Depósito: 14/10/2014

(45) Data de Concessão: 14/02/2023

(54) Título: INDICAÇÃO DE PROCESSAMENTO PARALELO EM CODIFICAÇÃO DE VÍDEO

(51) Int.Cl.: H04N 19/00.

(30) Prioridade Unionista: 10/10/2014 US 14/512,199; 14/10/2013 US 61/890,763.

(73) Titular(es): QUALCOMM INCORPORATED.

(72) Inventor(es): KRISHNAKANTH RAPAKA; FNU HENDRY; YE-KUI WANG.

(86) Pedido PCT: PCT US2014060445 de 14/10/2014

(87) Publicação PCT: WO 2015/057677 de 23/04/2015

(85) Data do Início da Fase Nacional: 13/04/2016

(57) Resumo: INDICAÇÃO DE PROCESSAMENTO PARALELO EM CODIFICAÇÃO DE VÍDEO. Em um exemplo, um método de decodificação de dados de vídeo inclui a decodificação, a partir de um conjunto de parâmetros de vídeo (VPS) de um fluxo de bits de múltiplas camadas, em que os dados indicam pelo menos uma dentre uma configuração em tile para camadas do fluxo de bits de múltiplas camadas ou uma configuração de processamento paralelo para camadas do fluxo de bits de múltiplas camadas. O método também inclui decodificar o fluxo de bits de múltiplas camadas de acordo com os dados decodificados a partir do VPS.

"INDICAÇÃO DE PROCESSAMENTO PARALELO EM CODIFICAÇÃO DE
VÍDEO"

[0001] Este pedido reivindica o benefício do Pedido de Patente Provisório nº 61/890.763, depositado em 14 de outubro de 2013, cujo conteúdo está incorporado a título de referência em sua totalidade ao presente documento.

CAMPO DA TÉCNICA

[0002] Esta revelação refere-se à codificação de vídeo e, mais particularmente, a técnicas para o processamento paralelo na codificação de vídeo.

ANTECEDENTES

[0003] As capacidades de vídeo digital podem ser incorporadas em uma ampla faixa de dispositivos, incluindo televisões digitais, sistemas de difusão direta digital, sistemas de difusão sem fio, assistentes digitais pessoais (PDAs), computadores do tipo laptop ou desktop, computadores do tipo tablet, leitores de e-book, câmeras digitais, dispositivo de gravação digitais, reprodutores de mídias digitais, dispositivos de video game, consoles de video game, telefones ou celulares de rádio-satélite, conhecidos como "telefones inteligentes", dispositivos de teleconferência por vídeo, dispositivos de transmissão contínua de vídeo e similares. Os dispositivos de vídeo digital implantam técnicas de compressão de vídeo, como as descritas nos padrões definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificação de Vídeo Avançada (AVC), o padrão de Codificação de Vídeo de Alta Eficiência (HEVC) atualmente sob desenvolvimento e extensões de tais padrões. Os dispositivos de vídeo podem transmitir, receber, criptar, decodificar e/ou armazenar informações de vídeo digital de maneira mais eficiente implantando-se tais técnicas de compressão de vídeo.

[0004] As técnicas de compressão de vídeo

incluem previsão espacial (intraimagem) e/ou previsão temporal (interimagem) para reduzir ou remover a redundância inerente em sequências de vídeo. Para a codificação de vídeo com base em bloco, uma fatia de vídeo (isto é, um quadro de vídeo ou uma porção de um quadro de vídeo) pode ser particionada em blocos de vídeo, que também podem ser chamados de CTBs, CUs e/ou nós de codificação. Os blocos de vídeo em uma fatia intracodificada (I) de uma imagem são criptados com o uso de previsão espacial em relação às amostras de referência em blocos próximos na mesma imagem. Os blocos de vídeo em uma fatia intercodificada (P ou B) de uma imagem podem usar previsão espacial em relação a amostras de referência em blocos próximos na mesma imagem ou previsão temporal em relação às amostras de referência em outras imagens de referência. As imagens podem ser denominadas como quadros, e as imagens de referência podem ser denominadas como quadros de referência.

[0005] A previsão temporal ou espacial 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. Um bloco intercodificado é criptado de acordo com um vetor de movimento que aponta para um bloco de amostras de referência que forma o bloco preditivo, sendo que os dados residuais indicam a diferença entre o bloco codificado e o bloco preditivo. Um bloco intracodificado é criptado de acordo com um modo de intracodificação e os dados residuais. Para compressão adicional, os dados residuais podem ser transformados do domínio de pixel para um domínio de transformada, resultando em coeficientes residuais que podem, em seguida, ser quantizados. O coeficiente de transformada quantizado, disposto inicialmente em uma matriz bidimensional e pode ter sua varredura realizada a fim de

produzir um vetor monodimensional de coeficiente de transformada, e a codificação por entropia pode ser aplicada para conseguir ainda mais compressão.

SUMÁRIO

[0006] De modo geral, esta revelação descreve técnicas para o processamento paralelo de codificação de vídeo. Por exemplo, esta revelação descreve técnicas para o processamento paralelo em processos de codificação de vídeo de múltiplas camadas, incluindo extensões de múltiplas camadas do padrão de Codificação de Vídeo Altamente Eficaz (HEVC). As técnicas desta revelação incluem fornecer uma indicação precoce do uso ou disponibilidade de ferramentas de processamento paralelo (por exemplo, em um nível de conjunto de parâmetros de vídeo (VPS)) para um processo de codificação de vídeo.

[0007] Em um exemplo, um método de decodificação dados de vídeo inclui decodificação, a partir de um conjunto de parâmetros de vídeo (VPS) de um fluxo de bits de múltiplas camadas, dados que indicam pelo menos uma dentre uma configuração de pequenos blocos para camadas do fluxo de bits de múltiplas camadas ou uma configuração de processamento paralelo para camadas do fluxo de bits de múltiplas camadas, e decodificação do fluxo de bits de múltiplas camadas de acordo com os dados decodificados a partir do VPS.

[0008] Em outro exemplo, um método de criptação dados de vídeo inclui criptar, em um conjunto de parâmetros de vídeo (VPS) de um fluxo de bits de múltiplas camadas, dados que indicam pelo menos uma dentre uma configuração de pequenos blocos para camadas do fluxo de bits de múltiplas camadas ou uma configuração de processamento paralelo para camadas do fluxo de bits de múltiplas camadas, e a criptação do fluxo de bits de múltiplas camadas que inclui criptar o

VPS do fluxo de bits de múltiplas camadas.

[0009] Em outro exemplo, um aparelho que processa dados de vídeo inclui uma memória que armazena dados de vídeo de múltiplas camadas, e um codificador de vídeo configurado para processar dados que indicam pelo menos uma dentre uma configuração de pequenos blocos para camadas dos dados de vídeo de múltiplas camadas ou uma configuração de processamento paralelo para camadas dos dados de vídeo de múltiplas camadas, em que os dados são associados a um conjunto de parâmetros de vídeo (VPS) de um fluxo de bits que inclui os dados de vídeo de múltiplas camadas, e processar os dados de vídeo de múltiplas camadas de acordo com os dados do VPS.

[0010] Em outro exemplo, um aparelho que realiza os dados de vídeo inclui meios para processar dados que indicam pelo menos uma dentre uma configuração de pequenos blocos para camadas do fluxo de bits de múltiplas camadas ou uma configuração de processamento paralelo para camadas do fluxo de bits de múltiplas camadas, em que os dados são associados a um conjunto de parâmetros de vídeo (VPS) do fluxo de bits de múltiplas camadas, e meios para processar o fluxo de bits de múltiplas camadas de acordo com os dados do VPS.

[0011] Em outro exemplo, um meio de armazenamento legível por computador não transitório tem instruções armazenadas no mesmo que, quando executadas, levam os um ou mais processadores a processar dados que indicam pelo menos uma dentre uma configuração de pequenos blocos para camadas do fluxo de bits de múltiplas camadas ou uma configuração de processamento paralelo para camadas do fluxo de bits de múltiplas camadas, em que os dados são associados a um conjunto de parâmetros de vídeo (VPS) do fluxo de bits de múltiplas camadas, e processar o fluxo de

bits de múltiplas camada de acordo com os dados do VPS.

[0012] Os detalhes de um ou mais aspectos das técnicas são estabelecidos nos desenhos anexos e na descrição abaixo. Outros recursos, objetos e vantagens da revelação ficarão evidentes a partir da descrição e dos desenhos, bem como a partir das reivindicações.

BREVE DESCRIÇÃO DOS DESENHOS

[0013] A Figura é um diagrama de blocos que ilustra um sistema de criptação e decodificação de vídeo exemplificativo que pode utilizar as técnicas descritas nesta revelação.

[0014] A Figura 2 é um diagrama de blocos que ilustra um criptador de vídeo exemplificativo que pode implantar as técnicas descritas nesta revelação.

[0015] A Figura 3 é um diagrama de blocos que ilustra um decodificador de vídeo exemplificativo que pode implantar as técnicas descritas nesta revelação.

[0016] A Figura 4A é um diagrama conceitual que ilustra um exemplo de pontos de sincronização para processamento paralelo de frente de onda.

[0017] Figura 4B é um diagrama conceitual que ilustra outro exemplo de pontos de sincronização para processamento paralelo de frente de onda.

[0018] A Figura 5 é um diagrama conceitual que ilustra exemplo pontos de sincronização para processamento paralelo de fileiras de bloco de vídeo.

[0019] A Figura 6 é um diagrama conceitual que ilustra exemplo de pequenos blocos de acordo com o padrão HEVC.

[0020] A Figura 7 é um diagrama conceitual que ilustra um exemplo de padronização de previsão de Codificação de Vídeo de Múltiplas Vistas (MVC).

[0021] A Figura 8 é um diagrama conceitual que

ilustra diversas dimensões escalonáveis para codificação de vídeo escalonável (SVC).

[0022] A Figura 9 é um diagrama conceitual que ilustra um exemplo da estrutura de codificação de SVC.

[0023] A Figura 10 é um diagrama conceitual que ilustra unidades de acesso exemplificativas (Alls) em um fluxo de bits de múltiplas camadas.

[0024] A Figura 11 ilustra uma técnica para processar dados de vídeo de múltiplas camadas consistente com esta revelação.

[0025] A Figura 12 ilustra outra técnica para processar dados de vídeo de múltiplas camadas consistente com esta revelação.

DESCRIÇÃO DETALHADA

[0026] Os aspectos desta revelação podem se referir a diversas técnicas para indicação precoce de ferramentas de processamento paralelo na codificação de vídeo, conforme descrito em maiores detalhes abaixo. Em alguns exemplos, as técnicas podem ser realizadas com extensões de múltiplas camadas para um padrão de Codificação de Vídeo Altamente Eficaz, tal como uma extensão de Codificação de Vídeo de Múltiplas Vistas para HEVC (MV-HEVC), uma Múltiplas vistas mais extensão de Codificação de Vídeo de profundidade para HEVC (3D-HEVC), ou uma extensão de Codificação de Vídeo Escalonável (SVC) para HEVC (SHVC), conforme observado abaixo. As técnicas desta revelação, no entanto, não são limitadas a qualquer padrão de codificação de vídeo particular, e também pode, ou alternativamente, ser usada com outras extensões para HEVC, outros padrões de codificação de múltiplas vistas (com ou sem uma componente de profundidade) e/ou outros padrões de vídeo de múltiplas camadas. Além disso, técnicas desta revelação, conforme descritas abaixo, podem ser aplicadas independentemente ou

em combinação.

[0027] As ferramentas de processamento paralelo podem ser usadas para criptar ou decodificar dados de vídeo em paralelo. Por exemplo, algumas ferramentas de processamento paralelo podem permitir porções espaciais diferentes de uma única imagem de vídeo a ser criptada ou decodificada em paralelo. Em um exemplo, para fins ilustrativos, com referência ao padrão HEVC observado acima, o processamento paralelo pode ser alcançado com o uso de processamento paralelo de frente de onda (WPP). WPP é uma codificação de vídeo técnica em que duas ou mais fileiras de blocos de vídeo (por exemplo, fileiras de bloco de árvore codificado (CTB), conforme descrito abaixo) são criptadas e/ou decodificadas aproximadamente no mesmo período de tempo (isto é, em paralelo).

[0028] Devido às interdependências em intraprevisão, interprevisão, e determinada análise de elemento de sintaxe, que usam WPP, a decodificação paralela das fileiras de blocos pode ser sincronizada. Isto é, os blocos de vídeo em uma fileira atual de blocos podem usar informações codificadas dos blocos de vídeo em uma fileira de blocos acima da fileira atual de blocos. Ao processar duas fileiras de blocos em paralelo, um decodificador de vídeo pode sincronizar o processamento entre blocos nas duas fileiras para garantir que os dados exigidos por uma fileira inferior de blocos já foram codificados na fileira superior de blocos, nesse sentido, o processo de WPP pode ser considerado como dividido em dois processos principais que incluem um processo de análise de fluxo de bits e um processo de reconstrução. O processo de reconstrução pode incluir a sincronização entre fileiras de blocos.

[0029] Outra técnica de processamento paralelo inclui usar os chamados pequenos blocos para a codificação

de vídeo. Por exemplo, HEVC define pequenos blocos como um número inteiro de blocos que ocorrem simultaneamente em uma coluna e uma fileira, em ordem consecutivamente em uma varredura de rastreamento do tile. A divisão de cada imagem de pequenos blocos pode ser chamado de partição. Pequenos blocos em uma imagem são ordenados consecutivamente na varredura de rastreamento de tile da imagem. O número de pequenos blocos e as localizações de seus limites podem ser definidos para toda sequência ou alterados de imagem a imagem. Os limites de tile, de modo semelhante aos limites de fatia, quebram as dependências de análise e de previsão de modo que um tile possa ser processado independentemente de outro tile. No entanto, em alguns exemplos, filtros in-loop (por exemplo, filtros de desbloqueio e de deslocamento adaptável de amostra) podem ainda atravessar os limites de tile.

[0030] O uso de pequenos blocos pode aprimorar o paralelismo, devido ao fato de que nenhuma (ou relativamente pouca) comunicação é necessária entre os processadores ou núcleos de processador para decodificação por entropia e reconstrução de compensação de movimento. Consequentemente, pequenos blocos diferentes de uma imagem podem ser processados por processadores ou núcleos de processador diferentes. Além disso, pequenos blocos podem exibir uma eficácia de codificação relativamente melhor quando comparados às fatias, devido ao fato dos pequenos blocos permitirem formatos de partição de imagem que contêm amostras com correlação potencialmente mais alta que as fatias. Pequenos blocos também podem reduzir a suspensão de cabeçalho de fatia.

[0031] Enquanto as técnicas de WPP e de tile descritas acima podem fornecer o processamento paralelo dentro de uma camada particular, em alguns exemplos,

múltiplas camadas de dados de vídeo também pode ser criptadas ou decodificadas em paralelo. Uma "camada" de dados de vídeo pode ser chamada de modo geral de uma sequência de imagens que tem pelo menos uma característica comum, tal como uma vista, uma taxa de quadro, uma resolução, ou similares. Por exemplo, uma camada pode incluir dados de vídeo associados a uma vista particular (por exemplo, perspectiva) de dados de vídeo de múltiplas vistas (conforme descrito, por exemplo, com referência à Figura 7 abaixo). Como outro exemplo, uma camada pode incluir dados de vídeo associados a uma camada particular de dados de vídeo escalonáveis (conforme descrito, por exemplo, com referência às Figuras 8 a 10 abaixo). Desse modo, esta revelação pode se referir de modo intercambiável a uma camada e uma vista de dados de vídeo. Isto é, uma vista de dados de vídeo pode ser chamada de uma camada de dados de vídeo, e vice-versa. Além disso, os termos previsão entre vistas e previsão de intercamada pode se referir de modo intercambiável à previsão entre múltiplas camadas e/ou vistas de dados de vídeo. Além disso, um codec de múltiplas camadas (também chamado de codificador de vídeo de múltiplas camadas ou criptador e decodificador de múltiplas camadas) pode se referir conjuntamente a um codec de múltiplas vistas ou um codec escalonável (por exemplo, um codec configurado para criptar e/ou decodificar dados de vídeo que usam MV-HEVC, 3D-HEVC, SHVC, ou outra técnica de codificação de múltiplas camadas).

[0032] Em alguns exemplos, múltiplas camadas de dados de vídeo podem ser codificadas em paralelo com o uso de um atraso de deslocamento. Um atraso de deslocamento pode ser chamado, de modo geral, de atraso associado ao processamento (por exemplo, criptação ou decodificação) de múltiplas camadas em paralelo. Por exemplo, um decodificador de vídeo pode decodificar mais que uma camada de um fluxo de

bits de múltiplas camadas em paralelo. Ao usar as técnicas de previsão de intercamada para decodificar uma camada atual, no entanto, o decodificador de vídeo pode acessar os dados de referência de uma camada de referência que é diferente da camada atual. Os dados de referência devem estar disponíveis (por exemplo, decodificados) a fim de serem usados como uma intercamada de referência para decodificar a camada atual. Consequentemente, de modo geral, o decodificador de vídeo finaliza a decodificação de todas camadas de referência em um esquema de decodificação de múltiplas camadas antes da decodificação de uma camada que se refere as uma ou mais dentre as camadas de referência.

[0033] Em vez de esperar que uma camada de referência inteira seja finalizada (por exemplo, decodificada), o decodificador de vídeo pode atrasar a iniciação da decodificação de uma camada atual até que pelo menos para da camada de referência tenha sido decodificada. O atraso de deslocamento pode indicar de modo geral uma quantidade mínima de uma camada de referência que deve ser decodificada antes do decodificador de vídeo começar a decodificação de uma camada atual. A implantação de um atraso de deslocamento pode ajudar a garantir que os dados de referência de intercamada estejam disponíveis para serem usados como uma referência, enquanto ainda permite que pelo menos uma porção da camada de referência e da camada atual sejam decodificadas em paralelo. Por exemplo, um decodificador paralelo pode começar a decodificar uma camada aprimorada assim que o atraso de deslocamento de camada de referência especificado for alcançado. Embora o exemplo acima seja descrito com referência a decodificação de vídeo (conforme realizada por um decodificador paralelo), deve ser entendido que similar técnicas podem ser aplicadas por um criptador de vídeo durante a criptação paralela,

[0034] O projeto de extensões de HEVC e HEVC (por exemplo, tais como MV-HEVC, 3D-HEVC ou SHVC) podem exibir determinadas desvantagens. Por exemplo, quando usados através de camadas, determinadas configurações de pequenos blocos e de WPP podem afetar o projeto e implantação de pipelining (por exemplo, alocação de recursos através de camadas) de um codec. Em alguns exemplos, um projeto de pipeline pode precisar se alterar para configurações diferentes de pequenos blocos e de WPP. Por exemplo, ao realizar a previsão de intercamada com o uso de pequenos blocos, em que pequenos blocos que têm a mesma localização espacial relativa nas camadas devem ser processados pelos mesmos recursos de processamento (por exemplo, o mesmo núcleo de processamento) de modo que os dados para previsão de intercamada não precisem ser acessados por núcleos de processamento diferentes. Além disso, em alguns exemplos, a memória, os ciclos e as exigências de atraso podem ser diferentes para camadas diferentes.

[0035] A utilização das ferramentas de codificação paralela descritas acima (por exemplo, pequenos blocos e/ou WPP) através de camadas pode ser derivada tipicamente analisando-se o conjunto de parâmetros de imagem ativo (PPS) de cada camada. No entanto, a análise de tais informações a partir de um PPS pode não ser um processo direto e pode introduzir o atraso, devido ao conteúdo de um PPS ativo ser conhecido apenas após o processo de ativação ter começado (por exemplo, enquanto as imagens de decodificação estão dentro de uma camada particular). Adicionalmente, encontrar todos dentre os PPSs relevantes para uma sequência de vídeo codificada e analisar os PPSs pode exigir esforços de análise complicados computacionalmente.

[0036] Outros problemas potenciais com o

projeto de HEVC e as extensões relevantes (por exemplo, tais como MV-HEVC, 3D-HEVC e/ou SHVC) é que quando pequenos blocos são usados através de camadas (por exemplo, em mais que uma camada de dados de vídeo de múltiplas camadas) para paralelismo, filtros de malha (*loop filters*) podem ser aplicados através de limites de tile. Um filtro de malha pode incluir um filtro de desbloqueio ou outro filtro que é usado no processo de codificação de vídeo para melhorar a qualidade de vídeo. Independente do potencial para aprimorar a qualidade de vídeo, que segue um filtro de malha a ser aplicado através de um limite de tile pode afetar o paralelismo e introduzir um atraso. Por exemplo, se a filtragem de loop for aplicada através de pequenos blocos, um bloco de um tile particular pode ser alterado devido à filtragem de loop, que pode causar problemas para realizar a previsão de intercamada que usa tal tile.

[0037] Esta revelação descreve técnicas para sinalizar informações relacionadas ao processamento paralelo. As técnicas desta revelação podem permitir que um decodificador de vídeo determine as configurações de pequenos blocos e/ou uma configuração de WPP através de camadas de uma sequência de vídeo codificada de um ponto relativamente precoce no processo de decodificação da sequência de vídeo codificada. Além disso, as técnicas desta revelação podem permitir que um decodificador de vídeo determina a possibilidade de os filtros de malha poderem ser aplicados através de limites de tile de um ponto relativamente precoce no processo de decodificação. A determinação de tais informações relativamente cedo no processo de decodificação pode, em alguns exemplos, permitir que o decodificador de vídeo atribua de modo mais eficaz os recursos de sistema e reduza o atraso (por exemplo, em relação a fornece tais informações em outra localização, tal

como um PPS, conforme descrito acima). Isto é, as informações de processamento paralelo permitem que o decodificador de vídeo determine a possibilidade de as ferramentas de codificação paralela serem usadas, e se sim, a maneira na qual as ferramentas de codificação paralelas são implantadas.

[0038] Em alguns exemplos, as informações de processamento paralelo podem ser fornecidas em um conjunto de parâmetros de vídeo (VPS), conforme descrito em maiores detalhes abaixo. Em outros exemplos, as informações de processamento paralelo podem ser fornecidas nas informações de capacidade de uso de vídeo de VPS (VUI), conforme descrito em maiores detalhes abaixo. O VPS é decodificado relativamente cedo no processo de decodificação (por exemplo, em relação ao PPS). Desse modo, de acordo com os aspectos desta revelação e conforme descrito em maiores detalhes abaixo, as informações de processamento paralelo do VPS (ou VPS VUI) podem fornecer uma indicação precoce das informações de processamento paralelo que são usadas durante a codificação. Isto é, as informações de processamento paralelo do VPS podem complementar outras informações de processamento paralelo que podem ser fornecidas em um PPS e usadas durante a codificação. Por exemplo, as informações de processamento paralelo do VPS (ou VPS VUI) podem fornecer uma indicação precoce das informações de processamento paralelo que são incluídas em um PPS. Dessa maneira, conforme descrito em maiores detalhes abaixo, um decodificador de vídeo pode decodificar as informações de processamento paralelo a partir do VPS (ou VPS VUI), determinar a possibilidade (e/ou como) de as ferramentas de codificação de processamento paralelo serem usadas no fluxo de bits de múltiplas camadas, e alocar recursos de decodificação com base nas informações do processamento paralelo a partir do

VPS. O decodificador de vídeo pode, então, começar a decodificação do fluxo de bits de múltiplas camadas com o uso das informações de processamento paralelo de um PPS que foi indicado pelo VPS.

[0039] De acordo com os aspectos desta revelação, um decodificador de vídeo pode ser configurado para analisar um ou mais elementos de sintaxe correspondente aos dados de vídeo, em que os um ou mais elementos de sintaxe indicam uma ou mais dentre uma configuração de pequenos blocos, uma configuração de filtro de malha e uma configuração de processamento paralelo, e realizar a decodificação paralela dos dados de vídeo de acordo com os um ou mais elementos de sintaxe.

[0040] A Figura 1 é um diagrama de blocos que ilustra um sistema de criptação e decodificação de vídeo exemplificativo 10 que pode utilizar as técnicas para o processamento de vídeo paralelo descritas nesta revelação. Conforme mostrado na Figura 1, o sistema 10 inclui um dispositivo de origem 12 que gera dados de vídeo criptados a serem decodificados posteriormente por um dispositivo de destino 14. O dispositivo de fonte 12 e o dispositivo de destino 14 podem compreender qualquer um dentre uma ampla faixa de dispositivos, incluindo computadores de mesa, computadores do tipo notebook (isto é, do tipo laptop), computadores do tipo tablet, decodificadores de sinais, pares de telefone, tais como, os chamados telefones "inteligentes", os chamados pads "inteligentes", televisões, câmeras, dispositivos de exibição, reprodutores de mídia digital, consoles de jogos eletrônicos, dispositivo de transmissão contínua de vídeo, ou semelhantes. Em alguns casos, o dispositivo de fonte 12 e o dispositivo de destino 14 podem ser equipados para comunicação sem fio.

[0041] O dispositivo de destino 14 pode receber

os dados de vídeo criptados a serem decodificados através de um enlace 16. O enlace 16 pode compreender qualquer tipo de mídia ou de dispositivo com capacidade para mover os dados de vídeo criptados do dispositivo de origem 12 ao dispositivo de destino 14. Em um exemplo, o enlace 16 pode compreender uma mídia de comunicação para possibilitar que um dispositivo de origem 12 transmite os dados de vídeo criptados diretamente a um dispositivo de destino 14 em tempo real. Os dados de vídeo criptados podem ser modulados de acordo com um padrão de comunicação, como um protocolo de comunicação sem fio, e transmitidos para o dispositivo de destino 14. O meio de comunicação pode compreender qualquer meio de comunicação sem fio ou com fio, como um espectro de radiofrequência (RF) ou uma ou mais linhas de transmissão físicas. A mídia de comunicação pode formar parte de uma rede com base em pacote, tal como, uma rede de área local, uma rede de longa distância ou uma rede global, tal como, a Internet. O meio de comunicação pode incluir roteadores, comutadores, estações-base ou qualquer outro equipamento que possa ser útil para facilitar a comunicação a partir do dispositivo de fonte 12 para o dispositivo de destino 14.

[0042] Alternativamente, os dados criptados podem ser emitidos a partir da interface de saída 22 para um dispositivo de armazenamento 32. De modo semelhante, os dados criptados podem ser acessados a partir do dispositivo de armazenamento 32 pela interface de entrada. O dispositivo de armazenamento 32 podem incluir qualquer um dentre uma variedade de mídias de armazenamento de dados distribuídos ou acessados localmente como um disco rígido, discos Blu-ray, DVDs, CD-ROMs, memória flash, memória volátil ou não volátil ou quaisquer outras mídias de armazenamento digital adequado para armazenar dados de vídeo criptados. Em um exemplo adicional, o dispositivo de armazenamento 32 pode

corresponder a um servidor de arquivos ou outro dispositivo de armazenamento intermediário que pode manter o vídeo criptado gerado por dispositivo de origem 12. O dispositivo de destino 14 pode acessar dados de vídeo armazenados a partir do dispositivo de armazenamento 32 através de transmissão contínua ou transferência por download. O servidor de arquivos pode ser qualquer tipo de servidor com capacidade para armazenar dados de vídeo criptados e transmitir esses dados de vídeo criptados para o dispositivo de destino 14. Os servidores de arquivo exemplificativos incluem um servidor da web (por exemplo, para um site da web), um servidor FTP, dispositivos de armazenamento anexado à rede (NAS) ou uma unidade de disco local. O dispositivo de destino 14 pode acessar os dados de vídeo criptados através de qualquer conexão de dados padrão, incluindo uma conexão de Internet. Isso pode incluir um canal sem fio (por exemplo, uma conexão Wi-Fi), uma conexão cabeada (por exemplo, DSL, modem a cabo, etc.), ou uma combinação dos dois que é adequada para cessar os dados de vídeo criptados armazenados em um servidor de arquivo. A transmissão de dados de vídeo criptados do dispositivo de armazenamento 32 pode ser uma transmissão contínua, uma transmissão por transferência por download ou uma combinação das mesmas.

[0043] As técnicas desta revelação não são necessariamente limitadas a aplicações ou definições sem fio. As técnicas podem ser aplicadas a codificação de vídeo em suporte a qualquer uma dentre uma variedade de aplicativos multimídia, tais como difusões de televisão pelo ar, transmissões de televisão por cabo, transmissões de televisão por satélite, transmissões de vídeo de transmissão contínua, por exemplo, através da Internet, criptação de vídeo digital para o armazenamento em uma mídia de armazenamento de dados, decodificação de vídeo digital

armazenado em uma mídia de armazenamento de dados, ou outras aplicações. Em alguns exemplos, o sistema 10 pode ser configurado para suportar transmissão de vídeo unidirecional ou bidirecional para suportar aplicações como difusão de vídeo, reprodução de vídeo, difusão de vídeo e/ou telefonia por vídeo.

[0044] No exemplo da Figura 1, o dispositivo de origem 12 inclui uma fonte de vídeo 18, o criptador de vídeo 20 e uma interface de saída 22. Em alguns casos, a interface de saída 22 pode incluir um modulador/demodulador (modem) e/ou um transmissor. No dispositivo de origem 12, a fonte de vídeo 18 pode incluir uma fonte tal como um dispositivo de captura de vídeo, por exemplo, uma câmera de vídeo, um arquivo de vídeo que contém vídeo capturado anteriormente, uma interface de alimentação de vídeo para receber vídeo a partir de um provedor de conteúdo de vídeo, e/ou um sistema de gráficos de computador para gerar dados de gráficos de computador como vídeo de fonte, ou uma combinação de tais fontes. Como exemplo, se a fonte de vídeo 18 for uma câmera de vídeo, o dispositivo de origem 12 e o dispositivo de destino 14 podem formar os então chamados telefones de câmera ou videofones. No entanto, as técnicas descritas nesta revelação podem ser aplicáveis à codificação de vídeo em geral e podem ser aplicadas às aplicações com fio e/ou sem fio.

[0045] O vídeo capturado, pré-capturado ou gerado por computador pode ser criptado pelo criptador de vídeo 20. Os dados de vídeo criptados podem ser transmitidos diretamente para o dispositivo de destino 14 através de interface de saída 22 de dispositivo-fonte 12. Os dados de vídeo criptados também podem (ou alternativamente) ser armazenados no dispositivo de armazenamento 32 para o acesso posterior pelo dispositivo de destino 14 ou outros

dispositivos, para decodificação e/ou reprodução.

[0046] O dispositivo de destino 14 inclui uma interface de entrada 28, um decodificador de vídeo 30 e um dispositivo de exibição 33. Em alguns casos, a interface de entrada 28 pode incluir um receptor e/ou um modem. A interface de entrada 28 de dispositivo de destino 14 recebe os dados de vídeo criptados através do enlace 16. Os dados de vídeo criptados comunicados através do enlace 16, ou fornecidos no dispositivo de armazenamento 32, podem incluir uma variedade de elementos de sintaxe gerados pelo criptador de vídeo 20 para o uso por um decodificador de vídeo, tal como decodificador de vídeo 30, na decodificação dos dados de vídeo. Tais elementos de sintaxe podem ser incluídos com dados de vídeo criptados transmitidos em uma mídia de comunicação, armazenados em uma mídia de armazenamento, ou armazenados em um servidor de arquivo.

[0047] O dispositivo de exibição 33 pode ser integrado com, ou externo ao, dispositivo de destino 14. Em alguns exemplos, o dispositivo de destino 14 pode incluir um dispositivo de exibição integrado e também pode ser configurado para realizar interface com um dispositivo de exibição externo. Em outros exemplos, o dispositivo de destino 14 pode ser um dispositivo de exibição. Em geral, o dispositivo de exibição 33 exibe os dados de vídeo decodificados para um usuário e pode compreender qualquer um dentre uma variedade de dispositivos de visores como um visor de cristal líquido (LCD), um visor de plasma, um visor de diodo emissor de luz orgânico (OLED) ou outro tipo de dispositivo de exibição.

[0048] Embora não seja mostrado na Figura 1, em alguns aspectos, o criptador de vídeo 20 e o decodificador de vídeo 30 podem ser, cada um, integrados a um criptador e decodificador de áudio, e podem incluir unidades MUX-DEMUX

apropriadas, ou outro hardware e software, para lidar com a criptação tanto de áudio quanto de vídeo em uma corrente de dados comum ou correntes de dados separadas. Caso aplicável, em alguns exemplos, as unidades MUX-DEMUX podem se conformar ao protocolo multiplexador ITU H.223, ou outros protocolos como protocolo de datagrama de usuário (UDP).

[0049] O criptador de vídeo 20 e o decodificador de vídeo 30, cada um, podem ser implantados como qualquer um dentre uma variedade de conjunto de circuitos de criptador adequado, como um ou mais microprocessadores, processadores de sinal digital (DSPs), circuitos integrados de aplicação específica (ASICs), matrizes de porta programáveis por campo (FPGAs), lógica discreta, software, hardware, firmware ou quaisquer combinações dos mesmos. Quando as técnicas são implantadas parcialmente em software, um dispositivo pode armazenar instruções para o software em um meio legível por computador não transitório adequado e executar as instruções em hardware com o uso de um ou mais processadores para realizar as técnicas desta revelação. Cada um dentre o criptador de vídeo 20 e o decodificador de vídeo 30 podem estar incluídos em um ou mais criptadores ou decodificadores, um dos quais pode ser integrado como parte de um criptador/decodificador (CODEC) combinado em um dispositivo respectivo.

[0050] Esta revelação pode, de modo geral, se referir ao criptador de vídeo 20 "que sinaliza" determinadas informações para outro dispositivo, tal como decodificador de vídeo 30. Deve ser entendido, no entanto, que o criptador de vídeo 20 pode sinalizar as informações associando-se determinados elementos de sintaxe a diversas porções criptadas de dados de vídeo. Isto é, criptador de vídeo 20 pode "sinalizar" dados armazenando-se determinados elementos de sintaxe para cabeçalhos de diversas porções criptadas de

dados de vídeo. Em alguns casos, tais elementos de sintaxe podem ser criptados e armazenados antes de serem recebidos e decodificados pelo decodificador de vídeo 30. Desse modo, o termo "sinalização" pode se referir de modo geral à comunicação de sintaxe ou de outros dados para decodificar dados de vídeo comprimidos, independente de tal comunicação ocorrer em tempo real ou próximo de tempo real ou através de uma amplitude de período de tempo, tal como pode ocorrer ao armazenar elementos de sintaxe para um meio no período de tempo de criptação, que, então, pode ser recuperado por um dispositivo de decodificação em qualquer período de tempo após ser armazenado nesse meio.

[0051] O criptador de vídeo 20 e o decodificador de vídeo 30 podem operar de acordo com um padrão de compressão de vídeo, tal como o HEVC desenvolvido pela Equipe de Colaboração Conjunta em Codificação de Vídeo (JCT-VC) de Grupo de Especialistas de Codificação de Vídeo de ITU-T (VCEG) e Grupo de Especialistas em Imagem com Movimento de ISO/IEC (MPEG). HEVC é descrita em ITU-T H.265, SÉRIE H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS, Infrastructure of Audiovisual Services— Coding of Moving Video, "High Efficiency Video Coding", abril de 2013 (doravante no presente documento, "HEVC"). Uma especificação de esboço de HEVC, e chamada de HEVC WD doravante no presente documento, é disponibilizada por http://phenix.mt-evry.fr/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1003-v1.zip.

[0052] Diversas extensões ao HEVC foram propostas. Por exemplo, a extensão de múltiplas vistas para HEVC, a saber, MV-HEVC, e outra extensão de HEVC para codificação de vídeo 3D mais avançada (3D-HEVC) são desenvolvidas por JCT-3V. Enquanto isso, a extensão de codificação de vídeo escalonável para HEVC, a saber, SHVC, é desenvolvida por JCT-VC.

[0053] Um Esboço de Trabalho recente (WD) de MV-HEVC, chamado de MV-HEVC WD5 no presente documento, é disponibilizado por http://phenix.it-sudparis.eu/jc12/doc_end_user/documents/5_Vienna/wg11/JCT3V-E1004-v6.zip. Um WD recente de 3D-HEVC, chamado de 3D-HEVC WD1 no presente documento, é disponibilizado por http://phenix.it-sudparis.eu/jc12/doc_end_user/documents/5_Vienna/wg11/JCT3V-E1001-v3.zip. A WD recente de SHVC é chamado de SHVC WD3 no presente documento, é disponibilizado por http://phenix.it-sudparis.eu/jc14/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1008-v3.zip. Outra extensão para HEVC é a extensão de Faixa de HEVC, descrita em "High Efficiency Video Coding (HEVC) Range Extensions text specification: Draft 4," JCTVC-N1005_v1, abril de 2013 (no presente documento, "JCTVC-N1005").

[0054] Os conjuntos de procedimentos desta revelação, no entanto, não se limitam a qualquer padrão de codificação particular. Consequentemente, as técnicas desta revelação podem ser aplicadas a qualquer dentre uma variedade de padrões de codificação nos quais a indicação de processamento paralelo pode ser desejável.

[0055] Em relação ao HEVC, como exemplo, os esforços de padronização tiveram como base um modelo de evolução de um dispositivo de codificação de vídeo chamado de Modelo de Teste de HEVC (HM). O HM presume um certo número de capacidades adicionais de dispositivos de vídeo de codificação em relação a dispositivos existentes de acordo com, por exemplo, ITU-T H.264/AVC. Por exemplo, enquanto o H.264 fornece nove modos de criptação intraprevisão, o HM pode fornecer até trinta e três modos de criptação de intraprevisão.

[0056] Em geral, o modelo em funcionamento do HM descreve que um quadro de vídeo ou imagem pode ser

dividido em uma sequência de blocos de árvore de codificação (CTBs) ou as maiores unidades de codificação (LCUs) que incluem tanto amostras luma quanto amostra croma. Em diversos exemplos, um CTB é um bloco de $N \times N$ de amostras (por exemplo, amostras de luma ou croma). Uma CTB tem um propósito similar ao de um macro bloco do padrão H.264. Uma fatia inclui diversas CTBs consecutivas em ordem de codificação. Um quadro de vídeo ou imagem pode ser particionado em uma ou mais fatias. Cada CTB pode ser dividida em unidades de codificação (CUs) de acordo com uma árvore quadrática. Por exemplo, uma CTB, como um nó raiz da árvore quadrática, pode ser dividida em quatro nós filhos e cada nó filho pode, por sua vez, ser um nó pai e ser dividido em outros quatro nós filhos. Um nó filho não dividido final, como um nó de folha da árvore quadrática, compreende um nó de codificação, isto é, um bloco de vídeo codificado. Os dados de sintaxe associados a um fluxo de dados codificados podem definir um número máximo de vezes que uma CTB pode ser dividida e também podem definir um tamanho mínimo dos nós de codificação.

[0057] Uma CU inclui um nó de codificação e unidades de previsão (PUs) e unidades de transformada (TUs) associadas ao nó de codificação. Um tamanho da CU corresponde de modo geral a um tamanho do nó de codificação e deve ter, tipicamente, formato quadrado. O tamanho da CU pode variar de 8×8 pixels até o tamanho do CTB com um máximo de 64×64 pixels ou maior. 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, a partição da CU em uma ou mais PUs. Os modos de partição podem diferir entre a possibilidade de a CU ser criptada em salto ou de modo direto, criptada de modo em intraprevisão ou criptação de modo em interprevisão. As PUs podem ser particionadas para terem formato não quadrado. Os dados de sintaxe associados a uma CU também

podem descrever, por exemplo, a partição da CU em uma ou mais TUs de acordo com uma árvore quadrática. Uma TU pode ter formato quadrado ou não quadrado.

[0058] O padrão HEVC prevê transformações de acordo com TUs, que podem ser diferentes para diferentes CUs. As TUs são dimensionadas tipicamente com base no tamanho de PUs dentro de uma CU dada definida por uma LCU particionada, embora esse possa nem sempre ser o caso. As TUs têm tipicamente o mesmo tamanho ou são menores que as PUs. Em alguns exemplos, as amostras residuais correspondentes a uma CU podem ser subdivididas em unidades menores com o uso de uma estrutura de árvore quadrática conhecida como "árvore quadrática residual" (RQT). Os nós de folha da RQT podem ser chamados de unidades de transformada (TUs). Os valores de diferença de pixel associados às TUs podem ser transformados para produzir o coeficiente de transformada, que pode ser quantizado.

[0059] Em geral, uma PU inclui dados relacionados ao processo de previsão. Por exemplo, quando a PU é criptada em intermodos, a PU pode incluir dados que descrevem um modo intraprevisão para a PU. Como outro exemplo, quando a PU for criptada em intermodo, a PU pode incluir dados que definem um vetor de movimento para a PU. Os dados que definem o vetor de movimento para um PU podem descrever, por exemplo, um componente horizontal do vetor de movimento, um componente vertical do vetor de movimento, uma resolução para o vetor de movimento (por exemplo, precisão de um quarto de pixel ou precisão de um oitavo de pixel), uma imagem de referência para a qual o vetor de movimento aponta e/ou uma lista de imagens de referência (por exemplo, Lista 0, Lista 1 ou Lista C) para o vetor de movimento.

[0060] Em geral, uma TU é usada para os processos de transformada e de quantização. Uma determinada

CU que tem uma ou mais PUs também podem incluir uma ou mais unidades de transformada (TUs). Após a previsão, o criptador de vídeo 20 pode calcular os valores residuais a partir do bloco de vídeo identificado pelo nó de codificação de acordo com a PU. O nó de codificação é, então, atualizado para referência aos valores residuais em vez de o bloco de vídeo original. Os valores residuais compreendem valores de diferença de pixel que podem ser transformados em coeficientes de transformada, quantizados e digitalizados com o uso das informações de transformadas e de outras transformadas especificados nas TUs para produzir coeficientes de transformada em série para codificação por entropia. O nó de codificação pode ser novamente substituído para se referir a esses coeficientes de transformada em série. Esta revelação usa tipicamente o termo "bloco de vídeo" para se referir a um ou mais dentro um CTB, LCU ou uma CU, que inclui um nó de codificação e PUs e TUs.

[0061] Uma sequência de vídeo inclui tipicamente uma série de quadros ou imagens de vídeo. Um grupo de imagens (GOP) compreende de modo geral uma série de uma ou mais dentro as imagens de vídeo. Um GOP pode incluir dados de sintaxe em um cabeçalho do GOP, um cabeçalho de uma ou mais das imagens ou, em outro lugar, que descreve um certo número de imagens incluídas no GOP. Cada fatia de uma imagem pode incluir uma fatia de dados de sintaxe que descrevem um modo de criptação para a fatia respectiva. O criptador de vídeo 20 opera tipicamente em blocos de vídeo dentro de fatias de vídeo individuais a fim de criptar 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 variados e podem diferir em tamanho de acordo com um padrão de codificação especificado.

[0062] Como exemplo, o HM sustenta uma previsão

em vários tamanhos de PU. Admitindo-se que o tamanho de uma CU particular seja $2N \times 2N$, o HM sustenta intraprevisão em tamanhos de PU de $2N \times 2N$ ou $N \times N$ e interprevisão em tamanhos simétricos de PU de $2N \times 2N$, $2N \times N$, $N \times 2N$ ou $N \times N$. O HM também sustenta a partição assimétrico para interprevisão em tamanhos de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ e $nR \times 2N$. Na partição assimétrica, uma direção de uma CU não é particionada, enquanto a outra direção é particionada em 25% e 75%. A porção da CU correspondente aos 25% de partição é indicada por um "n" seguida por uma indicação de "Cima", "Baixo", "Esquerda" ou "Direita". Dessa forma, por exemplo, " $2N \times nU$ " se refere a um CU de $2N \times 2N$ que é particionado horizontalmente com um PU de $2N \times 0.5N$ no topo e um PU de $2N \times 1.5N$ no fundo.

[0063] Nesta revelação, " $N \times N$ " e "N por N" podem ser usados de modo intercambiável para se referir às dimensões de pixel de um bloco de vídeo em termos de dimensões horizontal e vertical, por exemplo, 16×16 pixels ou 16 por 16 pixels. Em geral, um bloco de 16×16 terá 16 pixels em uma direção vertical ($y = 16$) e 16 pixels em uma direção horizontal ($x = 16$). De modo semelhante, um bloco $N \times N$ geralmente tem N pixels em uma direção vertical e N pixels em uma direção horizontal, em que N representa um valor de número inteiro não negativo. Os pixels em um bloco podem ser representados em fileiras e colunas. Além disso, os blocos não necessariamente têm a mesma quantidade de pixels tanto na direção horizontal quanto na direção vertical. Por exemplo, os blocos podem compreender Pixels $N \times M$, em que M não é necessariamente igual a N.

[0064] Após codificação de intrapredição e interpredição que usa as PUs de uma CU, o criptador de vídeo 20 pode calcular dados residuais aos quais as transformadas especificadas pelas TUs da CU são aplicadas. Os dados residuais podem corresponder a diferenças de pixel entre

pixels da imagem não criptada e valores de previsão correspondentes às CUs. O criptador de vídeo 20 pode formar as TUs que incluem os dados residuais para a CU, e, então, transformar os dados residuais para produzir o coeficiente de transformada.

[0065] Após quaisquer transformações para produzir os coeficientes de transformada, o criptador de vídeo 20 pode realizar a quantização do coeficiente de transformada. A quantização, em geral, se refere a um processo no qual os coeficientes de transformada são quantizados para reduzir possivelmente a quantidade de dados usados para representar os coeficientes, fornecendo compressão adicional. O processo de quantização pode reduzir a profundidade de bits associada a alguns ou todos os coeficientes. Por exemplo, um valor n-bit pode ser arredondado para baixo para um valor m-bit durante a quantização, em que n é maior que m.

[0066] Em alguns exemplos, o criptador de vídeo 20 pode utilizar uma ordem de varredura predefinida para varrer os coeficientes de transformada quantizados para produzir um vetor serializado que pode ser criptado por entropia. Em outros exemplos, o criptador de vídeo 20 pode realizar uma varredura adaptativa. Após varrer o coeficiente de transformada quantizado para formar um vetor monodimensional, o criptador de vídeo 20 pode criptar por entropia o vetor monodimensional, por exemplo, de acordo com a codificação de comprimento variável adaptativa a contexto (CAVLC), codificação aritmética binária adaptativa a contexto (CABAC), codificação aritmética binária adaptativa a contexto com base em sintaxe (SBAC), codificação por Entropia de Partição de Intervalo de Probabilidade (PIPE) ou outra metodologia de criptação de entropia. O criptador de vídeo 20 também pode criptar por entropia os elementos de

sintaxe associados aos dados de vídeo criptados para o uso pelo decodificador de vídeo 30 na decodificação dos dados de vídeo.

[0067] Conforme observado acima, as técnicas desta revelação podem ser usadas para fornecer indicação precoce de ferramentas de codificação paralelas, tais como codificação de dados de vídeo que usam pequenos blocos e/ou WPP. Por exemplo, uma configuração de fluxo de bits típica pode ser conforme o seguinte: Conjunto de Parâmetros de Vídeo (VPS), extensão de VPS. Conjunto de Parâmetros de Sequência (SPS), extensão SPS, Conjunto de Parâmetros de imagem (PPS), extensão de PPS, cabeçalho de fatia, carga de vídeo. Em alguns exemplos, as ferramentas de codificação paralelas podem ser indicadas no PPS e/ou na extensão de PPS.

[0068] As técnicas desta revelação podem, em alguns exemplos, ser usadas para fornecer uma indicação precoce a um decodificador de vídeo da possibilidade de as ferramentas de codificação paralelas serem usadas dentro de camadas de um fluxo de bits de múltiplas camadas, por exemplo, no VPS, recebidas pelo decodificador de vídeo. Por exemplo, as informações de processamento paralelo do VPS podem complementar outras informações de processamento paralelo que podem ser fornecidas em um PPS e usadas durante a codificação, portanto, fornecendo uma indicação precoce de informações de processamento paralelo que são incluídas no PPS (por exemplo, devido ao fato das informações do VPS serem decodificadas antes do PPS).

[0069] De acordo com os aspectos desta revelação, as técnicas podem permitir um decodificador de vídeo, tal como decodificador de vídeo 30, para atribuir eficazmente recursos de decodificação para decodificação dos dados de vídeo de múltiplas camadas e redução de atraso decodificação geral. Por exemplo, o criptador de vídeo 20

pode criptar (e o decodificador de vídeo 30 pode decodificar) os dados de um VPS e/ou VPS VUI que indicam uma configuração de pequenos blocos para as camadas do fluxo de bits de múltiplas camadas e uma configuração de processamento paralelo para as camadas do fluxo de bits de múltiplas camadas. Em particular, com referência à configuração de processamento paralelo, os dados podem indicar a possibilidade de a sincronização de processamento paralelo ser realizada (por exemplo, a sincronização de codificação por entropia, conforme realizada em WPP).

[0070] De modo geral, um VPS pode incluir dados que descreve as características gerais de uma sequência de vídeo codificada, que inclui dependências entre subcamadas. Um propósito do VPS pode ser habilitar a extensibilidade compatível de um padrão particular em termos de sinalização na camada de sistemas. Um VPS deve ser, tipicamente, incluída em um fluxo de bits de múltiplas camadas para o fluxo de bits a ser decodificado. Um VPS pode, adicional ou alternativamente, incluir dados que definem informações de capacidade de uso de vídeo (VUI), tais informações de representação de vídeo, parâmetros de decodificador de referência hipotético (HRD), e/ou informações de restrição de fluxo de bits. As informações de restrição de fluxo de bits podem incluir restrições na faixa de vetor de movimento, tamanho de armazenamento temporário de imagem decodificado (DPB) (isto é, em termos de um número de imagens a serem mantidas pelo DPB), o número de quadros de reordenação (por exemplo, uma indicação de um número de quadro a ser reordenado a partir da ordem de decodificação para ordem de exibição), tamanhos codificados de blocos (por exemplo, macro blocos (MBs) ou unidades de árvore de codificação), e tamanhos codificados de imagens. Um VPS pode fornecer adicionalmente dados para uma ou mais extensões de VPS, de

modo que o VPS possa ser estendido por padrões futuros ou extensões ao próximo padrão HEVC. Nesta revelação, o acrônimo VPS pode se referir conjuntamente tanto a um VPS quanto VPS VUI, conforme aplicável.

[0071] Um exemplo VPS VUI para indicar as configurações de pequenos blocos e em WPP para cada camada é ilustrado na Tabela 1 abaixo. WPP é descrito em maiores detalhes, por exemplo, com referência às Figuras 4A a 5 abaixo. Pequenos blocos são descritos em maiores detalhes, por exemplo, com referência às Figuras 6 abaixo. De acordo com os aspectos desta revelação, o criptador de vídeo 20 pode criptar os elementos de sintaxe mostrados na Tabela 1 e incluir os elementos de sintaxe criptados em um fluxo de bits criptado. De modo semelhante, o decodificador de vídeo 30 pode analisar e decodificar os elementos de sintaxe a partir de um fluxo de bits criptado.

TABELA 1 - EXEMPLO DE VPS VUI N° 1

vps_vui() {	Descritor
...	
for(i = 0; i <= vps_max_layers_minus1; i++) {	u(1)
tiles_enabled_vps_flag[i]	
if(tiles_enabled_vps_flag[i])	
loop_filter_across_tiles_disabled_vps_flag[i]	u(1)
entropy_coding_sync_enabled_vps_flag[i]	u(1)
}	
for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < NumDirectRefLayers[layer_id_in_nuh[i]]; j++) {	
if(tiles_enabled_vps_flag[i])	
tile_boundaries_aligned_flag[i][j]	u(1)
}	
...	

[0072] O elemento de sintaxe pequenos

blocos_enabled_vps_flag[i] que é igual a um valor de 1 indica que o valor da pequenos blocos_enabled_flag é igual para 1 para cada PPS que é referido por (por exemplo, associado a) pelo menos uma imagem da i-ésima camada especificada pelo VPS. O elemento de sintaxe pequenos blocos_enabled_vps_flag[i] que é igual a um valor de 0 indica que o valor da pequenos blocos_enabled_flag é igual para 0 para cada PPS que é referido por (por exemplo, associado a) pelo menos uma imagem da i-ésima camada especificada pelo VPS.

[0073] O elemento de sintaxe pequenos blocos_enabled_flag igual a 1 especifica que há mais que um tile em cada imagem, que se refere ao PPS. O elemento de sintaxe pequenos blocos_enabled_flag que é igual a um valor de 0 especifica que há apenas um tile em cada imagem, que se refere ao PPS.

[0074] O elemento de sintaxe filtro de malha através de pequenos blocos_disabled_vps_flag[i] igual a 1 indica que o valor do filtro de malha através de pequenos blocos_enabled_flag é igual a 0 para cada PPS que é referido por (por exemplo, associado a) pelo menos uma imagem da i-ésima camada especificada pelo VPS. O elemento de sintaxe filtro de malha através de pequenos blocos_enabled_vps_flag[i] que é igual a um valor de 0 indica que a restrição acima pode ou não pode ser aplicada.

[0075] O elemento de sintaxe loop_filter_across_pequenos blocos_enabled_flag igual a 1 especifica que operações de filtragem in-loop são realizadas através de limites de tile. O elemento de sintaxe loop_filter_across_pequenos blocos_enabled_flag que é igual a um valor de 0 especifica que as operações de filtragem in-loop não são realizadas através de limites de tile. As operações de filtragem in-loop podem incluir o filtro de

desbloqueio e operações de filtro de deslocamento adaptável por amostra. Quando não estiver presente (por exemplo, o elemento de sintaxe `loop_filter_across_pequenos_blocos_enabled_flag` não estiver incluído no fluxo de bits), o valor do elemento de sintaxe filtro de malha através de pequenos blocos `enabled_flag` é inferido como igual a 1.

[0076] O elemento de sintaxe `entropy_coding_sync_enabled_vps_flag[i]` igual a 1 indica que o valor do elemento de sintaxe `entropy_coding_sync_enabled_flag` é igual para 1 para cada PPS que é referido por (por exemplo, associado a) pelo menos uma imagem da *i*-ésima camada especificada pelo VPS. O elemento de sintaxe `entropy_coding_sync_enabled_vps_flag[i]` que é igual a um valor de 0 indica que o valor do elemento de sintaxe `entropy_coding_sync_enabled_flag` é igual para 0 para cada PPS que é referido por (por exemplo, associado a) pelo menos uma imagem da *i*-ésima camada especificada pelo VPS.

[0077] O elemento de sintaxe `entropy_coding_sync_enabled_flag` igual a 1 especifica que um processo de sincronização específico para contextos variáveis é invocado antes da decodificação do primeiro bloco de árvore de codificação de uma fileira de blocos de árvore de codificação em cada tile em cada imagem que se refere ao PPS, e um processo de memorização específico para contextos variáveis é invocado após a decodificação de dois blocos de árvore de codificação de uma fileira de blocos de árvore de codificação em cada tile em cada imagem que se refere ao PPS. O elemento de sintaxe `entropy_coding_sync_enabled_flag` que é igual a um valor de 1 especifica que nenhum processo de sincronização específico para contextos variáveis deve ser invocado antes da decodificação do primeiro bloco de árvore de codificação de uma fileira de blocos de árvore de

codificação em cada tile em cada imagem que se refere ao PPS, e nenhum processo de memorização específico para contextos variáveis deve ser invocado após a decodificação de dois blocos de árvore de codificação de uma fileira de blocos de árvore de codificação em cada tile em cada imagem que se refere ao PPS.

[0078] Consequentemente, mediante o recebimento dos elementos de sintaxe incluídos no exemplo da Tabela 1 acima, o decodificador de vídeo 30 pode determinar uma configuração de pequenos blocos e/ou uma configuração de processamento paralelo para camadas de um fluxo de bits de múltiplas camadas. Por exemplo, para cada camada a partir de 0 ao número máximo de camadas (por exemplo, para ($i=0$; $i \leq vps_max_layers_minus1$; $i++$) o decodificador de vídeo 30 pode obter, a partir de um VPS de um fluxo de bits criptado, dados que indicam a possibilidade de os pequenos blocos serem habilitados em pelo menos uma imagem na camada (por exemplo, o elemento de sintaxe pequenos_blocos_enabled_vps_flag[i]), além disso, para cada camada para a qual os pequenos blocos estão habilitados (por exemplo, se (pequenos_blocos_enabled_vps_flag[i])) o decodificador de vídeo 30 também pode obter dados que indicam a possibilidade de a filtragem de loop ser aplicada através de limites de tile, ou da possibilidade de a filtragem de loop de limites de tile ser desabilitada (por exemplo, o filtro de malha através de elemento de sintaxe pequenos_blocos_disabled_vps_flag[i]).

[0079] Além disso, o decodificador de vídeo 30 também pode obter os dados que indicam uma configuração de processamento paralelo. Por exemplo, o decodificador de vídeo 30 pode obter dados que indicam a possibilidade de a sincronização de codificação por entropia ser realizada, por exemplo, através de WPP (por exemplo, a

entropy_coding_sync_enabled_vps_flag[i])). Em outros exemplos, os elementos de sintaxe que tem uma convenção de nome diferente podem ser usados para transmitir a configuração de pequenos blocos e/ou configuração de processamento paralelo,

[0080] Em outro exemplo, as técnicas desta revelação incluem a sinalização/leitura de uma indicação de alinhamento de camada transversal de pequenos blocos, e configurações de WPP e filtro de malha (por exemplo, o uso de filtro de malha através de limites de tile, loopfilter_accross_tile_boundary) em um VPS. Em um primeiro exemplo, a indicação das configurações de pequenos blocos e em WPP para cada camada pode ser feita no VPS VUI modificado conforme mostrado abaixo na Tabela 2. O criptador de vídeo 20 pode criptar os elementos de sintaxe mostrados na Tabela 2 e incluir os elementos de sintaxe criptados em um fluxo de bits criptado. De modo semelhante, o decodificador de vídeo 30 pode analisar e decodificar os elementos de sintaxe a partir de um fluxo de bits criptado.

TABELA 2 - EXEMPLO DE VPS VUI N° 2

vps_vui(){	Descritor
...	
parallel_tools_vps_idc	u(2)
...	

[0081] O elemento de sintaxe parallel_tools_vps_idc igual a 0 indica que tanto o elemento de sintaxe pequenos_blocos_enabled_flag quanto o elemento de sintaxe entropy_coding_sync_enabled_flag são iguais a 0 para as imagens. O elemento de sintaxe parallel_tools_vps_idc igual a 1 indica que o elemento de sintaxe entropy_coding_sync_enabled_flag é igual para 1 e o elemento de sintaxe pequenos_blocos_enabled_flag é igual a 0 para as

imagens em todas camadas. O elemento de sintaxe `parallel_tools_vps_idc` igual a 2 indica que o elemento de sintaxe `pequenos_blocos_enabled_flag` é igual a 1 e o elemento de sintaxe `entropy_coding_sync_enabled_flag` é igual a 0 para as imagens em todas as camadas. O elemento de sintaxe `parallel_tools_vps_idc` igual a 3 indica que o elemento de sintaxe `pequenos_blocos_enabled_flag` pode ser igual a 1 para as imagens em zero ou mais camadas e o elemento de sintaxe `entropy_coding_sync_enabled_flag` pode ser igual a 1 para as imagens em zero ou mais camadas.

[0082] As Tabelas 3 e 4 abaixo mostram exemplos adicionais de como o VPS VUI pode ser modificado para indicar a configuração de pequenos blocos e em WPP. De acordo com os aspectos desta revelação, o criptador de vídeo 20 pode criptar os elementos de sintaxe mostrados nas Tabelas 3 e 4 e incluem os elementos de sintaxe criptados em um fluxo de bits criptado. De modo semelhante, o decodificador de vídeo 30 pode analisar e decodificar os elementos de sintaxe a partir de um fluxo de bits criptado.

TABELA 3 - EXEMPLO DE VPS VUI N° 3

<code>vps_vui(){</code>	Descritor
<code>...</code>	
<code>for(i = 0; i <= vps_max_layers_minus1; i++) {</code>	
<code> parallel_tools_vps_idc[i]</code>	<code>u(2)</code>
<code>}</code>	
<code>...</code>	

[0083] O elemento de sintaxe `parallel_tools_vps_idc[i]`: que é igual a um valor de 0 indica que tanto o elemento de sintaxe `pequenos_blocos_enabled_flag` quanto o elemento de sintaxe `entropy_coding_sync_enabled_flag` são iguais a 0 para a imagem da i-ésima camada que se refere a um PPS ativo. O

elemento de sintaxe `parallel_tools_vps_idc[i]` igual a 1 indica que o elemento de sintaxe `entropy_coding_sync_enabled_flag` é igual a 1 para a imagem da *i*-ésima camada e todas as suas imagens de camada de referência diretas que se referem a um PPS ativo. O elemento de sintaxe `parallel_tools_vps_idc[i]` igual a 2 indica que o elemento de sintaxe `pequenos_blocos_enabled_flag` é igual a 1 para a imagem da *i*-ésima camada e todas as suas imagens de camada de referência diretas que se referem a um PPS ativo. O elemento de sintaxe `parallel_tools_vps_idc[i]` igual a 3 indica que, quando o elemento de sintaxe `pequenos_blocos_enabled_flag` é igual a 1 para a imagem da *i*-ésima camada, então, pelo menos uma dentre as suas camadas de referência diretas tem o elemento de sintaxe `entropy_coding_sync_enabled_flag = 1` e, quando o elemento de sintaxe `entropy_coding_sync_enabled_flag = 1` para a imagem da *i*-ésima camada, então, pelo menos uma dentre as suas camadas de referência diretas tem o elemento de sintaxe `pequenos_blocos_enabled_flag` igual a 1 ,

TABELA 4 - EXEMPLO DE VPS VUI N° 4

<code>vps_vui()</code> {	Descritor
...	
for(i = 0; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < NumDirectRefLayers[layer_id_in_nuh[i]]; j++) {	u(1)
tile_boundaries_aligned_flag[i][j]	u(1)
if(tile_boundaries_aligned_flag[i][j] == 0)	
tile_entropy_sync_coding_not_mixed_flag[i][j]	u(1)
loopfilter_across_tile_boundary_not_mixed_flag[i][j]	u(1)
}	
...	

[0084] O elemento de sintaxe `tile_entropy_sync_coding_not_mixed_flag[i][j]`: igual a 1 indica que , dentro de um CVS, quando o elemento de sintaxe `pequenos_blocos_enabled_flag` é igual a 1 para um PPS ativo

de i-ésima camada, o elemento de sintaxe `entropy_coding_sync_enabled_flag` é igual a 0 para um PPS ativo da j-ésima camada de referência direta da i-ésima camada ou quando o elemento de sintaxe `entropy_coding_sync_enabled_flag` é igual a 1 para um PPS ativo de i-ésima camada, o elemento de sintaxe `pequenos_blocos_enabled_flag` é igual a 0 para um PPS ativo da j-ésima camada de referência direta da i-ésima camada. O elemento de sintaxe `tile_entropy_sync_coding_not_mixed_flag[i][j]` que é igual a um valor de 0 indica que tal uma restrição pode ou não pode ser aplicada, em tais exemplos, um ou mais elementos de sintaxe adicionais podem ser usados para indicar a possibilidade de aplicar a restrição.

[0085] O elemento de sintaxe `loopfilter_across_tile_boundary_not_mixed_flag[i][j]`: igual a 1 indica que, dentro de um CVS, quando o elemento de sintaxe `loop_filter_across_pequenos_blocos_enabled_flag` é igual a 1 para um PPS ativo de i-ésima camada, o elemento de sintaxe `loop_filter_across_pequenos_blocos_enabled_flag` também é igual a 1 para um PPS ativo da j-ésima camada de referência direta da i-ésima camada. O elemento de sintaxe `loopfilter_across_tile_boundairy_not_mixed_flag[i][j]` que é igual a um valor de 0 indica que tal uma restrição pode ou não pode ser aplicada, em tais exemplos, um ou mais elementos de sintaxe adicionais podem ser usados para indicar a possibilidade de aplicar a restrição.

[0086] Dessa maneira, o criptador de vídeo 20 pode criptar, em um conjunto de parâmetros de vídeo (VPS) de um fluxo de bits de múltiplas camadas, pelo menos um dentre os dados que indicam uma configuração de pequenos blocos para camadas do fluxo de bits de múltiplas camadas ou dados que indicam a configuração de processamento paralelo para camadas do fluxo de bits de múltiplas camadas. De modo

semelhante, o decodificador de vídeo 30 pode decodificar, a partir de um conjunto de parâmetros de vídeo (VPS) de um fluxo de bits de múltiplas camadas, pelo menos um dentre os dados que indicam uma configuração de pequenos blocos para camadas do fluxo de bits de múltiplas camadas ou dados que indicam a configuração de processamento paralelo para camadas do fluxo de bits de múltiplas camadas.

[0087] A Figura 2 é um diagrama em bloco que ilustra um exemplo criptador de vídeo 20 que pode implantar as técnicas descritas nesta revelação, em particular, o criptador de vídeo 20 pode ser configurado para criptar os dados de um fluxo de bits de múltiplas camadas de acordo com as técnicas desta revelação. Conforme observado acima, o criptador de vídeo 20 pode ser adaptado para realizar múltiplas vistas e/ou codificação de vídeo escalonável. Por exemplo, o criptador de vídeo 20 pode ser configurado para criptar um fluxo de bits que está em conformidade com um ou mais padrões de codificação de vídeo extensões, tais como SHVC, MV-HEVC ou 3D-HEVC. No entanto, embora seja feita referência a padrões de codificação específicos, deve ser entendido que as técnicas não são específicas a qualquer padrão de codificação, e podem ser implantadas com padrões futuros e/ou padrões ainda não desenvolvidos.

[0088] No exemplo da Figura 2, criptador de vídeo 20 inclui uma unidade de partição 35, memória de dados de vídeo 37, unidade de processamento de previsão 41, memória de imagem de referência 64, somador 50, unidade de processamento de transformada 52, unidade de quantização 54 e unidade de criptação por entropia 56. A unidade de processamento de previsão 41 inclui unidade de estimativa de movimento 42 uma unidade de compensação de movimento 44 e uma unidade de processamento de intraprevisão 46. Para a reconstrução de bloco de vídeo, o criptador de vídeo 20

também inclui a unidade de quantização inversa 58, a unidade de processamento de transformada inversa 60, um somador 62, e uma memória de imagem de referência 64 (que também pode ser chamada de armazenamento temporário de imagem decodificado (DPB)), e uma unidade de filtragem 66.

[0089] A memória de dados de vídeo 37 pode armazenar dados de vídeo a serem criptados pelos componentes de criptador de vídeo 20. Os dados de vídeo armazenados na memória de dados de vídeo 37 podem ser obtidos, por exemplo, a partir da fonte de vídeo 18. A memória de imagem de referência 64 pode ser chamada de um armazenamento temporário de imagem decodificado que armazena dados de vídeo de referência para o uso na criptação de dados de vídeo pelo criptador de vídeo 20, por exemplo, em modos de intra ou intercodificação. A memória de dados de vídeo 37 e a memória de imagem de referência 64 podem ser formados por qualquer um dentre uma variedade de dispositivos de memória, tal como memória dinâmica de acesso aleatório (DRAM), que inclui DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) ou outros tipos de dispositivos de memória. A memória de dados de vídeo 37 e a memória de imagem de referência 64 podem ser fornecidos pelo menos dispositivo de memória ou dispositivos de memória separados. Em vários exemplos, a memória de dados de vídeo 37 pode ser no chip com outros componentes de criptador de vídeo 20, ou fora do chip em relação àqueles componentes.

[0090] O criptador de vídeo 20 pode realizar intra e intercodificação de blocos de vídeo dentre fatias de vídeo. A intracodificação depende da previsão espacial para reduzir ou remover a redundância espacial em vídeo dentro de um determinado quadro de vídeo ou imagem, a intercodificação depende da previsão temporal para reduzir ou remover a redundância temporal no vídeo dentro de quadro ou imagens

adjacentes de uma sequência de vídeo. O intramodo (modo I) pode se referir a qualquer um dentre os diversos modos de compressão com base espacial, intermodos, tais como previsão unidirecional (modo P) ou biprevisão (modo B), pode se referir a qualquer um dentre os diversos modos de compressão com base temporal.

[0091] Por exemplo, conforme mostrado na Figura 2, o criptador de vídeo 20 recebe dados de vídeo, e a partição unidade 35 particiona os dados em blocos de vídeo. Essa partição também pode incluir a partição em fatias, em cascata, ou outras unidades maiores, bem como a partição de bloco de vídeo, por exemplo, de acordo com uma estrutura de árvore do tipo quadtree de LCUs e CUs. O criptador de vídeo 20 ilustra, de modo geral, os componentes que criptam blocos de vídeo dentro de uma fatia de vídeo a ser criptada. A fatia pode ser dividida em múltiplos blocos de vídeo (e possivelmente em conjuntos de blocos de vídeo também chamados de cascata). A unidade de processamento de previsão 41 também pode selecionar um dentre uma pluralidade de modos de codificação possíveis, tal como um dentre uma pluralidade de modos de intracodificação ou um dentre uma pluralidade de modos de intercodificação, para o bloco de vídeo atual com base nos resultados de erro (por exemplo, taxa de codificação e o nível de distorção). A unidade de processamento de previsão 41 pode fornecer o bloco intra ou intercodificado resultante para o somador 50 para gerar os dados de bloco residual e para o somador 62 para reconstruir o bloco criptado para o uso como uma imagem de referência.

[0092] A unidade de processamento de intraprevisão 46 dentro da unidade de processamento de previsão 41 pode realizar a codificação intrapreditiva do bloco de vídeo atual em relação a um ou mais blocos próximos no mesmo quadro ou fatia como o bloco atual a ser codificado

para fornecer a compressão espacial. A unidade de estimativa de movimento 42 e a unidade de compensação de modo 44 dentro da unidade de processamento de previsão realizam a codificação interpreditiva do bloco de vídeo atual em relação a um ou mais blocos em um ou mais imagens de referência para fornecer compressão temporal.

[0093] A unidade de estimativa de movimento 42 pode ser configurada para determinar o modo de interprevisão para uma fatia de vídeo de acordo com uma padronização predeterminada para uma sequência de vídeo. A padronização predeterminada pode designar fatias de vídeo na sequência como fatias P ou fatias B. A unidade de estimativa de movimento 42 e a unidade de compensação de modo 44 podem ser altamente integradas, mas são ilustradas separadamente para propósitos de conceito. A estimativa de movimento, realizada pela unidade de estimativa de movimento 42, é o processo de geração de vetores de movimento, que estima o movimento para blocos de vídeo. Um vetor de movimento, por exemplo, pode indicar que o deslocamento de uma PU de um bloco de vídeo dentro de um quadro ou imagem de vídeo atual em relação a um bloco preditivo dentro de uma imagem de referência.

[0094] Um bloco preditivo é um bloco que se encontra em correspondência muito próxima à PU do bloco de vídeo a ser codificado, em termos de diferença de pixels, que pode ser determinado pelo somatório da diferença absoluta (SAD), somatório da diferença quadrada (SSD) ou outras medidas de diferença. Em alguns exemplos, o criptador de vídeo pode calcular valores para posições de pixel subinteiro de imagens de referência armazenadas na memória de imagem de referência 64. Por exemplo, o criptador de vídeo 20 pode interpolar valores de posições de um quarto de pixel, posições de um oitavo de pixel ou outras posições fracionadas de pixel da imagem de referência. Portanto, a unidade de

estimativa de movimento 42 pode realizar uma busca de movimento em relação a todas as posições de pixel e posições fracionadas de pixel e emitir um vetor de movimento com precisão fracionada de pixel.

[0095] A unidade de estimativa de movimento 42 calcula um vetor de movimento para uma PU de um bloco de vídeo em uma fatia intercodificada pela comparação da posição da PU em relação à posição de um bloco preditivo de uma imagem de referência. A imagem de referência pode ser selecionada a partir de uma primeira lista de imagem de referência (Lista 0) ou uma segunda lista de imagem de referência (Lista 1), cada uma das quais identifica uma ou mais imagens de referência armazenadas na memória de imagem de referência 64. A unidade de estimativa de movimento 42 envia o vetor de movimento calculado para a unidade de criptação por entropia 56 e a unidade de compensação de modo 44.

[0096] A compensação de movimento, realizada pela unidade de compensação de modo 44, pode envolver obter ou gerar o bloco preditivo com base no vetor de movimento determinado pela unidade de estimativa de movimento, realizando, possivelmente, interpolações para a precisão de pixel. Mediante o recebimento do vetor de movimento para a PU do bloco de vídeo atual, a unidade de compensação de modo 44 pode localizar o bloco preditivo para o qual o vetor de movimento aponta em uma das listas de imagens de referência. O criptador de vídeo 20 forma um bloco de vídeo residual subtraindo-se valores de pixel do bloco preditivo dos valores de pixel do bloco de vídeo atual que é codificado, o que forma a diferença de valores de pixels. Os valores de diferença de pixel formam dados residuais para o bloco, e podem incluir tanto componentes de diferença de luma quanto de croma. O somador 50 representa o componente ou os

componentes que realizam essa operação de subtração. A unidade de seleção de movimento 44 também pode gerar elementos de sintaxe associados aos blocos de vídeo e a fatia de vídeo para uso pelo decodificador de vídeo 30 na decodificação dos blocos de vídeo da fatia de vídeo.

[0097] A unidade de intraprevisão 46 pode intraprever ou calcular um bloco atual, como uma alternativa à interprevisão realizada pela unidade de estimativa de movimento 42 e a unidade de compensação de modo 44, conforme descrito acima. Em particular, a unidade de processamento de intraprevisão 46 pode determinar um modo de intraprevisão para usar para criptar um bloco atual. Em alguns exemplos, a unidade de processamento de intraprevisão 46 pode criptar um bloco atual com o uso de vários modos intraprevisões, por exemplo, durante passos de criptação separados, e a unidade de processamento de intraprevisão 46 (ou unidade de seleção de modo 40, em alguns exemplos) pode selecionar um modo de intraprevisão apropriado para usar a partir dos modos testados. Por exemplo, a unidade de intraprevisão 46 pode calcular valores de distorção de taxa com o uso de uma análise de distorção de taxa para os vários modos de intraprevisão testados, e selecionar o modo de intraprevisão que tem as melhores características de distorção de taxa entre os modos testados. A análise de distorção de taxa geralmente determina uma quantidade de distorção (ou erro) entre um bloco criptado e um bloco original não criptado que foi criptado para produzir o bloco criptado, bem como uma taxa de bits (isto é, vários bits) usados para produzir o bloco criptado. A unidade de processamento de intraprevisão 46 pode calcular razões a partir de distorções e taxas para os vários blocos criptados para determinar qual modo de intraprevisão exibe o melhor valor de distorção de taxa para o bloco.

[0098] Em qualquer caso, após selecionar um modo de intraprevisão para um bloco, a unidade de processamento de intraprevisão 46 pode fornecer informações que indicam o modo de intraprevisão selecionado para o bloco em relação à unidade de criptação por entropia 56. A unidade de criptação por entropia 56 pode criptar a indicação de informações do modo de intraprevisão selecionada de acordo com as técnicas desta revelação. O criptador de vídeo 20 pode incluir nos dados de configuração de fluxo de bits transmitidos, que podem incluir uma pluralidade de tabelas de índices de modo de intraprevisão e uma pluralidade de tabelas de índices de modo de intraprevisão modificados (também chamadas de tabelas de mapeamento de palavras-código), as definições de criptação de contextos para vários blocos, e indicações de um modo de intraprevisão mais provável, uma tabela de índices de modo de intraprevisão e uma tabela de índices de modo de intraprevisão modificados para usar para cada um dos contextos.

[0099] Após a unidade de processamento de previsão 41 gerar o bloco preditivo para o bloco de vídeo atual através da interprevisão ou da intraprevisão, o criptador de vídeo 20 forma um bloco de vídeo residual subtraindo-se o bloco preditivo a partir do bloco de vídeo atual. Os dados de vídeo residuais no bloco residual podem ser incluídos em uma ou mais TUs e aplicados à unidade de processamento de transformada 52. A unidade de processamento de transformada 52 transforma os dados de vídeo residuais em coeficientes de transformada residuais com o uso de uma transformada, tal como uma transformada de cosseno distinta (DCT) ou uma transformada conceitualmente semelhante. A unidade de processamento de transformada 52 pode converter os dados de vídeo residuais a partir de um domínio de pixel a um domínio de transformada, tal como um domínio de

frequência.

[00100] A unidade de processamento de transformada 52 pode enviar o coeficiente de transformada resultante para a unidade de quantização 54. A unidade de quantização 54 quantiza o coeficiente de transformada para reduzir ainda mais a taxa de bits. O processo de quantização pode reduzir a profundidade de bits associada a alguns ou todos os coeficientes. O grau de quantização pode ser modificado ajustando-se um parâmetro de quantização. Em alguns exemplos, a unidade de quantização 54, então, pode realizar uma varredura da matriz, incluindo o coeficiente de transformada quantizado. Alternativamente, a unidade de criptação por entropia 56 pode realizar a varredura.

[00101] Após a quantização, a unidade de criptação por entropia 56 cripta por entropia o coeficiente de transformada quantizado. Por exemplo, uma unidade de criptação por entropia 56 pode realizar codificação de comprimento variável adaptativa a contexto (CAVLC), codificação aritmética binária adaptativa a contexto (CABAC), codificação aritmética binária adaptativa a contexto com base em sintaxe (SBAC), codificação por entropia de partição de Intervalo de Probabilidade (PIPE) ou outra metodologia ou técnica de criptação por entropia. Após a criptação por entropia pela unidade de criptação por entropia 56, o fluxo de bits criptado pode ser transmitido para o decodificador de vídeo 30 ou arquivado para transmissão ou recuperação posterior pelo decodificador de vídeo. A unidade de criptação por entropia 56 também pode criptar por entropia os vetores de movimento e os outros elementos de sintaxe para a fatia de vídeo atual é codificada.

[00102] A unidade de quantização regredir 58 e a unidade de processamento de transformada regredir 60 aplicam quantização regredir e transformada regredir,

respectivamente, para reconstruir o bloco residual no domínio de pixels, para uso posterior como um bloco de referência de uma imagem de referência. A unidade de compensação de movimento 44 pode calcular um bloco de referência adicionando-se o bloco residual a um bloco preditivo de uma dentre as imagens de referência em uma dentre as listas de imagem de referência 64. A unidade de compensação de movimento 44 também pode aplicar um ou mais filtros de interpolação ao bloco residual reconstruído para calcular valores de pixel subinteiros para uso na estimativa de movimento. O somador 62 adiciona o bloco residual reconstruído ao bloco de previsão compensado de movimento produzido pela unidade de compensação de movimento 44 para produzir um bloco de referência para o armazenamento na memória de imagem de referência 64. O bloco de referência pode ser usado pela unidade de estimativa de movimento 42 e pela unidade de compensação de movimento 44 como um bloco de referência para interpretar um bloco em um quadro de vídeo subsequente ou imagem.

[00103] A unidade de filtragem 66 pode realizar uma variedade de processos de filtragem. Por exemplo, a unidade de filtragem 66 pode realizar a filtragem de desbloqueio. Isto é, unidade de filtragem 66 pode receber uma pluralidade de blocos de vídeo reconstruídos que formam uma fatia ou um quadro de limites de bloco de vídeo e de filtro reconstruídos para remover artefatos com qualidade de bloco de uma fatia ou quadro. Em um exemplo, a unidade de filtragem 66 avalia a chamada "força de limite" de um bloco de vídeo. Com base na força de limite de um bloco de vídeo, pixels de borda de um bloco de vídeo podem ser filtrados com referência aos pixels de borda de um bloco de vídeo adjacente de modo que as transições de um bloco de vídeo para outro sejam mais difíceis para um visualizador perceber. Em alguns

exemplos, as variáveis usadas por um filtro de desbloqueio podem ser derivadas a partir de blocos de vídeo reconstruídos sem uma comparação de blocos de vídeo reconstruídos com blocos de vídeo de fonte original. Desse modo, o criptador de vídeo 20 e o decodificador de vídeo 30 (Figura 3) podem, cada uma, ser programados para realizar o mesmo processo de desbloqueio nos blocos de vídeo reconstruídos com informações adicionais mínimas com relação ao original quadro de vídeo codificado no fluxo de bits. No entanto, em alguns casos, a unidade de filtragem 66 pode incluir elementos de sintaxe no fluxo de bits para indicar a possibilidade de o desbloqueio ser realizado e/ou da possibilidade de um tipo particular de modos de desbloqueio ser realizado. Em outros exemplos, a unidade de filtragem 66 pode aplicar um número de filtros adicionais e/ou alternativos, tais como filtros ou outros filtros de deslocamento adaptável de amostra (SAO).

[00104] De acordo com os aspectos desta revelação, o criptador de vídeo 20 pode ser configurado para criptar dados que indicam uma configuração de pequenos blocos e/ou uma configuração de processamento paralelo para camadas de um fluxo de bits de múltiplas camadas. Por exemplo, o criptador de vídeo 20 pode ser configurado para criptar os dados de um VPS e/ou VPS VUI que indicam uma configuração de pequenos blocos para as camadas do fluxo de bits de múltiplas camadas e uma configuração de processamento paralelo para as camadas do fluxo de bits de múltiplas camadas. Em particular, com referência à configuração de processamento paralelo, os dados podem indicar a possibilidade de a sincronização de processamento paralelo ser realizada (por exemplo, a sincronização de codificação por entropia, conforme realizada em WPP).

[00105] Em alguns exemplos, os dados que indicam

a configuração de pequenos blocos e/ou a configuração de processamento paralelo podem incluir um número de elementos de sintaxe. Por exemplo, para cada camada de dados de vídeo no fluxo de bits de múltiplas camadas, o criptador de vídeo 20 pode criptar um ou mais elementos de sintaxe que indicam a possibilidade de os pequenos blocos estarem habilitados em pelo menos uma imagem na camada (por exemplo, tal como elemento de sintaxe `pequenos_blocos_enabled_vps_flag[i]` mostrado no exemplo da Tabela 1 acima). Além disso, para cada camada para a qual pequenos blocos são habilitados, o criptador de vídeo 20 também pode criptar um ou mais elementos de sintaxe que indicam a possibilidade de a filtragem de loop ser aplicada através de limites de tile, ou a possibilidade de a filtragem de loop através de limites de tile ser desabilitada (por exemplo, tal como o elemento de sintaxe `loop_filter_across_pequenos_blocos_disabled_vps_flag[i]` mostrado no exemplo da Tabela 1 acima). Em alguns exemplos, o criptador de vídeo 20 também pode criptar um ou mais elementos de sintaxe que indicam uma configuração de processamento paralelo. Por exemplo, criptador de vídeo 20 pode criptar um ou mais elementos de sintaxe que indicam a possibilidade de a sincronização de codificação por entropia ser realizada, por exemplo, através WPP (por exemplo, tal como o elemento de sintaxe `entropy_coding_sync_enabled_vps_flag[i]` mostrado no exemplo da Tabela 1 acima).

[00106] A figura 3 é um diagrama de blocos que ilustra um decodificador de vídeo exemplificativo 30 que pode implantar as técnicas descritas nesta revelação. Em particular, o decodificador de vídeo pode ser configurado para realizar o processamento paralelo de frente de onda de dados de vídeo de acordo com técnicas desta revelação. Conforme observado acima, decodificador de vídeo 30 pode ser

adaptado para realizar codificação de múltiplas vistas e/ou de vídeo escalonável. Por exemplo, o decodificador de vídeo 30 pode ser configurado para decodificar um fluxo de bits que está em conformidade com um ou mais padrões de codificação de vídeo extensões, tais como SHVC, MV-HEVC ou 3D-HEVC. No entanto, embora seja feita referência a padrões de codificação específicos, deve ser entendido que as técnicas não são específicas a qualquer padrão de codificação, e podem ser implantadas com padrões futuros e/ou padrões ainda não desenvolvidos.

[00107] No exemplo da Figura 3, decodificador de vídeo 30 inclui memória de dados de vídeo 79, uma unidade de decodificação por entropia 80, unidade de processamento de previsão 81, unidade de quantização inversa 86, unidade de processamento de transformação inversa 88, somador 90, memória de imagem de referência e DPB 94. A unidade de processamento de previsão 81 inclui uma unidade de compensação de movimento 82 e uma unidade de processamento de intraprevisão 84. O decodificador de vídeo 30 pode, em alguns exemplos, realizar um passo de decodificação geralmente recíproco ao passo de criptação descrito em relação ao criptador de vídeo 20 a partir da Figura 2.

[00108] A memória de dados de vídeo 79 pode armazenar dados de vídeo, tal como um fluxo de bits de vídeo criptado, a ser decodificado pelos componentes de decodificador de vídeo 30. Os dados de vídeo armazenados na memória de dados de vídeo 79 podem ser obtidos, por exemplo, a partir do meio legível por computador, por exemplo, a partir de uma fonte de vídeo local, tal como uma câmera, através de comunicação de rede sem fio ou com fio de dados de vídeo ou através do acesso de meio de armazenamento de dados físicos. A memória de dados de vídeo 79 pode formar um armazenamento temporário de imagem criptada (CPB) que

armazena dados de vídeo criptados a partir de um fluxo de bits de vídeo codificado. A memória de imagem de referência 92 pode ser chamada de um armazenamento temporário de imagem decodificado que armazena dados de vídeo de referência para o uso na decodificação de dados de vídeo pelo decodificador de vídeo 30, por exemplo, em modos de intra ou intercodificação. A memória de dados de vídeo 79 e a memória de imagem de referência 92 podem ser formados por qualquer um dentre uma variedade de dispositivos de memória, tal como memória dinâmica de acesso aleatório (DRAM), que inclui DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) ou outros tipos de dispositivos de memória. A memória de dados de vídeo 79 e a memória de imagem de referência 92 podem ser fornecidos pelo menos dispositivo de memória ou dispositivos de memória separados. Em vários exemplos, a memória de dados de vídeo 79 pode ser no chip com outros componentes de decodificador de vídeo 30, ou fora do chip em relação àqueles componentes.

[00109] Durante o processo de decodificação, o decodificador de vídeo 30 recebe um fluxo de bits de vídeo criptado que representa blocos de vídeo de uma fatia de vídeo criptada e elementos de sintaxe associados provenientes do criptador de vídeo 20. A unidade de decodificação por entropia 80 do decodificador de vídeo 30 decodifica por entropia o fluxo de bits para gerar coeficientes quantizados, vetores de movimento e outros elementos de sintaxe. A unidade de decodificação por entropia 80 encaminha os vetores de movimento e outros elementos de sintaxe para a unidade de processamento de previsão 81. O decodificador de vídeo 30 pode receber os elementos de sintaxe no nível de fatia de vídeo e/ou no nível de bloco de vídeo.

[00110] Quando a fatia de vídeo é codificada como uma fatia intracodificada (I), unidade de processamento de

intraprevisão 84 da unidade de processo de previsão 81 pode gerar dados de previsão para um bloco de vídeo da fatia de vídeo atual com base em um modo de intraprevisão sinalizado e dados provenientes de blocos decodificados anteriormente do quadro ou imagem atual. Quando o quadro de vídeo for codificado como uma fatia intercodificada (por exemplo, B, P ou GPB), unidade de compensação de movimento, a unidade de processamento de previsão 81 produz blocos preditivos para um bloco de vídeo da fatia de vídeo atual com base nos vetores de movimento e outros elementos de sintaxe recebidos a partir da unidade de decodificação e entropia 80. 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 quadro de referência, a Lista 0 e a Lista 1, com o uso de técnicas de construção padrão com base em imagens de referência armazenadas na memória de imagem de referência 92.

[00111] A unidade de compensação de movimento 82 determina informações de previsão para um bloco de vídeo da fatia de vídeo atual analisando-se os vetores de movimento e outros elementos de sintaxe, e usa as informações de previsão para produzir os blocos preditivos para o bloco de vídeo atual em decodificação. Por exemplo, a unidade de compensação de modo 82 usa uma parte dos elementos de sintaxe recebidos para determinar um modo de previsão (por exemplo, intra ou interprevisão) usado para codificar os blocos de vídeo da fatia de vídeo, um tipo de fatia de interprevisão (por exemplo, fatia B ou fatia P), informações de construção para uma ou mais dentre as listas de imagens de referência para a fatia, vetores de movimento para cada bloco de vídeo intercriptado da fatia, situações de interprevisão para cada bloco de vídeo intercodificado da fatia e outras informações

para decodificar os blocos de vídeo na fatia de vídeo atual.

[00112] A unidade de compensação de movimento 82 também pode realizar interpolação com base em filtros de interpolação. A unidade de compensação de movimento 82 pode usar filtros de interpolação conforme usados pelo criptador de vídeo 20 durante a criptação dos blocos de vídeo para calcular valores interpolados para pixels subinteiros de blocos de referência, nesse caso, a unidade de compensação de movimento 82 pode determinar os filtros de interpolação usados pelo criptador de vídeo 20 a partir dos elementos de sintaxe recebidos e usar os filtros de interpolação para produzir blocos preditivos.

[00113] A unidade de quantização inversa 86 quantiza inversamente, por exemplo, de-quantiza, o coeficiente de transformada quantizado fornecido no fluxo de bits e decodificado pela unidade de decodificação por entropia 80. O processo de quantização inversa pode incluir o uso de um parâmetro de quantização calculado pelo criptador de vídeo 20 para cada bloco de vídeo na fatia de vídeo para determinar um grau de quantização e, de modo similar, um grau de quantização inversa que deve ser aplicado. A unidade de processamento de transformada inversa 88 aplica uma transformada inversa, por exemplo, uma DCT inversa, uma transformada de número inteiro inversa ou um processo de transformada inversa conceitualmente similar, ao coeficiente de transformada a fim de produzir blocos residuais no domínio de pixels.

[00114] Após a unidade de compensação de movimento 82 gerar o bloco preditivo para o bloco de vídeo atual com base nos vetores de movimento e outros elementos de sintaxe, o decodificador de vídeo 30 forma um bloco de vídeo decodificado somando-se os blocos residuais provenientes da unidade de processamento de transformada

inversa 88 com os blocos preditivos correspondentes gerados pela unidade de compensação de movimento 82. O somador 90 representa o componente ou os componentes que realizam essa operação de soma.

[00115] A unidade de filtragem 94 pode, em alguns exemplos, ser configurado de modo semelhante à unidade de filtragem 66 do criptador de vídeo 20 (Figura 2). Por exemplo, a unidade de filtragem 94 pode ser configurada para realizar o desbloqueio, SAO, ou outras operações de filtragem ao decodificar ou reconstruir dados de vídeo a partir de um fluxo de bits criptado. Os blocos de vídeo decodificados em um quadro ou imagem dados são, então, armazenados na memória de imagem de referência 92, que armazena imagens de referência usadas para a composição de movimento subsequente. A memória de imagem de referência 92 também armazena o vídeo decodificado para apresentação posterior em um dispositivo de exibição, como o dispositivo de exibição 33 da Figura 1.

[00116] De acordo com os aspectos desta revelação, o decodificador de vídeo 30 pode ser configurado para decodificar dados que indicam uma configuração de pequenos blocos e/ou uma configuração de processamento paralelo para camadas de um fluxo de bits de múltiplas camadas. Por exemplo, o decodificador de vídeo 30 pode ser configurado para decodificar os dados de um VPS e/ou VPS VUI que indicam uma configuração de pequenos blocos para as camadas do fluxo de bits de múltiplas camadas e uma configuração de processamento paralelo para as camadas do fluxo de bits de múltiplas camadas. Em particular, com referência à configuração de processamento paralelo, os dados podem indicar a possibilidade de a sincronização de processamento paralelo ser realizada (por exemplo, a sincronização de codificação por entropia, conforme

realizada em WPP).

[00117] Em alguns exemplos, os dados que indicam a configuração de pequenos blocos e/ou a configuração de processamento paralelo podem incluir um número de elementos de sintaxe. Por exemplo, para cada camada de dados de vídeo no fluxo de bits de múltiplas camadas, o decodificador de vídeo 30 pode decodificar um ou mais elementos de sintaxe que indicam a possibilidade de os pequenos blocos estarem habilitados em pelo menos uma imagem na camada (por exemplo, tal como elemento de sintaxe pequenos_blocos_enabled_vps_flag[i] mostrado no exemplo da Tabela 1 acima). Além disso, para cada camada para a qual pequenos blocos são permitidos, o decodificador de vídeo 30 também pode decodificar um ou mais elementos de sintaxe que indicam a possibilidade de a filtragem de loop ser aplicada através de limites de tile, ou a possibilidade de a filtragem de loop através de limites de tile ser desabilitada (por exemplo, tal como o elemento de sintaxe loop_filter_across_pequenos_blocos_disabled_vps_flag[i] mostrado no exemplo da Tabela 1 acima). Em alguns exemplos, o decodificador de vídeo 30 também pode decodificar um ou mais elementos de sintaxe que indicam uma configuração de processamento paralelo. Por exemplo, decodificador de vídeo 30 pode decodificar um ou mais elementos de sintaxe que indicam a possibilidade de a sincronização de codificação por entropia ser realizada, por exemplo, através WPP (por exemplo, tal como o elemento de sintaxe entropy_coding_sync_enabled_vps_flag[i] mostrado no exemplo da Tabela 1 acima).

[00118] De acordo com os aspectos desta revelação, o decodificador de vídeo 30 pode realizar a decodificação de vídeo dos dados de vídeo de acordo com os elementos de sintaxe. Por exemplo, o decodificador de vídeo

30 pode atribuir recursos de decodificação com base nos elementos de sintaxe decodificados e prosseguir com a decodificação dos dados de vídeo.

[00119] A Figura 4A e 4B retrata diagramas conceituais que mostram exemplos de sincronização de CTB no processamento paralelo de frente de onda. Deve-se notar que os exemplos da Figura 4 A e Figura 4B são apresentados para fins ilustrativos. Em outros exemplos, mais que duas fileiras de CTB podem ser processadas em paralelo. Além disso, um quadro de vídeo pode compreender mais ou menos CTBs em uma fileira de CTB que as 7 CTBs que são retratadas na Figura 4A e Figura 4B.

[00120] Figura 4A mostra um exemplo de processamento paralelo de frente de onda com pontos de sincronização atrasado por um CTB. No exemplo da Figura 4A, duas fileiras de CTB são processadas (por exemplo, decodificada) em paralelo. A decodificação de CTB 0 na fileira 1 de CTB, no ponto de sincronização A, não inicia até que a decodificação da fileira 0 de CTB também alcance o ponto de sincronização A.

[00121] No exemplo da Figura 4A, ponto de sincronização A ocorre após a decodificação de CTB 0 na fileira 0 de CTB ter sido completada. Esse atraso permite a decodificação de informações de elementos de sintaxe e/ou de pixel em CTB 0 na fileira 1 que dependem de informações de elementos de sintaxe e/ou pixel decodificadas em um CTB diretamente acima, ou acima e no lado esquerdo do CTB. Isto é, o decodificador de vídeo 30 pode exigir que determinados elementos de sintaxe na fileira 0 de CTB 0 já tenham sido decodificados, a fim de decodificar adequadamente os elementos de sintaxe na fileira 1 de CTB 0. Os pontos de sincronização B-G mostram outros pontos nos quais a codificação da fileira 0 pode ser finalizada, antes da

codificação da fileira 1 pode começar no ponto correspondente. Conforme pode ser observado, a sincronização é exigida em cada CTB. Nesse contexto, a sincronização não exige necessariamente que a decodificação, por exemplo, ponto A, em ambas as fileiras ocorra simultaneamente. Em vez disso, a sincronização exige meramente que a decodificação da fileira superior alcance o ponto de sincronização antes da decodificação na fileira inferior pode começar no ponto de sincronização correspondente.

[00122] Figura 4B é um diagrama conceitual que ilustra outro exemplo de pontos de sincronização para processamento paralelo de frente de onda. Nesse exemplo, a decodificação de informações de pixel e/ou de elementos de sintaxe em uma fileira inferior de CTB (por exemplo, fileira 1 de CTB) também pode depender de informações de pixel e/ou de elementos de sintaxe decodificados em uma CU localizada na parte de cima e do lado direito da CU decodificada atualmente. Desse modo, dois CTBs na fileira 0 (isto é, CTB 0 e CTB 1) devem ser decodificados antes da decodificação poder iniciar na fileira 1 de CTB. Conforme mostrado na Figura 4B, o decodificador de vídeo 30 pode iniciar a decodificação de CTB 0 na fileira 1 de CTB quando a decodificação de CTB 1 na fileira 0 de CTB tiver sido completada.

[00123] Conforme discutido acima, os elementos de sintaxe são criptados tipicamente por criptador por entropia no criptador de vídeo 20. Por exemplo, criptador de vídeo 20 pode criptar elementos de sintaxe com o uso de CABAC. Cada elemento de sintaxe pode ser binarizado em um ou mais bins para a aplicação de CABAC. Desse modo, os elementos de sintaxe para todos os CTBs podem ser sinalizados por um único bin ou por diversos bins. Deseja-se ter equilíbrio de carga para a operação de análise entre os CTBs. Se a

sincronização do módulo de análise é reduzida para ser realizada no início de cada fileira de CTB, em vez de em cada CTB, o equilíbrio de carga para análise dos subfluxos de onda de frente (isto é, a análise de cada fileira de CTB) pode ser alcançada. Portanto, a fim de eliminar a necessidade de sincronização de por CTB, o criptador de vídeo 20 pode ser configurado de modo que todos os elementos de sintaxe sejam decodificáveis sem o uso de quaisquer informações de elemento de sintaxe a partir de uma fileira de CTB acima do CTB codificado atualmente.

[00124] WPP, conforme mostrado nos exemplos da Figura 4 A e 4B, pode ser habilitado com o uso de um ou mais elementos de sintaxe. Por exemplo, conforme estabelecido na especificação FIEVC, o elemento de sintaxe `entropy_coding_sync_enabled_flag` igual a 1 especifica que um processo de sincronização específico para contextos variáveis é invocado antes da decodificação da unidade de árvore de codificação que inclui o primeiro bloco de árvore de codificação de uma fileira de blocos de árvore de codificação em cada tile em cada imagem que se refere ao PPS, e um processo de armazenamento específico para contextos variáveis é invocado após a decodificação da unidade de árvore de codificação que inclui o segundo bloco de árvore de codificação de uma fileira de blocos de árvore de codificação em cada tile em cada imagem que se refere ao PPS. Além disso, o elemento de sintaxe `entropy_coding_sync_enabled_flag` que é igual a um valor de 0 especifica que nenhum processo de sincronização específico para contextos variáveis deve ser invocado antes da decodificação da unidade de árvore de codificação que inclui o primeiro bloco de árvore de codificação de uma fileira de blocos de árvore de codificação em cada tile em cada imagem que se refere ao PPS, e nenhum processo de armazenamento

específico para contextos variáveis deve ser invocado após a decodificação da unidade de árvore de codificação que inclui o segundo bloco de árvore de codificação de uma fileira de blocos de árvore de codificação em cada tile em cada imagem que se refere ao PPS. Em alguns exemplos, o elemento de sintaxe `entropy_coding_sync_enabled_flag` pode ser usado para controlar a sincronização descrita acima com referência às Figuras 4 A e 4B.

[00125] A Figura 5 é um diagrama conceitual que ilustra outro exemplo de pontos de sincronização para processamento paralelo de fileiras de bloco de vídeo. No exemplo da Figura 5, o criptador de vídeo 20 e o decodificador de vídeo 30 são configurados para analisar todos os elementos de sintaxe sem usar quaisquer informações a partir de uma fileira de CTB acima da fileira de CTB codificada atualmente. Desse modo, a parte de análise de processamento paralelo de frente de onda pode prosseguir a partir de um único ponto de sincronização no início da fileira de CTB, visto que quaisquer diferenças na complexidade de análise e no período de tempo de análise entre as fileiras não tem suporte em outra fileira.

[00126] Conforme mostrado na Figura 5, a codificação da fileira 0 de CTB pode começar antes da codificação da fileira 1 de CTB, conforme as técnicas de codificação preditiva (por exemplo, interprevisão ou intraprevisão) usadas durante o processo de reconstrução podem ainda depender das informações de pixel na fileira acima. Conforme mostrado na Figura 5, o processamento da fileira 1 de CTB pode começar assim que a reconstrução de CTB 1 na fileira 0 de CTB for completada. No entanto, diferente do exemplo da Figura 4A e da Figura 4B, nenhum ponto de sincronização adicional é usado para processar fileiras de CTB 0 e 1. Desse modo, uma primeira pluralidade

de fileiras de CTB pode ser processada em paralelo, com o uso de dois atrasos de CTB, com apenas um ponto de sincronização, isto é, ponto de sincronização A'. Uma segunda pluralidade de fileiras de CTB (isto é, fileiras de CTB 2 e 3) pode ser codificada em paralelo pelo criptador de vídeo 20 ou decodificador de vídeo 30 que usa um único ponto de sincronização B' após a primeira pluralidade de fileiras de CTB ser processada. No exemplo da Figura 5, o primeiro e o segundo grupos de fileiras de CTB incluem duas fileiras de CTB. No entanto, mais que duas fileiras de CTB podem ser processadas em paralelo com o uso das técnicas desta revelação.

[00127] De acordo com os aspectos desta revelação, o criptador de vídeo 20 pode ser configurado para criptar dados que indicam uma configuração de processamento paralelo para camadas de um fluxo de bits de múltiplas camadas. Por exemplo, o criptador de vídeo 20 pode ser configurado para criptar os dados de um VPS e/ou VPS VUI que indicam uma configuração de processamento paralelo para as camadas do fluxo de bits de múltiplas camadas. Em particular, com referência à configuração de processamento paralelo, os dados podem indicar a possibilidade de a sincronização de processamento paralelo, tal como aquela descrita acima com referência às Figuras 4A a 5, ser realizada. De modo semelhante, o decodificador de vídeo 30 pode ser configurado para decodificar tais dados, por exemplo, a partir de VPS e/ou VPS VUI. Em alguns exemplos, os dados que indicam a configuração de processamento paralelo podem incluir uma diversidade de elementos de sintaxe. Por exemplo, o criptador de vídeo 20 pode criptar um ou mais elementos de sintaxe que indicam a possibilidade de a sincronização de codificação por entropia ser realizada em quaisquer imagens de uma camada em um fluxo de bits de múltiplas camadas. Isto é, o criptador

de vídeo 20 pode fornecer uma indicação precoce, com o uso do VPS, da possibilidade de a sincronização de codificação por entropia habilitada ser realizada nas camadas do fluxo de bits de múltiplas camadas (por exemplo, devido ao VPS que ocorre relativamente mais cedo no fluxo de bits que os conjuntos de parâmetros que realmente habilitam a sincronização de codificação por entropia, tal como um PPS). Em um exemplo para fins ilustrativos, o criptador de vídeo 20 pode criptar um elemento de sintaxe `entropy_coding_sync_enabled_vps_flag[i]` que indica a possibilidade de o valor do elemento de sintaxe `entropy_coding_sync_enabled_flag` ser igual a 1 para cada PPS que é referido por pelo menos uma imagem de uma camada particular. O decodificador de vídeo 30 também pode ser configurado para decodificar tais informações do VPS. O decodificador de vídeo 30 pode, em alguns exemplos, usar tais informações para atribuir recursos de sistema para decodificar os dados de vídeo de múltiplas camadas.

[00128] A Figura 6 é um diagrama conceitual que ilustra exemplo de pequenos blocos de acordo com o padrão HEVC. O uso da estrutura de tile mostrada na Figura 6 pode, em alguns exemplos, tornar um codec de uso paralelo mais intuitivo. HEVC define pequenos blocos como um número inteiro de CTBs que ocorrem simultaneamente em uma coluna e uma fileira, em ordem consecutivamente em uma varredura de rastreamento de CTB do tile. A divisão de cada imagem de pequenos blocos pode ser chamado de partição. Pequenos blocos em uma imagem são ordenados consecutivamente na varredura de rastreamento de tile da imagem, conforme mostrado na Figura 6.

[00129] Por exemplo, a Figura 6 ilustra uma ordem de codificação de CTB exemplificativa para uma imagem 96 que é particionada em múltiplos pequenos blocos 98A, 98B, 98C,

98D, 98E, 98F, 98G, 98H, e 98I (coletivamente, "pequenos blocos 98"), com limites de tile indicados pelas linhas espessas. Cada bloco quadrado na imagem 96 representa um bloco de pixel associado a um CTB. Os números nos blocos de pixel indicam as posições dos CTBs correspondentes (por exemplo, LCUs) em uma ordem de codificação de tile para a imagem 96. Conforme ilustrado no exemplo da Figura 6, CTBs no tile 98A são codificados primeiro, seguidos pelos CTBs no tile 98B, seguidos pelos CTBs no tile 98C, seguidos pelos CTBs no tile 98D, seguidos pelos CTBs no tile 98E, seguidos pelos CTBs no tile 98F, seguidos pelos CTBs no tile 98G, seguidos pelos CTBs no tile 98H, e seguidos pelos CTBs no tile 98I. No interior de cada um dentre os pequenos blocos 98, os CTBs são codificados de acordo com a ordem de varredura de rastreamento.

[00130] O número de pequenos blocos e as localizações de seus limites podem ser definidos para toda sequência ou alterados de imagem a imagem. Os limites de tile, de modo semelhante aos limites de fatia, quebram as dependências de análise e de previsão de modo que um tile possa ser processado independentemente. No entanto, em alguns exemplos, filtros in-loop (por exemplo, filtros de desbloqueio e de deslocamento adaptável de amostra) podem ainda atravessar os limites de tile. Por exemplo, HEVC fornece um filtro de malha através elemento de sintaxe `pequenos blocos_enabled_flag` especificado em um PPS. Quando o valor do filtro de malha através elemento de sintaxe `pequenos blocos_enabled_flag` é igual para 1, as operações de filtragem in-loop podem ser realizadas através de limites de tile em imagens que se referem ao PPS. A elemento de sintaxe `loop_filter_across_pequenos blocos_enabled_flag` que é igual a um valor de 0 especifica que as operações de filtragem in-loop não são realizadas através de limites de tile in

imagens, que se refere ao PPS.

[00131] O uso de pequenos blocos pode aprimorar o paralelismo, devido ao fato de que nenhuma (ou relativamente pouca) comunicação é necessária entre os processadores ou núcleos de processador para decodificação por entropia e reconstrução de compensação de movimento. Além disso, pequenos blocos podem exibir uma eficácia de codificação relativamente melhor quando comparados às fatias, devido ao fato dos pequenos blocos permitirem formatos de partição de imagem que contêm amostras com correlação potencialmente mais alta que as fatias. Pequenos blocos também podem reduzir a suspensão de cabeçalho de fatia.

[00132] Quando um tile é usado em uma codificação de cada única, o elemento de sintaxe `min_spatial_segmentation_idc` pode ser usado por um decodificador de vídeo (tal como decodificador de vídeo 30) para calcular o número máximo de amostras de luma a ser processado por um encadeamento de processamento, presumindo que o decodificador de vídeo 30 utilize maximamente as informações de decodificação paralela. Quando não for igual a 0, um elemento de sintaxe `mm_spatial_segmentation_idc` pode ajudar a estabelecer uma ligação no tamanho máximo possível de regiões de segmentação espacial codificadas distintas nas imagens da sequência de vídeo codificada. No HEVC pode haver algumas interdependências de imagem entre os encadeamentos diferentes, por exemplo, devido à sincronização de codificação por entropia ou filtragem de desbloqueio através de tile ou limites de fatia. Tais interdependências podem afetar a maneira na qual o decodificador de vídeo 30 atribui recursos para decodificar dados de vídeo, conforme descrito no presente documento.

[00133] De acordo com os aspectos desta

revelação, o criptador de vídeo 20 pode criptar dados que indicam uma configuração de pequenos blocos para camadas de um fluxo de bits de múltiplas camadas. Por exemplo, criptador de vídeo 20 pode criptar dados de um VPS e/ou VPS VUI que indicam a possibilidade de os pequenos blocos, tais como pequenos blocos mostrados e descritos acima com referência à Figura 6, serem habilitados para uma camada particular de um fluxo de bits de múltiplas camadas. Adicional ou alternativamente, o criptador de vídeo 20 pode criptar dados que indicam a possibilidade de a filtragem de loop através de pequenos blocos estar habilitada ou desabilitada. O decodificador de vídeo 30 pode ser configurada para decodificar tais dados, por exemplo, a partir de um VPS e/ou VPS VUI.

[00134] Em alguns exemplos, os dados que indicam a configuração de pequenos blocos podem incluir um número de elementos de sintaxe. Por exemplo, para cada camada de dados de vídeo no fluxo de bits de múltiplas camadas, o criptador de vídeo 20 pode criptar um ou mais elementos de sintaxe que indicam a possibilidade de os pequenos blocos estarem habilitados em pelo menos uma imagem na camada. Além disso, para cada camada para a qual pequenos blocos são permitidos, o criptador de vídeo 20 também pode criptar um ou mais elementos de sintaxe que indicam a possibilidade de a filtragem de loop ser aplicada através de limites de tile, ou a possibilidade de a filtragem de loop através de limites de tile ser desabilitada. Dessa maneira, o criptador de vídeo 20 pode fornecer uma indicação precoce, com o uso do VPS, de uma configuração de pequenos blocos (por exemplo, a possibilidade de os pequenos blocos serem habilitados e/ou a possibilidade de a filtragem ser aplicada através de pequenos blocos) para as camadas do fluxo de bits de múltiplas camadas (por exemplo, devido ao VPS que ocorre

relativamente mais cedo no fluxo de bits que os conjuntos de parâmetros que realmente habilitam a sincronização de codificação por entropia, tal como um PPS).

[00135] Em um exemplo para fins ilustrativos, o criptador de vídeo 20 pode criptar um elemento de sintaxe pequenos blocos_enabled_vps_flag[i] que indica a possibilidade de o valor do elemento de sintaxe pequenos blocos_enabled_flag ser igual a 1 para cada PPS que é referido por pelo menos uma imagem de uma camada particular. Adicional ou alternativamente, o criptador de vídeo 20 pode criptar um elemento de sintaxe de loop_filter_across_pequenos blocos_disabled_vps_flag[i] que indica a possibilidade de o valor do filtro de malha através de pequenos blocos_enabled_flag ser igual a 0 para cada PPS que é referido por pelo menos uma imagem de uma camada particular. O decodificador de vídeo 30 também pode ser configurado para decodificar tais informações do VPS. O decodificador de vídeo 30 pode, em alguns exemplos, usar tais informações para atribuir recursos de sistema para decodificar os dados de vídeo de múltiplas camadas.

[00136] As Figuras 7 a 10 ilustram uma variedade de técnicas de codificação de múltiplas camadas que podem ser usadas em conjunto com as técnicas desta revelação. Por exemplo, as técnicas podem ser aplicadas a componentes de vista de textura, componentes de vista de profundidade e diversas camadas temporais, espaciais e de qualidade, conforme descrito no presente documento,

[00137] A Figura 7 é um diagrama conceitual que ilustra uma padronização de previsão de MVC exemplificativa. Embora a Figura 7 seja descrita com referência a H.264/AVC e MVC, deve ser entendido que uma padronização de previsão semelhante pode ser usada com outros esquemas de codificação de vídeo de múltiplas vistas, incluindo MV-HEVC, 3D-HEVC (a

múltiplas vistas mais extensão de profundidade para HEVC), e múltiplas vistas que usam codificação de vídeo escalonável (consultar, por exemplo, uma codificação de vídeo escalonável exemplificativo descrito com referência às Figuras 8 a 10. Desse modo, as referências à codificação de múltiplas vistas e/ou MVC abaixo se aplicam, de modo geral, a codificação de vídeo de múltiplas vistas, e não são restritas a H.264/MVC.

[00138] No exemplo da Figura 7, as oito vistas (que têm IDs de vista "S0" a "S7") são ilustradas, e as doze localizações temporais ("T0" a "T11") são ilustradas para cada vista. Ou seja, cada fileira na Figura 7 corresponde a uma vista, ao passo que cada coluna indica uma localização temporal.

[00139] Embora a MVC tenha uma então chamada vista de base, que é passível de decodificação por decodificações H.264/AVC, e pares de vista estéreos podem ser suportados também pela MVC, a vantagem da MVC é que pode suportar um exemplo que usa mais que duas vistas como uma entrada de vídeo 3D e decodifica esse vídeo 3D representado pelas múltiplas vistas. Um renderizador de um cliente que tem um decodificador de MVC pode esperar o conteúdo de vídeo 3D com múltiplas vistas.

[00140] As imagens na Figura 7 são indicadas na interseção de cada fileira e cada coluna na Figura 7 que usa um bloco sombreado que inclui uma letra, que designa a possibilidade de a imagem correspondente ser intracodificada (isto é, um quadro I), ou intercodificada em uma direção (isto é, como um quadro P) ou em múltiplas direções (isto é, como um quadro B). Em geral, as previsões são indicadas por setas, em que a imagem apontada usa o objeto apontado para referência de previsão. Por exemplo, o quadro P da vista S2 em uma localização temporal T0 é prevista a partir do quadro

I da vista S0 em uma localização temporal T0.

[00141] Conforme a criptação de vídeo de única vista, as imagens de uma sequência de vídeos de codificação de vídeo de múltiplas vistas pode ser criptada de maneira prevista em relação às imagens em diferentes localizações temporais. Por exemplo, o quadro b da vista S0 na localização temporal T1 tem uma seta apontada para a mesma a partir do quadro I da vista S0 na localização temporal T0, indicando que o quadro b é previsto a partir do quadro I. No entanto, adicionalmente, no contexto de criptação de vídeo de múltiplas vistas, as imagens podem ser previstas entre vistas. Ou seja, um componente de vista pode usar os componentes de vista em outras vistas para referência. Em MVC, como exemplo de codificação de vídeo de múltiplas camadas, a previsão entre vistas é notada como se o componente de vista em outra vista fosse uma referência de interprevisão. As referências de entre vistas potenciais são sinalizadas na extensão de MVC de Conjunto de Parâmetros de Sequência (SPS) e podem ser modificadas pelo processo de construção de lista de imagens de referência, o que habilita o ordenamento flexível das referências de interprevisão ou de previsão entre vistas.

[00142] Em MVC, a previsão entre vistas é permitida entre as imagens na mesma unidade de acesso (isto é, com o mesmo instante do tempo). Uma unidade de acesso é, de modo geral, uma unidade de dados que incluem todas os componentes de vista (por exemplo, todas unidades de camada de abstração de rede (NAL)) para um exemplo temporal comum. Desse modo, em MVC, a previsão de entre vistas é permitida entre as imagens na mesma unidade de acesso. Durante uma imagem em uma dentre as vistas de não base, a imagem pode ser adicionada a uma lista de imagens de referência caso esteja em uma vista diferente, porém, dentro do mesmo

instante do tempo (por exemplo, o mesmo valor de POC, e, desse modo, na mesma unidade de acesso). Uma imagem de referência entre vistas de previsão pode ser colocada em qualquer posição de uma lista de imagens de referência, exatamente como qualquer imagem de referência de interprevisão.

[00143] A Figura 7 fornece vários exemplos de previsão entre vistas. As imagens da vista S1, no exemplo da Figura 7, são ilustradas como previstas a partir das imagens em diferentes localizações temporais da vista S1, assim como, previstas entre vistas a partir das imagens das vistas S0 e S2 nas mesmas localizações temporais. Por exemplo, a quadro b da vista S1 na localização temporal T1 é prevista a partir de cada uma dentre os quadros B da vista S1 nas localizações temporais T0 e T2, assim como, os quadros b das vistas S0 e S2 na localização temporal T1.

[00144] No exemplo da Figura 7, "B" maiúsculo e "b" minúsculo destinam-se a indicar relações hierárquicas diferentes entre imagens, em vez de metodologias de criptação diferentes. De modo geral, as imagens de "B" maiúsculo são relativamente mais altas na hierarquia de previsão que imagens de "b" minúsculo. A Figura 7 também ilustra variações na hierarquia de previsão com o uso de níveis diferentes de sombreamento, em que uma quantidade de sombreamento maior (isto é, relativamente mais escuro) de imagens é maior na hierarquia de previsão que aquelas imagens que têm menos sombreamento (isto é, relativamente mais claro). Por exemplo, os quadros I na Figura 7 são ilustrados com sombreamento completo, enquanto o quadro P tem um sombreamento ligeiramente mais claro, e o quadro B (e quadro com letra b minúscula) têm diversos níveis de sombreamento um em relação ao outro, porém, sempre mais claro que o sombreamento do quadro P e do quadro I.

[00145] De modo geral, a hierarquia de previsão é relacionada aos índices de ordem de vista, em que as imagens relativamente maiores na hierarquia de previsão devem ser decodificadas antes das imagens de decodificação que são relativamente menores na hierarquia, de modo que tais imagens relativamente maiores na hierarquia possam ser usadas como imagens de referência durante decodificação das imagens relativamente menores na hierarquia. Um índice de ordem de vista é um índice que indica a ordem de decodificação de componentes de vista em uma unidade de acesso. Dessa maneira, as imagens usadas como imagens de referência podem ser decodificadas antes das imagens que são criptadas com referência às imagens de referência.

[00146] De acordo com os aspectos desta revelação, o criptador de vídeo 20 pode criptar dados que indicam uma configuração de pequenos blocos e/ou uma configuração de processamento paralelo para camadas de um fluxo de bits de múltiplas camadas, tais como as vistas S0 a S7 mostradas no exemplo da Figura 7. Por exemplo, criptador de vídeo 20 pode criptar dados de um VPS e/ou VPS VUI que indicam uma configuração de pequenos blocos para vistas dos dados de vídeo de múltiplas vistas mostrados no exemplo da Figura 7. Por exemplo, criptador de vídeo 20 pode criptar dados de um VPS e/ou VPS VUI que indicam a possibilidade de os pequenos blocos, tais como pequenos blocos mostrados e descritos acima com referência à Figura 6, são habilitados para uma vista particular dos dados de vídeo de múltiplas vistas. Adicional ou alternativamente, o criptador de vídeo 20 pode criptar dados que indicam a possibilidade de a filtragem de loop através de pequenos blocos estar habilitada ou desabilitada. O decodificador de vídeo 30 pode ser configurada para decodificar tais dados, por exemplo, a partir de um VPS e/ou VPS VUI.

[00147] Adicional ou alternativamente, o criptador de vídeo 20 pode criptar dados de um VPS e/ou VPS VUI que indicam uma configuração de processamento paralelo para vistas dos dados de vídeo de múltiplas vistas mostradas no exemplo da Figura 7. Por exemplo, o criptador de vídeo 20 pode criptar os dados de um VPS e/ou VPS VUI que indica a possibilidade de a sincronização descrita acima em relação às Figuras 4A a 5, ser habilitada para a imagens de uma vista particular de dados de vídeo de múltiplas vistas. O decodificador de vídeo 30 pode ser configurada para decodificar tais dados, por exemplo, a partir de um VPS e/ou VPS VUI.

[00148] A Figura 8 é um diagrama conceitual que ilustra diversas dimensões escalonáveis para codificação de vídeo escalonável (SVC). Nesse exemplo, as escalabilidades são habilitadas e três dimensões. No período de tempo dimensão, as taxas de quadro com 7,5 Hz, 15 Hz ou 30 Hz podem ser suportadas pela escalabilidade temporal (T). Quando escalabilidade espacial (S) é suportada, resoluções diferentes, tais como QCIF, GIF, e 4CIF, podem ser habilitadas. Para cada resolução espacial específica e taxa de quadro, camadas de sinal para ruído (SNR) (Q) que podem ser adicionadas para aprimorar a qualidade de imagem.

[00149] Uma vez que o conteúdo de vídeo tenha sido criptado em tal forma escalonável, uma ferramenta de extrator pode ser usada para adaptar o conteúdo entregue atual de acordo com as exigências de aplicação, que podem ser dependentes, por exemplo, dos clientes ou do canal de transmissão. No exemplo mostrado na Figura 8, cada volume cúbico, isto é, cubo, contém as imagens com a mesma taxa de quadro (nível temporal), camadas de resolução espacial e SNR. A melhor representação pode ser alcançada adicionando-se aqueles cubos (imagens) em qualquer dimensão. A

escalabilidade combinada pode ser suportada quando há dois, três ou mesmo mais escalabilidades habilitadas.

[00150] De acordo com a especificação de SVC, as imagens com a menor camada espacial e de qualidade são compatíveis com H.264/AVC, e as imagens no nível temporal mais baixo formam a camada-base temporal, que podem ser aprimoradas com as imagens em níveis temporais altos. Além da camada compatível de H.264/AVC, diversas camadas de aprimoramento espaciais e/ou de SMI podem ser adicionadas para fornecer escalabilidades espaciais e/ou de qualidade. A escalabilidade de SNR também é chamada de escalabilidade de qualidade. Cada camada de aprimoramento espacial ou SNR pode ser escalonável temporalmente, com a mesma estrutura de escalabilidade temporal que a camada compatível de H.264/AVC. Para uma camada aprimorada espacial ou SNR, a camada inferior dependente também é chamada de camada-base de tal camada aprimorada espacial ou SNR específica.

[00151] A Figura 9 é um diagrama conceitual que ilustra um exemplo da estrutura de codificação de SVC. Nesse exemplo, as imagens com a camada espacial e de qualidade mais baixa (as imagens na camada 0 e na camada 1, com resolução de QCIF) são compatíveis com H.264/AVC. Entre essas, aquelas imagens do nível temporal mais baixo forma a camada-base temporal (conforme mostrado na camada 0) da Figura 9. Essa camada-base temporal (camada 0) pode ser aprimorada com imagens de níveis temporais mais altos (camada 1). Além da camada compatível de H.264/AVC, diversas camadas de aprimoramento espaciais e/ou de SNR podem ser adicionadas para fornecer escalabilidades espaciais e/ou de qualidade. Por exemplo, a camada aprimorada pode ser uma representação em GIF com a mesma resolução que a camada 2. Nesse exemplo, a camada 3 é uma camada de SNR aprimorada. Conforme mostrado nesse exemplo, cada camada de aprimoramento espacial ou SNR

pode ser escalonável temporalmente, com a mesma estrutura de escalabilidade temporal que a camada compatível de H.264/AVC. Ademais, uma camada aprimorada pode aprimorar tanto a resolução espacial quanto a taxa de quadro. Por exemplo, a camada 4 fornece uma camada aprimorada de 4CIF, que aumenta adicionalmente a taxa de quadro a partir de 15 Hz a 30 Hz.

[00152] A Figura 10 é um diagrama conceitual que ilustra as unidades de acesso exemplificativas (Alls). Cada AU inclui uma ou mais fatias, encapsuladas no interior de unidades de camada de abstração de rede (NAL). Pode ter zero ou mais unidades de NAL por unidade de acesso por camada. Um conjunto de unidades de NAL correspondente a uma camada no interior de uma AU pode ser chamado de "componente de camada". O exemplo da Figura 10 retrata os componentes de camada correspondentes aos componentes de camada de Figura 9. Conforme mostrado no exemplo da Figura 10, as fatias codificadas no mesmo período de tempo exemplificativo (isto é, em uma AU comum) são sucessivas na ordem de fluxo de bits e formam uma unidade de acesso no contexto de SVC. Aquelas unidades de acesso de SVC, então, seguem a ordem de decodificação, que pode ser diferente da ordem de exibição e decidida, por exemplo, pela relação temporal de previsão,

[00153] As extensões escalonáveis de H.264/AVC (Codificação de Vídeo Avançada) são descritas abaixo. Algumas funcionalidades de SVC são herdadas do H.264/AVC. Em comparação aos padrões escalonáveis anteriores, parte das maiores vantagens da extensão de SVC para H.264/AVC, a saber, previsão de intercâmara e decodificação de loop único, são revistas abaixo.

[00154] A extensão de SVC de H.264/AVC suportam decodificação de loop único. Com a decodificação de loop único, cada camada suportada pode ser decodificada com um

loop único de compensação de movimento. Para alcançar isso, o uso de intraprevisão de intercamada é permitida apenas para macro blocos de camada aprimorada, para os quais o sinal de camada de referência colocado é intracodificado. Exige-se adicionalmente que todas as camadas que são usadas para camadas mais altas de previsão de intercamada sejam codificadas com o uso de intraprevisão restringida.

[00155] A extensão de SVC de H.264/AVC também suportar a previsão de intercamada. SVC introduz a previsão de intercamada para escalabilidades espaciais e de SNR com base em textura, resíduo e movimento. A escalabilidade espacial em SVC foi generalizada para qualquer razão de resolução entre duas camadas. A escalabilidade de SNR pode ser realizada pela Escalabilidade de Granularidade Bruta (COS) ou Escalabilidade de Granularidade Média (MGS). Em SVC, duas camadas espaciais ou CGS pertencem a camadas de dependência diferentes (indicadas pelo elemento de sintaxe `dependency_id` no cabeçalho de unidade NAL), enquanto duas camadas de MGS possam estar na mesma camada de dependência. Uma camada de dependência inclui camadas de qualidade com o valor do elemento de sintaxe `quality_id` a partir de 0 aos valores maiores, que corresponde às camadas de qualidade aprimoradas. Em SVC, os métodos de previsão de intercamada são utilizados para reduzir a redundância de intercamada.

[00156] Conforme observado acima, semelhante ao H.264/AVC, HEVC também terá a extensão de codificação de vídeo escalonável, atualmente chamada de SHVC, que fornecerá pelo menos a escalabilidade temporal. Uma escalabilidade de SNR e escalabilidade espacial. Em SHVC, para alcançar a previsão de textura de intercamada, as amostras reconstruídas de camada de referência primeiro aumentam a resolução quando a resolução da camada de referência é inferior àquela da camada aprimorada. Mesmo no caso de

escalabilidade de SNR, as amostras de camada de referência podem ser filtradas antes de serem usadas para a previsão de intercamada para obter eficácia de codificação mais alta. O processo de aumento de resolução e de filtragem de intercamada podem ser realizados para todas as imagens de camada, que também podem ser chamadas de componentes de camada ou simplesmente de imagens. Em SHVC, uma estrutura de decodificação de múltiplos loop pode ser usada e um decodificador de vídeo (tal como decodificador de vídeo 30) pode processar camadas diferentes em paralelo.

[00157] De acordo com os aspectos desta revelação, o criptador de vídeo 20 pode criptar dados que indicam uma configuração de pequenos blocos e/ou uma configuração de processamento paralelo para camadas de um fluxo de bits de múltiplas camadas, tais como as camadas escalonáveis mostradas nos exemplos das Figuras 8 a 10. Por exemplo, o criptador de vídeo 20 pode criptar dados de um VPS e/ou VPS VUI que indicam uma configuração de pequenos blocos para camadas de dados de vídeo escalonáveis mostrados nos exemplos das Figuras 8 a 10. Por exemplo, criptador de vídeo 20 pode criptar dados de um VPS e/ou VPS VUI que indicam a possibilidade de os pequenos blocos, tais como pequenos blocos mostrados e descritos acima com referência à Figura 6, estarem habilitados para uma camada particular de dados de vídeo escalonáveis. Adicional ou alternativamente, o criptador de vídeo 20 pode criptar dados que indicam a possibilidade de a filtragem de loop através de pequenos blocos estar habilitada ou desabilitada. O decodificador de vídeo 30 pode ser configurada para decodificar tais dados, por exemplo, a partir de um VPS e/ou VPS VUI.

[00158] Adicional ou alternativamente, o criptador de vídeo 20 pode criptar dados de um VPS e/ou VPS

VUI que indicam que uma configuração de processamento paralelo para camadas de dados de vídeo escalonáveis, tais como as camadas mostradas nas Figuras 8 a 10. Por exemplo, o criptador de vídeo 20 pode criptar os dados de um VPS e/ou VPS VUI que indica a possibilidade de a sincronização descrita acima em relação às Figuras 4A a 5, estar habilitada para imagens de uma camada particular de dados de vídeo escalonáveis. O decodificador de vídeo 30 pode ser configurada para decodificar tais dados, por exemplo, a partir de um VPS e/ou VPS VUI.

[00159] A Figura 11 ilustra uma técnica para processar dados de vídeo de múltiplas camadas consistente com esta revelação. Embora descritas em relação ao criptador de vídeo 20 (Figuras 1 e 2) para fins exemplificativos, as técnicas desta revelação podem ser realizadas com o uso de uma variedade de outros processadores. Além disso, o processo particular mostrado nas Figura 11 é fornecido para fins exemplificativos. Em outros exemplos, um processo consistente com as técnicas desta revelação pode incluir mais, menos ou uma disposição alternativa de etapas.

[00160] O criptador de vídeo 20 pode determinar uma configuração de pequenos blocos para camadas de dados de vídeo de múltiplas camadas (110). Por exemplo, o criptador de vídeo 20 pode determinar a possibilidade de usar pequenos blocos ao criptar as camadas de dados de vídeo. O criptador de vídeo 20 também pode determinar a possibilidade de aplicar filtros de malha através de limites de tile ao criptar as camadas de dados de vídeo. Em alguns exemplos, o criptador de vídeo 20 pode determinar a possibilidade de usar pequenos blocos e a possibilidade de aplicar filtros de malha através de limites de tile com base em uma análise de taxa de distorção.

[00161] O criptador de vídeo 20 também pode

determinar uma configuração de processamento paralelo para codificar as camadas de dados de vídeo (112). Por exemplo, o criptador de vídeo 20 pode determinar aplicar WPP, bem como da maneira na qual o WPP é realizado, conforme descrito nos exemplos acima com referência às Figuras 4A a 5. O criptador de vídeo 20 pode, então, criptar as camadas dos dados de vídeo de múltiplas camadas que usam a configuração de pequenos blocos e a configuração de processamento paralelo determinadas (114). Por exemplo, o criptador de vídeo 20 pode realizar intra ou interprevisão de dados de vídeo (que incluem previsão de intercamada), conforme descrito acima.

[00162] De acordo com os aspectos desta revelação, o criptador de vídeo 20 também pode criptar dados que indicam a configuração de pequenos blocos e a configuração de processamento paralelo (116). Por exemplo, o criptador de vídeo 20 pode criptar os dados de um VPS e/ou VPS VUI que indicam uma configuração de pequenos blocos para as camadas do fluxo de bits de múltiplas camadas e uma configuração de processamento paralelo para as camadas do fluxo de bits de múltiplas camadas. Em um exemplo para fins ilustrativos, o criptador de vídeo 20 pode criptar um ou mais elementos de sintaxe que indicam a possibilidade de os pequenos blocos estarem habilitados em pelo menos uma imagem na camada (por exemplo, tal como elemento de sintaxe pequenos_blocos_enabled_vps_flag[i] mostrado no exemplo da Tabela 1 acima). Além disso, para cada camada para a qual pequenos blocos são habilitados, o criptador de vídeo 2,0 também pode criptar um ou mais elementos de sintaxe que indicam a possibilidade de a filtragem de loop ser aplicada através de limites de tile, ou a possibilidade de a filtragem de loop através de limites de tile ser desabilitada (por exemplo, tal como o elemento de sintaxe de loop_filter_across_pequenos_blocos_disabled_vps_flag[i]

mostrado no exemplo da Tabela 1 acima). O criptador de vídeo 20 também pode criptar um ou mais elementos de sintaxe que indica uma configuração de processamento paralelo. Por exemplo, criptador de vídeo 20 pode criptar um ou mais elementos de sintaxe que indicam a possibilidade de a sincronização de codificação por entropia ser realizada, por exemplo, através WPP (por exemplo, tal como o elemento de sintaxe `entropy_coding_sync_enabled_vps_flag[i]` mostrado no exemplo da Tabela 1 acima).

[00163] A Figura 12 ilustra outra técnica para processar dados de vídeo de múltiplas camadas consistente com esta revelação. Embora descritas em relação ao decodificador de vídeo 30 (Figuras 1 e 3) para fins exemplificativos, as técnicas desta revelação podem ser realizadas com o uso de uma variedade de outros processadores. Além disso, o processo particular mostrado na Figura 12 é fornecido para fins exemplificativos. Em outros exemplos, um processo consistente com as técnicas desta revelação pode incluir mais, menos ou uma disposição alternativa de etapas.

[00164] O decodificador de vídeo 30 pode receber um fluxo de bits de múltiplas camadas (120). Por exemplo, o decodificador de vídeo 30 pode receber um fluxo de bits que está em conformidade com um padrão de codificação de vídeo particular, tal como o HEVC ou suas extensões (por exemplo, tal como MV-HEVC, 3D-HEVC, SHVC ou similares).

[00165] De acordo com os aspectos desta revelação, o decodificador de vídeo 30 pode decodificar os dados que indicam uma configuração de pequenos blocos (122). O decodificador de vídeo 30 também pode decodificar os dados que indicam uma configuração de processamento paralelo (124). Por exemplo, o decodificador de vídeo 30 pode decodificar os dados de um VPS e/ou VPS VUI que indicam uma

configuração de pequenos blocos para as camadas do fluxo de bits de múltiplas camadas e uma configuração de processamento paralelo para as camadas do fluxo de bits de múltiplas camadas. Em um exemplo para fins ilustrativos, o decodificador de vídeo 30 pode decodificar um ou mais elementos de sintaxe que indicam a possibilidade de os pequenos blocos estarem habilitados em pelo menos uma imagem na camada (por exemplo, tal como elemento de sintaxe pequenos_blocos_enabled_vps_flag[i] mostrado no exemplo da Tabela 1 acima). Além disso, para cada camada para a qual pequenos blocos são habilitados, o decodificador de vídeo 30 também pode decodificar um ou mais elementos de sintaxe que indicam a possibilidade de a filtragem de loop ser aplicada através de limites de tile, ou a possibilidade de a filtragem de loop através de limites de tile ser desabilitada (por exemplo, tal como o elemento de sintaxe loop_filter_across_pequenos_blocos_disabled_vps_flag[i] mostrado no exemplo da Tabela 1 acima). Em alguns exemplos, o decodificador de vídeo 30 também pode decodificar um ou mais elementos de sintaxe que indicam uma configuração de processamento paralelo. Por exemplo, decodificador de vídeo 30 pode decodificar um ou mais elementos de sintaxe que indicam a possibilidade de a sincronização de codificação por entropia ser realizada, por exemplo, através WPP (por exemplo, tal como o elemento de sintaxe entropy_coding_sync_enabled_vps_flag[i] mostrado no exemplo da Tabela 1 acima).

[00166] Em alguns exemplos, o decodificador de vídeo 30 pode, então, atribuir recursos de decodificação para decodificar os dados de múltiplas camadas (126). Por exemplo, de acordo com aspectos desta revelação, as informações descritas acima podem fornecer uma indicação precoce da configuração de pequenos blocos e/ou da

configuração de processamento paralelo de dados de vídeo, por exemplo, antes das imagens de decodificação dos dados de vídeo. O decodificador de vídeo 30 pode atribuir pequenos blocos particulares de dados de vídeo (se habilitados) a serem decodificados por núcleos de processamento particulares. O decodificador de vídeo 30 pode, ao atribuir recursos, considerar a possibilidade de a filtragem de loop poder ser aplicada através de limites de tile, além disso, o decodificador de vídeo 30 pode determinar recursos de decodificação com base na configuração de processamento paralelo (por exemplo, configuração de WPP) de imagens dentro de uma camada particular. O decodificador de vídeo 30 pode, então, decodificar os dados de vídeo de múltiplas camadas, por exemplo, usar um processo recíproco conforme aplicado pelo criptador de vídeo 20 (128).

[00167] Determinados aspectos desta revelação foram descritos com referência a padrão e extensões HEVC do padrão HEVC para fins ilustrativos. No entanto, as técnicas descritas nesta revelação podem ser úteis para outros processos de codificação de vídeo, que incluem outro padrão ou processos de codificação de vídeo de proprietário ainda não desenvolvidos.

[00168] Um codificador de vídeo, conforme descrito nesta revelação, pode se referir a um criptador de vídeo ou um decodificador de vídeo. De modo similar, uma unidade de codificação de vídeo pode se referir a um criptador de vídeo ou um decodificador de vídeo. De modo semelhante, a codificação de vídeo pode se referir a criptação de vídeo ou decodificação de vídeo, conforme aplicável.

[00169] Deve ser reconhecido que, dependendo do exemplo, certos atos ou eventos de quaisquer uma dentre as técnicas descritas no presente documento podem ser

realizados em uma sequência diferente, podem ser adicionados, fundidos ou deixados de fora todos juntos (por exemplo, nem todos os atos e eventos descritos são necessários para a prática das técnicas). Ademais, em certos exemplos, os atos e eventos podem ser realizados de modo concorrente, por exemplo, através de processamento de múltiplos encadeamentos, processamento interrompido ou em múltiplos processadores, em vez de sequencialmente.

[00170] Em um exemplo, o decodificador de vídeo 30 pode ser configurado para analisar um ou mais elementos de sintaxe correspondente aos dados de vídeo, em que os um ou mais elementos de sintaxe indicam uma ou mais dentre uma configuração de pequenos blocos, uma configuração de filtro de malha e uma configuração de processamento paralelo, e realizar a decodificação paralela dos dados de vídeo de acordo com os um ou mais elementos de sintaxe. O decodificador de vídeo 30 pode ser configurado para receber os um ou mais elementos de sintaxe em um conjunto de parâmetros de vídeo.

[00171] Em um exemplo da revelação, os um ou mais elementos de sintaxe incluem uma pequenos blocos_enabled_vps_flag que indica a configuração de pequenos blocos para cada conjunto de parâmetros de imagem referida por pelo menos uma imagem dos dados de vídeo, uma loop_filter_across_pequenos_blocos_disabled_vps_flag que indica a configuração de filtro de malha para cada conjunto de parâmetros de imagem referida por pelo menos uma imagem dos dados de vídeo, e uma entropy_coding_sync_enabled_vps_flag que indica a configuração de processamento paralelo para cada conjunto de parâmetros de imagem referida por pelo menos uma imagem dos dados de vídeo.

[00172] Em outro exemplo da revelação, os um ou

mais elementos de sintaxe incluem um elemento de sintaxe `parallel_tools_vps_idc`, em que o elemento de sintaxe `parallel_tools_vps_idc` indica tanto a configuração de pequenos blocos quanto a configuração de processamento paralelo para cada camada dos dados de vídeo referindo-se ao conjunto de parâmetros de vídeo.

[00173] Em outro exemplo da revelação, os um ou mais elementos de sintaxe incluem um elemento de sintaxe `parallel_tools_vps_idc`, em que o elemento de sintaxe `parallel_tools_vps_idc` indica tanto a configuração de pequenos blocos quanto a configuração de processamento paralelo para cada conjunto de parâmetros de imagem referida por pelo menos uma imagem dos dados de vídeo.

[00174] Em outro exemplo da revelação, os um ou mais elementos de sintaxe incluem uma `tile_entropy_sync_decoding_not_mixed_flag` e uma `loopfilter_across_tile_boundary_not_mixed_flag`, em que a `tile_entropy_sync_coding_not_mixed_flag` indica a configuração de pequenos blocos e a configuração de processamento paralelo para cada conjunto de parâmetros de imagem referida por pelo menos uma imagem dos dados de vídeo, e em que o `loop_filter_across_tile_boundary_not_mixed_flag` indica a configuração de filtro de malha para cada conjunto de parâmetros de imagem referida por pelo menos uma imagem dos dados de vídeo.

[00175] Em outro exemplo da revelação, criptador de vídeo 20 pode ser configurado para determinar um ou mais elementos de sintaxe correspondentes aos dados de vídeo, em que os um ou mais elementos de sintaxe indicam uma ou mais dentre uma configuração de pequenos blocos, uma configuração de filtro de malha e uma configuração de processamento paralelo, realizar criptação paralela dos dados de vídeo de acordo com os um ou mais elementos de sintaxe, e sinalizar

os um ou mais elementos de sintaxe. O criptador de vídeo 20 pode sinalizar os um ou mais elementos de sintaxe em um conjunto de parâmetros de vídeo.

[00176] Em um exemplo da revelação, os um ou mais elementos de sintaxe incluem uma pequenos blocos_enabled_vps_flag que indica a configuração de pequenos blocos para cada conjunto de parâmetros de imagem referida por pelo menos uma imagem dos dados de vídeo, uma loop_filter_across_pequenos_blocos_disabled_vps_flag que indica a configuração de filtro de malha para cada conjunto de parâmetros de imagem referida por pelo menos uma imagem dos dados de vídeo, e uma entropy_coding_sync_enabled_vps_flag que indica a configuração de processamento paralelo para cada conjunto de parâmetros de imagem referida por pelo menos uma imagem dos dados de vídeo.

[00177] Em outro exemplo da revelação, os um ou mais elementos de sintaxe incluem um elemento de sintaxe parallel_tools_vps_idc, em que o elemento de sintaxe parallel_tools_vps_idc indica tanto a configuração de pequenos blocos quanto a configuração de processamento paralelo para cada camada dos dados de vídeo referindo-se ao conjunto de parâmetros de vídeo.

[00178] Em outro exemplo da revelação, os um ou mais elementos de sintaxe incluem um elemento de sintaxe parallel_tools_vps_idc, em que o elemento de sintaxe parallel_tools_vps_idc indica tanto a configuração de pequenos blocos quanto a configuração de processamento paralelo para cada conjunto de parâmetros de imagem referida por pelo menos uma imagem dos dados de vídeo.

[00179] Em outro exemplo da revelação, os um ou mais elementos de sintaxe incluem uma tile_entropy_sync_coding_not_mixed_flag e uma

loopfilter_across_tile_boundary_not_mixed_flag, em que a tile_entropy_sync_coding_not_mixed_flag indica a configuração de pequenos blocos e a configuração de processamento paralelo para cada conjunto de parâmetros de imagem referida por pelo menos uma imagem dos dados de vídeo, e em que o loopfilter_across_tile_boundary_not_mixed_flag indica a configuração de filtro de malha para cada conjunto de parâmetros de imagem referida por pelo menos uma imagem dos dados de vídeo.

[00180] Em um ou mais exemplos, as funções descritas podem ser implantadas em hardware, software, firmware ou qualquer combinação dos mesmos. Se implantado em software, as funções podem ser armazenadas em, ou transmitidas sobre, como uma ou mais instruções ou código em um meio legível por computador e executadas por uma unidade de processamento com base em hardware.

[00181] Meios legíveis por computador podem incluir meios de armazenamento legíveis por computador, que correspondem a um meio tangível, tal como meios de armazenamento de dados, ou meios de comunicação que inclui qualquer meio que facilita a transferência de um programa de computador a partir de um lugar para outro, por exemplo, de acordo com um protocolo de comunicação. Dessa maneira, meios legíveis por computador, de modo geral, podem corresponder a (1) meios de armazenamento legíveis tangíveis por computador que é não transitório ou (2) um meio de comunicação tal como um sinal ou onda de portadora. As mídias de armazenamento de dados podem ser quaisquer mídias disponíveis que possam ser acessadas por um ou mais computadores ou um ou mais processadores para recuperar instruções, estruturas de código e/ou dados para a implantação das técnicas descritas nesta revelação. Um produto de programa de computador pode incluir uma mídia

legível por computador.

[00182] A título de exemplo, e não de limitação, tais mídias de armazenamento legíveis por computador podem compreender RAM, ROM, EEPROM, CD-ROM ou outro armazenamento de disco óptico, armazenamento de disco magnético ou outros dispositivos de armazenamento magnético, ou qualquer outro meio que possa ser usado para armazenar o código de programa desejado na forma de instruções ou estruturas de dados e que possa ser acessado por um computador. Também, qualquer conexão pode ser propriamente denominada um meio legível por computador. Por exemplo, se as instruções forem transmitidas a partir de um sítio da web, servidor ou outra fonte remota com o uso de um cabo coaxial, cabo de fibra óptica, par trançado, linha de inscrição digital (DSL) ou tecnologias sem fio como infravermelho, rádio e micro-onda, então, o cabo coaxial, o cabo de fibra óptica, o par trançado, a DSL ou as tecnologias sem fio como infravermelho, rádio e micro-onda estão incluídos na definição de mídia. Deve ser entendido, no entanto, que as mídias de armazenamento legíveis por computador e as mídias de armazenamento de dados não incluem conexões, ondas transportadoras, sinais ou outras mídias transitórias, mas são, em vez disso, direcionadas para mídias não transientes e tangíveis. O disco magnético e o disco óptico, conforme usados no presente documento, incluem disco compacto (CD), disco laser, disco óptico, disco versátil digital (DVD), disquete e disco blu-ray, em que os discos magnéticos, em geral, reproduzem os dados de modo magnético, enquanto os discos ópticos reproduzem os dados de modo óptico com lasers. As combinações dos supracitados também devem ser incluídas no escopo de mídia legível por computador.

[00183] As instruções podem ser executadas por um ou mais processadores, como um ou mais processadores de

sinal digital (DSPs), microprocessadores para propósitos gerais, circuitos integrados de aplicação específica (ASICs), matrizes lógicas programáveis por campo (FPGAs) ou outro conjunto de circuitos lógicos equivalentes integrados ou discretos. Portanto, o termo "processador", conforme usado no presente documento pode se referir a qualquer uma das estruturas supracitadas ou qualquer outra estrutura adequada para a implantação das técnicas descritas no presente documento. Adicionalmente, em alguns aspectos, a funcionalidade descrita no presente documento pode ser fornecida dentro de módulos dedicados de hardware e/ou software configurados para criptar e decodificar ou incorporados em um codec combinado. Além disso, as técnicas podem ser totalmente implantadas em um ou mais circuitos ou elementos lógicos.

[00184] As técnicas desta revelação podem ser implantadas em uma ampla variedade de dispositivos ou aparelhos, incluindo um monofone, um circuito integrado (IC) ou um conjunto de ICs (por exemplo, um conjunto de chips). Vários componentes, módulos ou unidades são descritos nesta revelação para enfatizar os aspectos funcionais dos dispositivos configurados para realizar as técnicas reveladas, mas não exigem necessariamente a realização por diferentes unidades de hardware. Em vez disso, conforme descrito acima, várias unidades podem ser combinadas em uma unidade de hardware de codec ou fornecidas por uma coleção de unidades de hardware interoperativos, incluindo um ou mais processadores conforme descrito acima, em conjunto com software e/ou firmware adequados.

[00185] Vários exemplos foram descritos. Essas e outras implantações estão dentro do escopo das reivindicações a seguir.

REIVINDICAÇÕES

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

decodificar, a partir de um conjunto de parâmetros de vídeo (VPS) de um fluxo de bits de múltiplas camadas, dados que indicam uma configuração de pequenos blocos para uma i-ésima camada do fluxo de bits de múltiplas camadas, em que decodificar os dados que indicam a configuração de pequenos blocos compreende decodificar um primeiro elemento de sintaxe, `tiles_enabled_vps_flag[i]`, no VPS para a i-ésima camada do fluxo de bits de múltiplas camadas, em que um valor de `tiles_enabled_vps_flag[i]` igual a 1 indica que o valor de um segundo elemento de sintaxe, `tiles_enabled_flag`, incluído em cada conjunto de parâmetros de imagem (PPS), que é referido por pelo menos uma imagem da i-ésima camada sendo igual a 1, em que um valor de `tiles_enabled_flag` igual a 1 indica que pequenos blocos estão disponíveis para imagens referentes àquele PPS; e

decodificar o fluxo de bits de múltiplas camadas de acordo com os dados decodificados a partir do VPS.

2. Método, de acordo com a reivindicação 1, caracterizado pelo fato de que decodificar o VPS compreende decodificar o VPS antes de uma sequência de imagens do fluxo de bits de múltiplas camadas, e em que decodificar o fluxo de bits de múltiplas camadas compreende decodificar a sequência de imagens de acordo com os dados decodificados a partir do VPS.

3. Método de codificação de dados de vídeo caracterizado pelo fato de que compreende:

codificar, em um conjunto de parâmetros de vídeo (VPS) de um fluxo de bits de múltiplas camadas, dados que indicam uma configuração de pequenos blocos para uma i-ésima camada do fluxo de bits de múltiplas camadas, em que

codificar os dados que indicam a configuração de pequenos blocos compreende codificar um primeiro elemento de sintaxe, `tiles_enabled_vps_flag[i]`, no VPS, para a *i*-ésima camada do fluxo de bits de múltiplas camadas, em que um valor de `tiles_enabled_vps_flag[i]` indica que o valor de um segundo elemento de sintaxe, `tiles_enabled_flag`, incluído em cada conjunto de parâmetros de imagem (PPS), que é referido por pelo menos uma imagem da *i*-ésima camada, indica que pequenos blocos estão disponíveis para imagens referentes àquele PPS; e

codificar o fluxo de bits de múltiplas camadas inclui codificar o VPS do fluxo de bits de múltiplas camadas.

4. Método, de acordo com a reivindicação 1 ou 3, caracterizado pelo fato de que decodificar ou codificar os dados que indicam a configuração de pequenos blocos compreende adicionalmente decodificar ou codificar um elemento de sintaxe `loop_filter_across_tiles_disabled_vps_flag` que indica uma configuração de filtro de malha (*loop filter*) para cada conjunto de parâmetros de imagem referido por pelo menos uma imagem da respectiva camada.

5. Método, de acordo com a reivindicação 1 ou 3, caracterizado pelo fato de que compreende adicionalmente atribuir recursos de codificação de um codificador ou decodificador de vídeo de acordo com os dados do VPS.

6. Método, de acordo com a reivindicação 3, caracterizado pelo fato de que codificar o VPS compreende codificar o VPS antes de codificar uma sequência de imagens do fluxo de bits de múltiplas camadas, e em que codificar o fluxo de bits de múltiplas camadas compreende codificar a sequência de imagens.

7. Aparelho que processa dados de vídeo caracterizado pelo fato de que compreende:

meios para decodificar, a partir de um conjunto de parâmetros de vídeo (VPS) de um fluxo de bits de múltiplas camadas, dados que indicam uma configuração de pequenos blocos para uma i-ésima camada do fluxo de bits de múltiplas camadas, em que decodificar os dados que indicam a configuração de pequenos blocos compreende decodificar um primeiro elemento de sintaxe, `tiles_enabled_vps_flag[i]`, no VPS, para a i-ésima camada do fluxo de bits de múltiplas camadas, em que um valor de `tiles_enabled_vps_flag[i]` igual a 1 indica que o valor de um segundo elemento de sintaxe, `tiles_enabled_flag`, incluído em cada conjunto de parâmetros de imagem (PPS), que é referido por pelo menos uma imagem da i-ésima camada sendo igual a 1, em que um valor de `tiles_enabled_flag` igual a 1 indica que pequenos blocos estão disponíveis para imagens referentes àquele PPS; e

meios para decodificar o fluxo de bits de múltiplas camadas de acordo com os dados do VPS.

8. Aparelho que processa dados de vídeo caracterizado pelo fato de que compreende:

meios para codificar, em um conjunto de parâmetros de vídeo (VPS) de um fluxo de bits de múltiplas camadas, dados que indicam uma configuração de pequenos blocos para uma i-ésima camada do fluxo de bits de múltiplas camadas, em que codificar os dados que indicam a configuração de pequenos blocos compreende codificar um primeiro elemento de sintaxe, `tiles_enabled_vps_flag[i]`, no VPS, para a i-ésima camada do fluxo de bits de múltiplas camadas, em que um valor de `tiles_enabled_vps_flag[i]` indica que o valor de um segundo elemento de sintaxe, `tiles_enabled_flag`, incluído em cada conjunto de parâmetros de imagem (PPS), que é referido por pelo menos uma imagem da i-ésima camada indica que pequenos blocos estão disponíveis para imagens referentes àquele PPS; e

meios para codificar o fluxo de bits de múltiplas camadas incluindo o VPS do fluxo de bits de múltiplas camadas.

9. Memória legível por computador caracterizada pelo fato de que possui instruções nela armazenadas que, quando executadas, fazem com que um computador realizem o método conforme definido em qualquer uma das reivindicações 1 a 6.

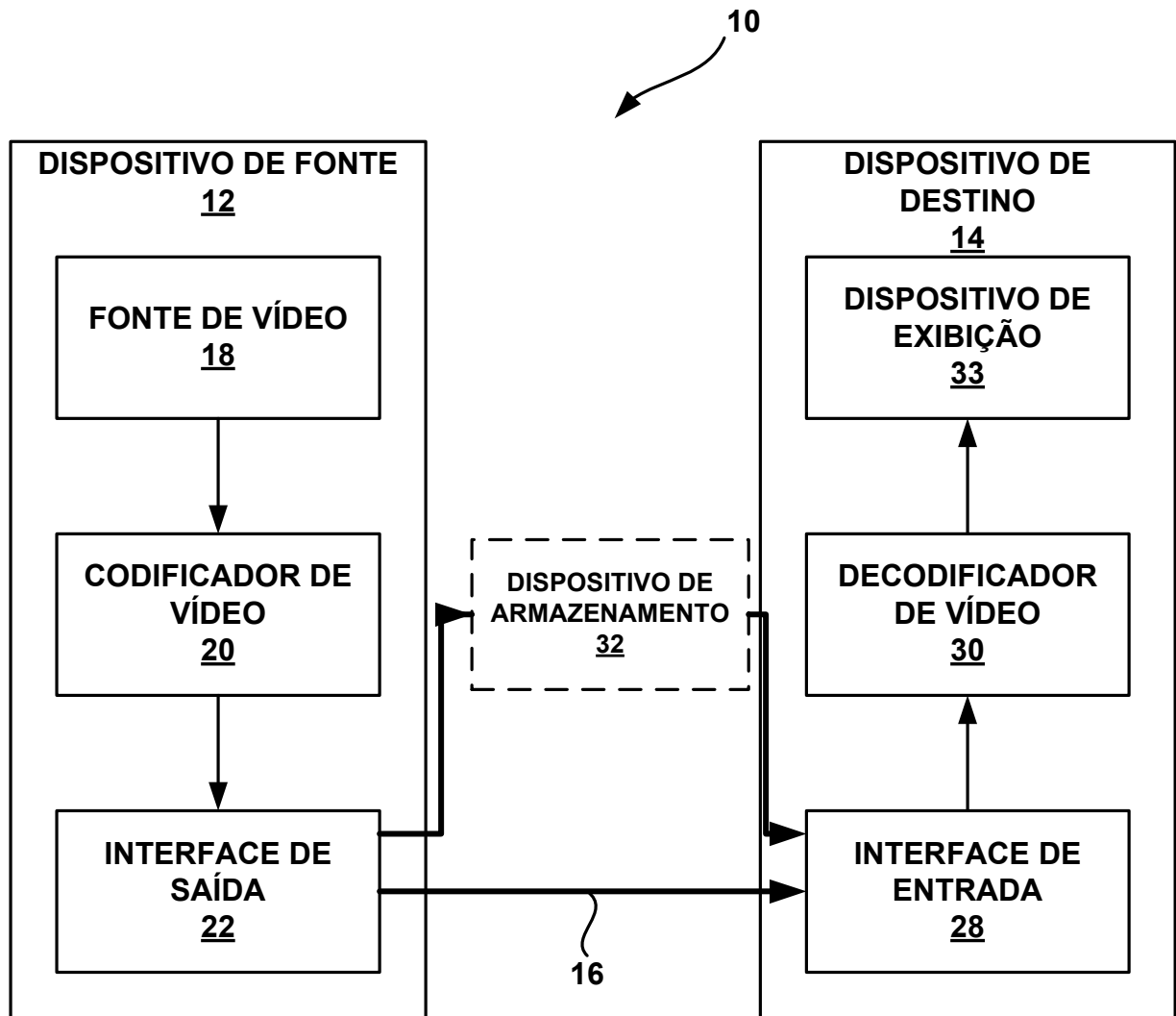


FIG. 1

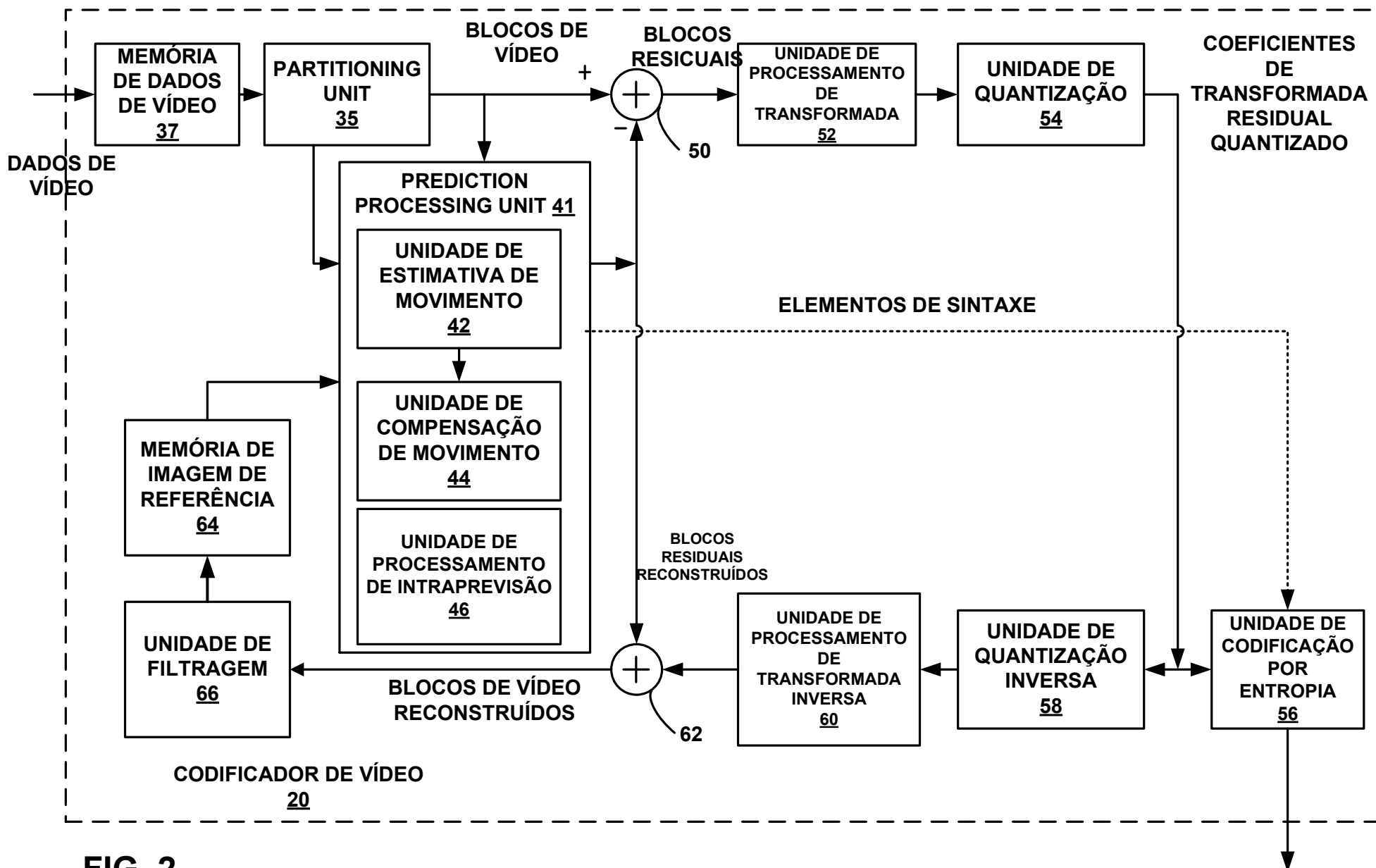


FIG. 2

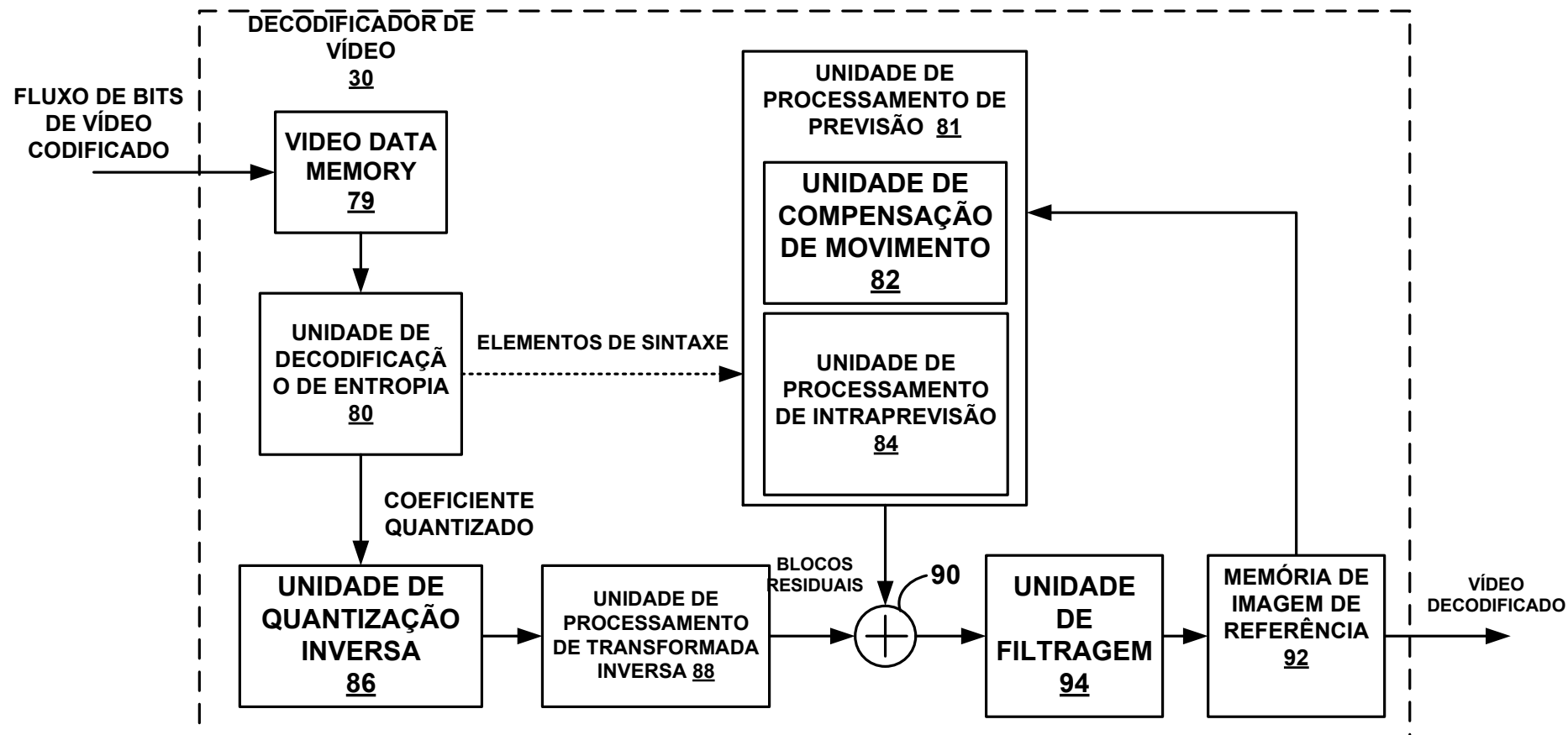


FIG. 3

	A	B	C	D	E	F	G
	CTB 0 FILEIRA 0	CTB 1 FILEIRA 0	CTB 2 FILEIRA 0	CTB 3 FILEIRA 0	CTB 4 FILEIRA 0	CTB 5 FILEIRA 0	CTB 6 FILEIRA 0
	CTB 0 FILEIRA 1	CTB 1 FILEIRA 1	CTB 2 FILEIRA 1	CTB 3 FILEIRA 1	CTB 4 FILEIRA 1	CTB 5 FILEIRA 1	CTB 6 FILEIRA 1
A	B	C	D	E	F	G	

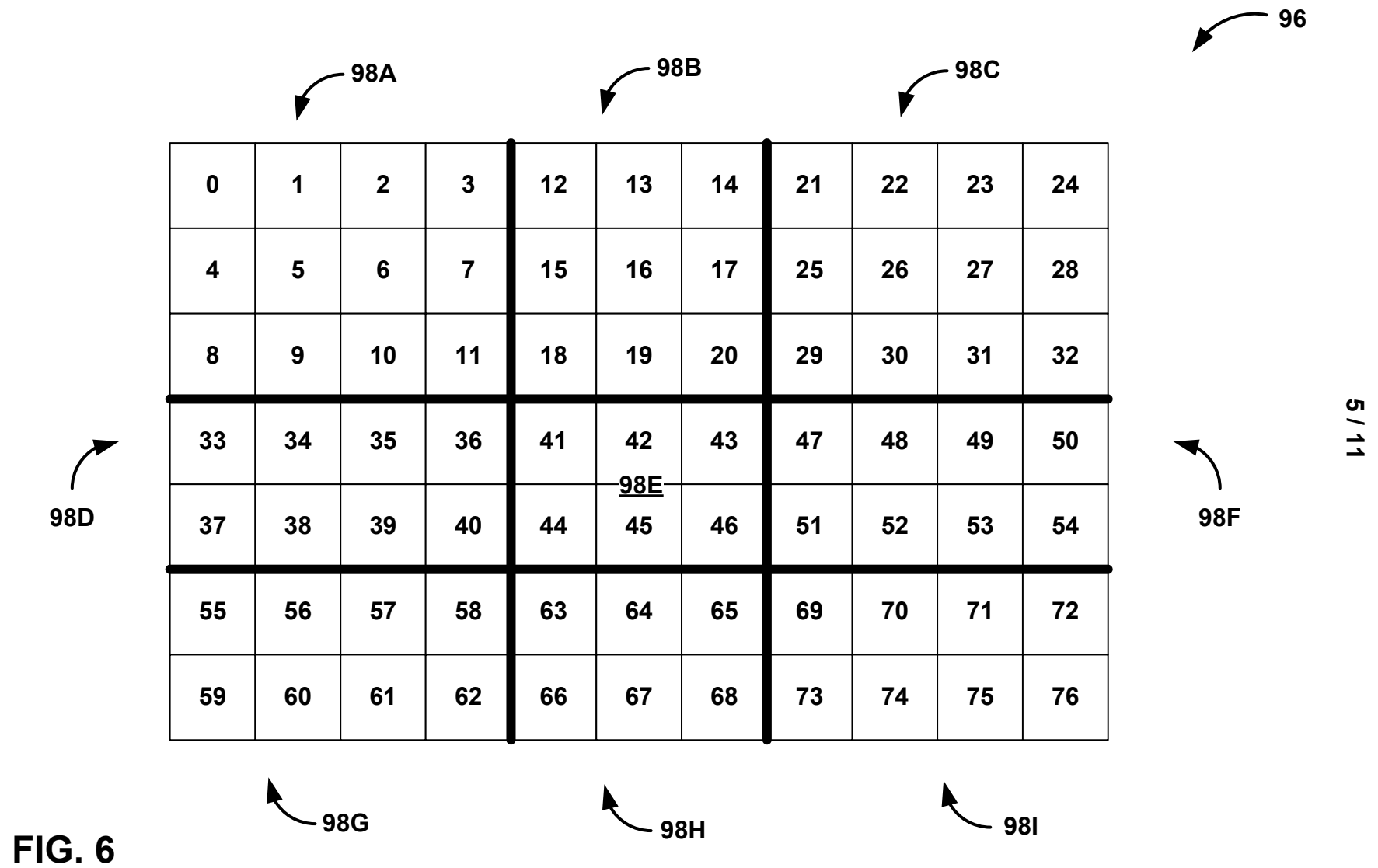
FIG. 4A

		A	B	C	D	E	F
	CTB 0 FILEIRA 0	CTB 1 FILEIRA 0	CTB 2 FILEIRA 0	CTB 3 FILEIRA 0	CTB 4 FILEIRA 0	CTB 5 FILEIRA 0	CTB 6 FILEIRA 0
	CTB 0 FILEIRA 1	CTB 1 FILEIRA 1	CTB 2 FILEIRA 1	CTB 3 FILEIRA 1	CTB 4 FILEIRA 1	CTB 5 FILEIRA 1	CTB 6 FILEIRA 1
A	B	C	D	E	F		

FIG. 4B

		A'					
	CTB 0 FILEIRA 0	CTB 1 FILEIRA 0	CTB 2 FILEIRA 0	CTB 3 FILEIRA 0	CTB 4 FILEIRA 0	CTB 5 FILEIRA 0	CTB 6 FILEIRA 0
A'	CTB 0 FILEIRA 1	CTB 1 FILEIRA 1	CTB 2 FILEIRA 1	CTB 3 FILEIRA 1	CTB 4 FILEIRA 1	CTB 5 FILEIRA 1	CTB 6 FILEIRA 1
	CTB 0 FILEIRA 2	CTB 1 FILEIRA 2	CTB 2 FILEIRA 2	CTB 3 FILEIRA 2	CTB 4 FILEIRA 2	CTB 5 FILEIRA 2	CTB 6 FILEIRA 2
B'	CTB 0 FILEIRA 3	CTB 1 FILEIRA 3	CTB 2 FILEIRA 3	CTB 3 FILEIRA 3	CTB 4 FILEIRA 3	CTB 5 FILEIRA 3	CTB 6 FILEIRA 3

FIG. 5



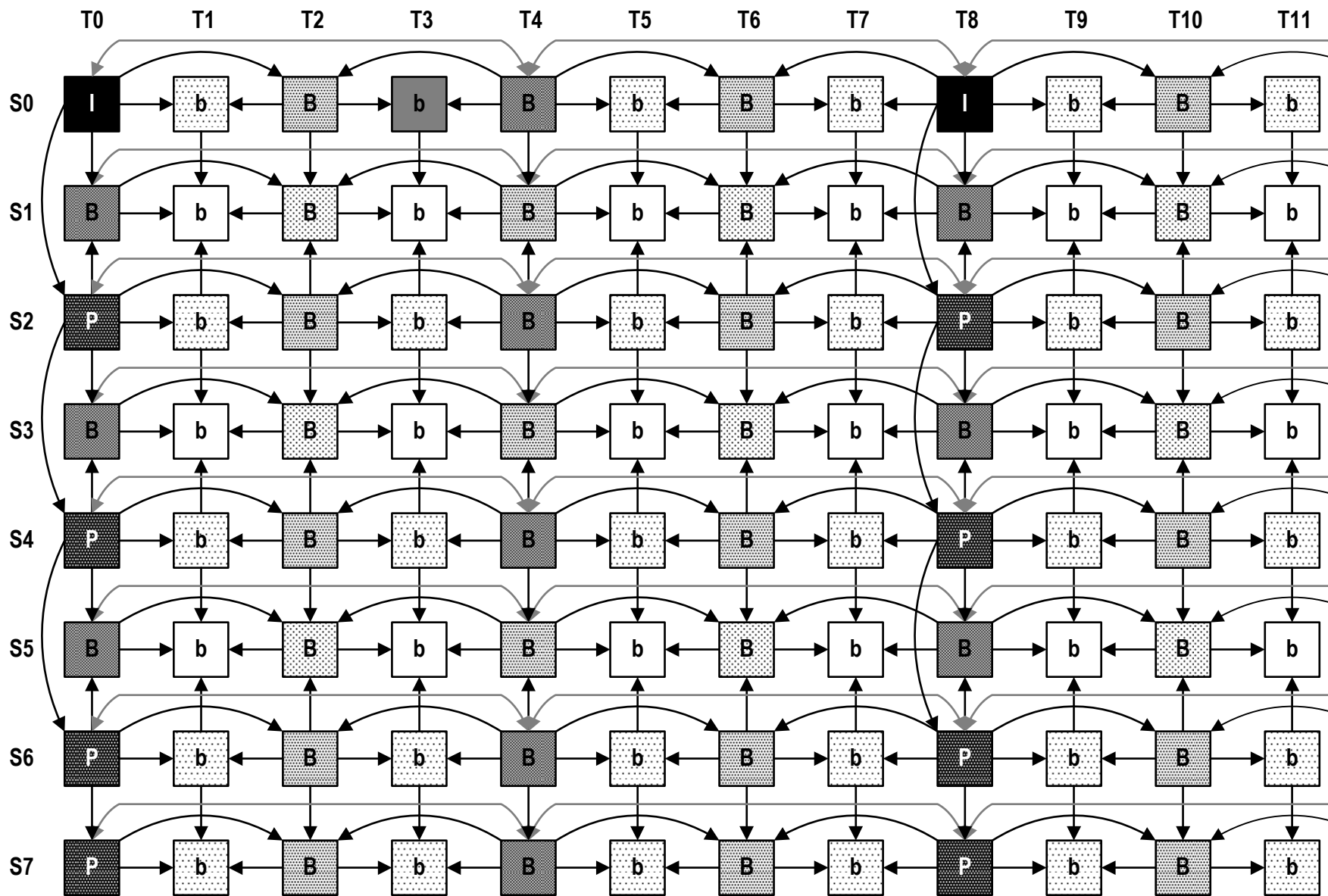


FIG. 7

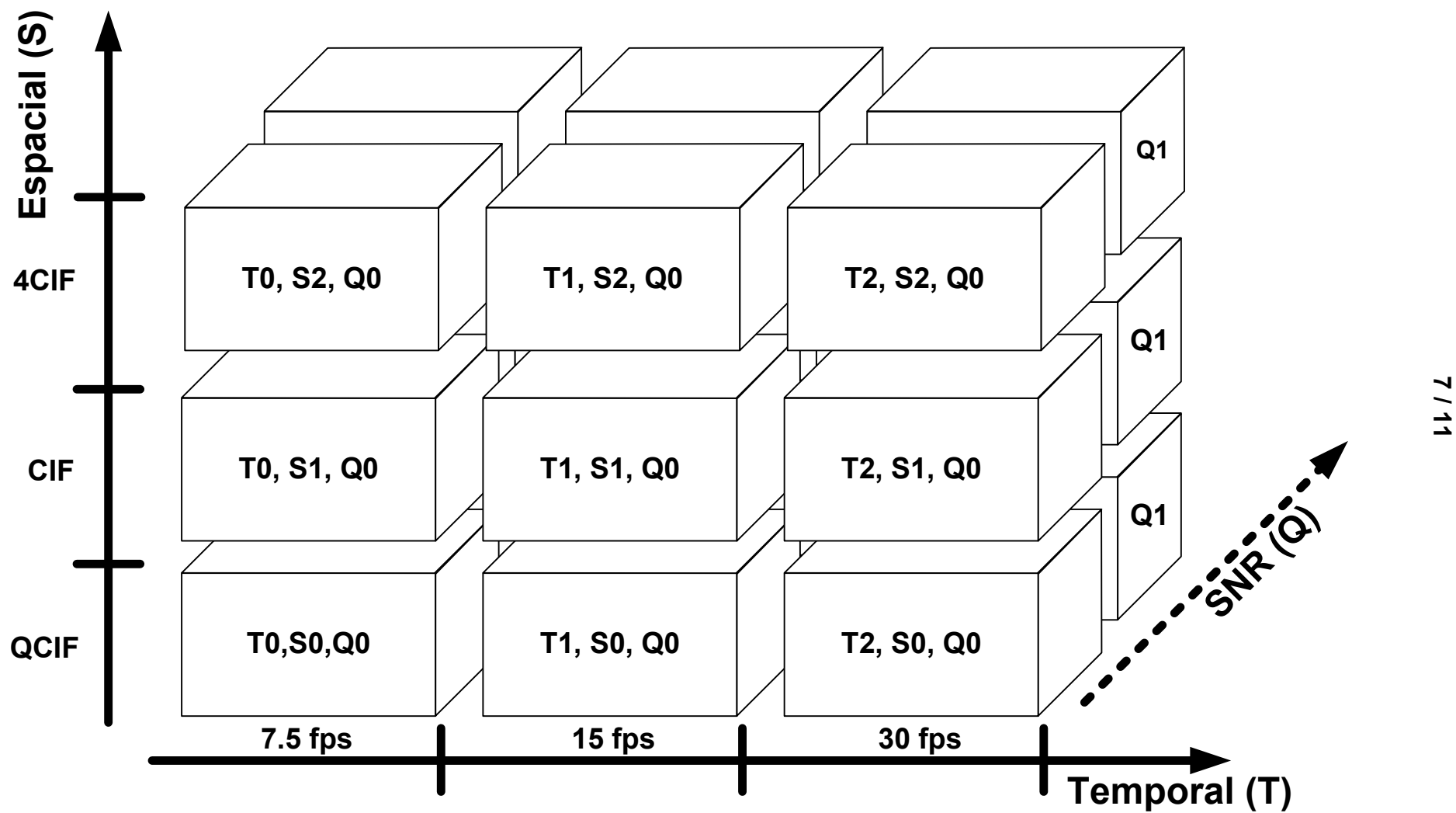


FIG. 8

CAMADA 0: QCIF, 7.5 HZ, 64 KBPS

CAMADA 1: QCIF, 15 HZ, 96 KBPS

CAMADA 2: CIF, 15 HZ, 256 KBPS

CAMADA 3: CIF, 15 HZ, 512 KBPS

CAMADA 4: 4CIF, 30 HZ, 1.0 MBPS

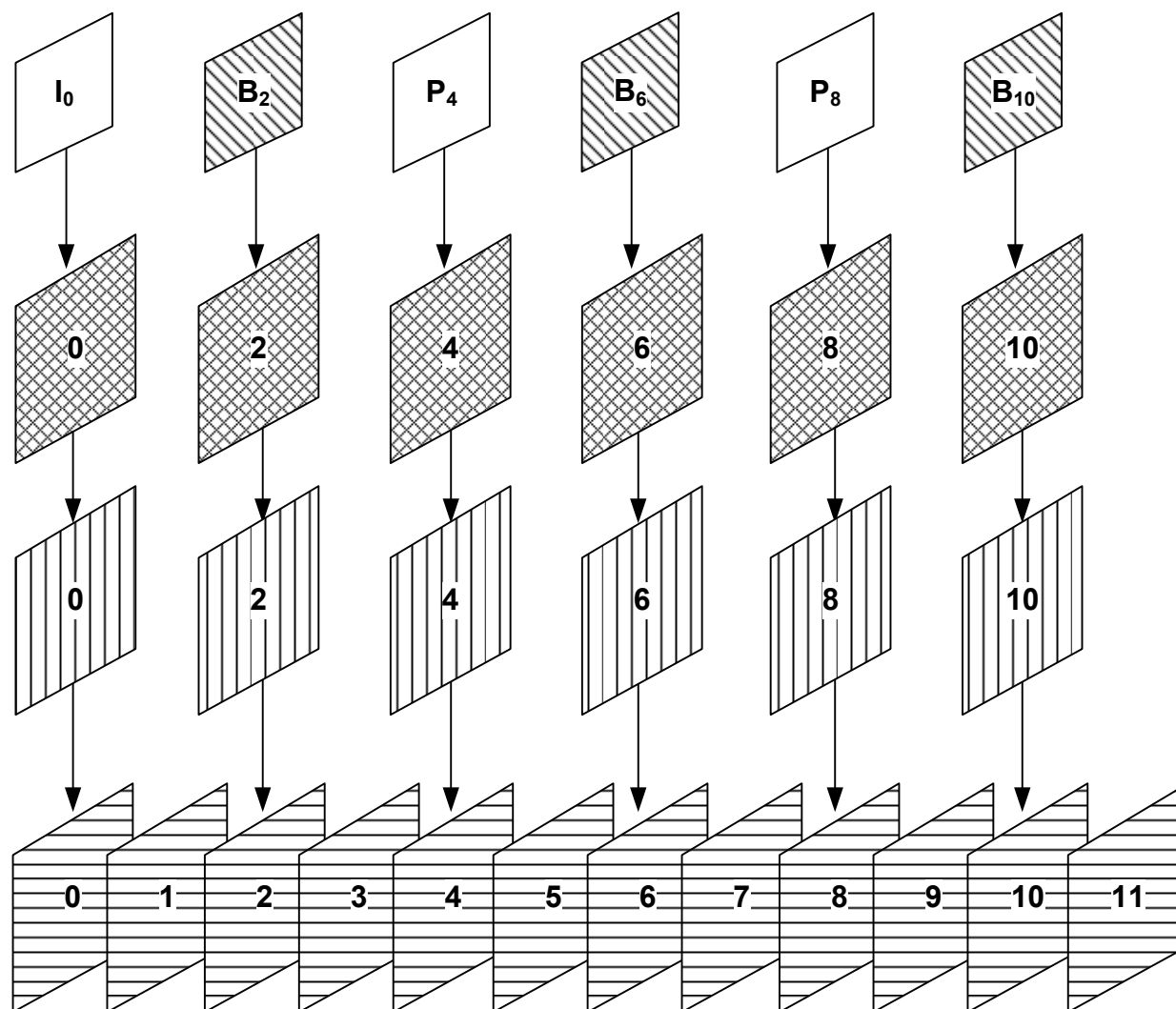


FIG. 9

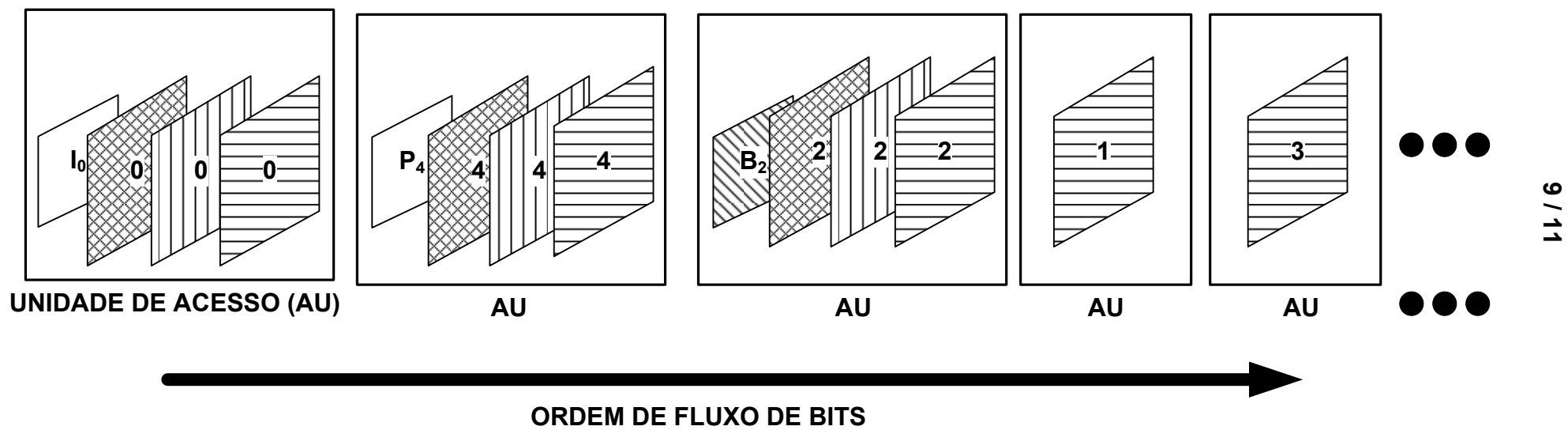


FIG. 10



FIG. 11

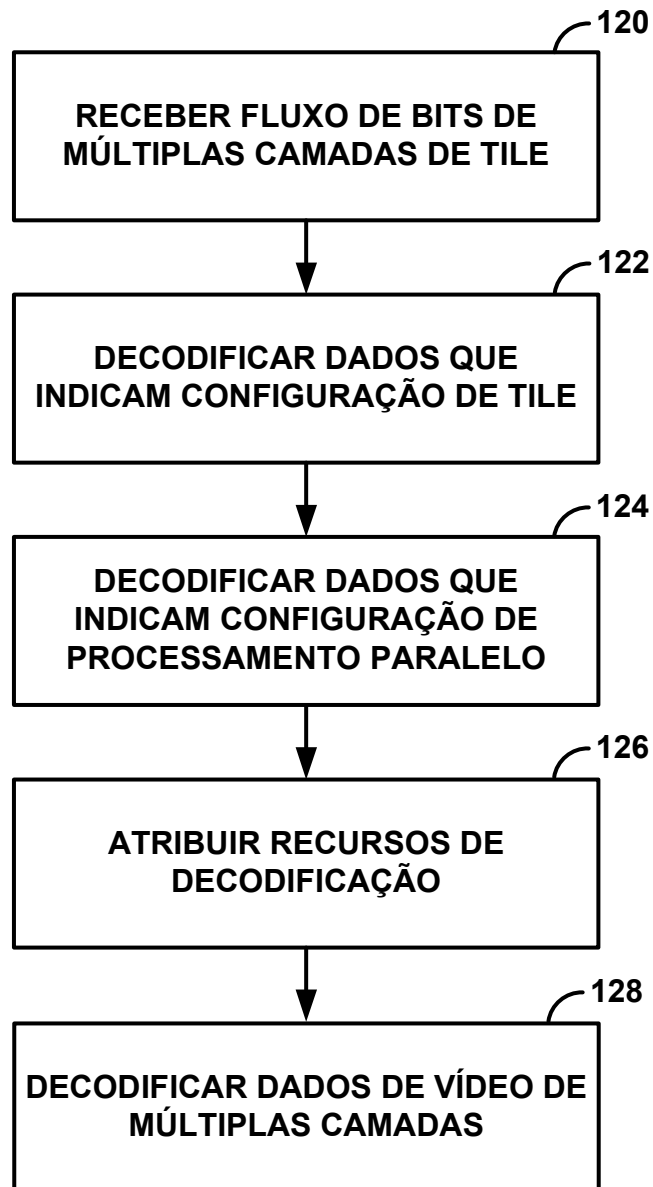


FIG. 12