República Federativa do Brasil

Ministério do Desenvolvimento, Indústria, Comércio e Serviços

Instituto Nacional da Propriedade Industrial

(11) BR 112016022261-0 B1

(22) Data do Depósito: 25/03/2015

(45) Data de Concessão: 12/12/2023

(54) Título: MÉTODOS DE CODIFICAÇÃO E DECODIFICAÇÃO DE DADOS DE VÍDEO, APARELHOS CONFIGURADOS PARA CODIFICAR E DECODIFICAR DADOS DE VÍDEO, E, MEMÓRIA LEGÍVEL POR COMPUTADOR

(51) Int.Cl.: H04N 19/27; H04N 19/593; H04N 19/93.

(52) CPC: H04N 19/27; H04N 19/593; H04N 19/93.

(30) Prioridade Unionista: 17/04/2014 US 61/981,105; 20/06/2014 US 62/015,327; 23/05/2014 US 62/002,668; 24/03/2015 US 14/667,411; 24/08/2014 US 62/041,119; (...).

(73) Titular(es): QUALCOMM INCORPORATED.

(72) Inventor(es): MARTA KARCZEWICZ; WEI PU; VADIM SEREGIN; RAJAN LAXMAN JOSHI; JOEL SOLE ROJALS; FENG ZOU.

(86) Pedido PCT: PCT US2015022468 de 25/03/2015

(87) Publicação PCT: WO 2015/148652 de 01/10/2015

(85) Data do Início da Fase Nacional: 26/09/2016

(57) Resumo: SINALIZAÇÃO DE PREDITOR DE PALETA COM CÓDIGO RUN LENGTH PARA CODIFICAÇÃO DE VÍDEO. Técnicas para codificar um vetor de predição binário para predizer uma paleta para codificação de vídeo baseada em paleta são descritas. Em um exemplo, um método de decodificação de vídeo compreende receber um vetor de predição binário codificado para um bloco atual de dados de vídeo, decodificar o vetor de predição binário codificado usando uma técnica de decodificação run length, gerar uma paleta para o bloco atual de dados de vídeo baseado no vetor de predição de binário, o vetor de predição binário compreendendo entradas indicando se ou não entradas de paleta previamente utilizadas são reutilizadas para a paleta para o bloco atual de dados de vídeo e decodificação do bloco atual de dados de vídeo usando a paleta.

"MÉTODOS DE CODIFICAÇÃO E DECODIFICAÇÃO DE DADOS DE VÍDEO, APARELHOS CONFIGURADOS PARA CODIFICAR E DECODIFICAR DADOS DE VÍDEO, E, MEMÓRIA LEGÍVEL POR COMPUTADOR"

[0001] Este pedido reivindica o benefício do Pedido Provisório U.S. No. 61/970,257, depositado em 25 de março de 2014, Pedido Provisório U.S. No. 61/981,105, depositado em 17 de abril de 2014, Pedido Provisório U.S. No. 62 /002,668 depositado em 23 de maio de 2014, Pedido Provisório U.S. No. 62/015,327, depositado em junho 20, 2014, Pedido Provisório U.S. No. 62/018,461, depositado em 27 de junho de 2014, e Pedido Provisório US No. 62/041,119, depositado em 24 de agosto de 2014, todo o conteúdo de cada um dos quais é incorporado aqui por referência.

CAMPO TÉCNICO

[0002] Esta descrição refere-se à codificação e decodificação de vídeo.

FUNDAMENTOS

[0003] Capacidades de vídeo digitais podem ser incorporadas em uma ampla gama de dispositivos, incluindo televisores digitais, sistemas de broadcast direto digitais, sistemas de transmissão sem fio, assistentes digitais pessoais (PDAs), computadores laptop ou desktop, tablets, e-books, câmeras digitais, dispositivos de gravação digital, reprodutores de mídia digital, dispositivos de jogos de vídeo, consoles de jogos de vídeo, telefones celulares de rádio ou satélite, os chamados "smartphone", aparelhos de vídeo teleconferência, dispositivos de streaming de vídeo, e similares. Dispositivos de vídeo digital implementam as técnicas de compressão de vídeo, tais como as descritas nos padrões definidos pela MPEG-2, MPEG-4, a ITU-T H.263, a ITU-T H.264 / MPEG-4, Parte 10, Codificação de Vídeo Avançada (AVC), o padrão de vídeo de alta eficiência de codificação (HEVC)

atualmente em desenvolvimento, e a extensão de tais normas. Os dispositivos de vídeo podem transmitir, receber, codificar, decodificar e/ou armazenar informações de vídeo digital de forma mais eficiente através da implementação de tais técnicas de compressão de vídeo.

[0004] As técnicas de compressão de vídeo podem executar predição espacial (intraimagem) e/ou predição temporal (interimagem) para reduzir ou eliminar a redundância inerente em sequências de vídeo. Para a codificação de vídeo baseada em blocos, uma fatia de vídeo (isto é, um quadro de vídeo ou uma porção de um quadro de vídeo) pode ser dividida em blocos de vídeo. Blocos de vídeo em uma fatia intracodificada (I) de uma imagem são codificados usando a predição espacial em relação às amostras de referência em blocos vizinhos no mesmo quadro. Blocos de vídeo em uma fatia intercodificada (P ou B) de uma imagem espacial pode utilizar a predição no que diz respeito às amostras de referência em blocos vizinhos no mesmo quadro, ou a predição temporal com respeito às amostras de referência em outros quadros de referência. As imagens podem ser referidas como quadros, e imagens de referência podem ser referidas como quadros de referência.

[0005] Predição espacial ou temporal resulta em um bloco preditivo para um bloco a ser codificado. Dados residuais representam diferenças de pixel entre o bloco original a ser codificado e o bloco preditivo. Um bloco intercodificado é codificado de acordo com um vetor de movimento que aponta para um bloco de amostras de referência que formam o bloco de predição, e os dados residuais indicam a diferença entre o bloco codificado e o bloco de predição. Um bloco intracodificado é codificado de acordo com um modo de intracodificação e os dados residuais. Para a compressão adicional, os dados residuais

podem ser transformados a partir do domínio de pixel para um domínio de transformada, resultando em coeficientes residuais, que, em seguida, podem ser quantizados. Os coeficientes quantizados, inicialmente dispostos em uma matriz bidimensional, podem ser varridos, a fim de produzir um vetor unidimensional de coeficientes, e codificação de entropia pode ser aplicada para conseguir ainda mais compressão.

SUMÁRIO

[0006] Esta divulgação refere-se a técnicas de codificação e decodificação de vídeo. Em particular, esta divulgação descreve técnicas para codificação e decodificação de dados de vídeo com um modo de codificação baseada em paleta. Em um modo de codificação baseada em paleta, os valores de pixel para um bloco de dados de vídeo podem ser codificados em relação a uma paleta de valores de cor associada com o bloco de dados de vídeo. A paleta de valores de cor pode ser determinada por um codificador de vídeo e pode conter os valores de cor que são mais comuns para um bloco particular. O codificador de vídeo pode atribuir um índice para a paleta de valores de cor para cada pixel do bloco de dados de vídeo, e sinalizar tal um índice para um decodificador de vídeo. O decodificador de vídeo pode, então, usar o índice na paleta para determinar qual o valor de cor a ser utilizado para um pixel particular no bloco.

[0007] Além de sinalizar índices na paleta, um codificador de vídeo também pode transmitir a paleta no fluxo de bits de vídeo codificado. Técnicas para a transmissão da paleta podem incluir sinalizar explicitamente os valores da paleta, bem como predizer as entradas de paleta para um bloco atual a partir de entradas de paleta de um ou mais blocos previamente codificados.

Esta divulgação descreve técnicas para a sinalização de um ou mais elementos de sintaxe (por exemplo, um vetor de predição binário) que indicam quais entradas de paleta de blocos previamente codificados de dados de vídeo podem ser reutilizadas para um bloco atual de dados de vídeo. Esta divulgação descreve técnicas de codificação e de decodificação adicionais do vetor de predição binário.

[0008] Em um exemplo da divulgação, um método de decodificar dados de vídeo compreende receber um vetor de predição binário codificado para um bloco atual de dados de vídeo, decodificar o vetor de predição binário codificado usando uma técnica de decodificação run length, gerar uma paleta para o bloco atual de dados de vídeo com base no vetor de predição binário, o vetor de predição binário compreendendo entradas que indicam se ou não entradas de paleta previamente utilizadas são reutilizadas para a paleta para o bloco atual de dados de vídeo, e decodificar o bloco atual de dados de vídeo usando a paleta.

[0009] Em outro exemplo da divulgação, um método de codificar de dados de vídeo compreende gerar uma paleta para o bloco atual de dados de vídeo, gerar um vetor de predição binário para a paleta para o bloco atual de dados de vídeo, o vetor de predição binário compreendendo entradas indicando se ou não entradas de paleta previamente utilizadas são reutilizadas para a paleta para o bloco atual de dados de vídeo, codificar o vetor de predição binário usando uma técnica de codificação run length, e codificar o bloco atual de dados de vídeo usando a paleta.

[0010] Em um outro exemplo da divulgação, um aparelho configurado para decodificar os dados de vídeo compreende uma memória configurada para armazenar os dados de vídeo, e um decodificador de vídeo em comunicação com a memória configurada para receber um vetor de predição

binário codificado para um bloco atual dos dados de vídeo, decodificar o vetor de predição binário codificado usando uma técnica de decodificação run length, gerar uma paleta para o bloco atual de dados de vídeo com base no vetor de predição binário, o vetor de predição binário compreendendo entradas indicando se ou não entradas de paleta previamente utilizadas são reutilizadas para a paleta para o bloco atual de dados de vídeo e decodificar o bloco atual de dados de vídeo usando a paleta.

[0011] Em um outro exemplo da divulgação, um aparelho configurado para codificar os dados de vídeo compreende uma memória configurada para armazenar os dados de vídeo de um codificador de vídeo em comunicação com a memória configurada para gerar uma paleta para um bloco atual dos dados de vídeo, gerar um vetor de predição binário para a paleta para o bloco atual de dados de vídeo, o vetor de predição binário compreendendo entradas indicando se ou não entradas de paleta previamente utilizadas são reutilizadas para a paleta para o bloco atual de dados de vídeo, codificar o vetor de predição binário usando uma técnica de codificação run length e codificar o bloco atual de dados de vídeo usando a paleta.

[0012] Em um outro exemplo da divulgação, um aparelho configurado para decodificar os dados de vídeo compreende meios para receber um vetor de predição binário codificado para um bloco atual de dados de vídeo, meios para decodificar o vetor de predição binário codificado usando uma técnica de decodificação run length, meios para gerar uma paleta para o bloco atual de dados de vídeo com base no vetor de predição binário, o vetor de predição binário compreendendo entradas indicando se ou não entradas de paleta previamente utilizadas são reutilizadas para a paleta para o bloco atual de dados de vídeo, e meios para

decodificar o bloco atual de dados de vídeo usando a paleta.

[0013] Em um outro exemplo da divulgação, um aparelho configurado para codificar os dados de vídeo compreende meios para gerar uma paleta para o bloco atual de dados de vídeo, meios para gerar um vetor de predição binário para a paleta para o bloco atual de dados de vídeo, o vetor de predição binário compreendendo entradas que indicam se ou não entradas de paleta previamente utilizadas são reutilizadas para a paleta para o bloco atual de dados de vídeo, meios para codificar o vetor de predição binário usando uma técnica de codificação run length, e meios para codificar o bloco atual de dados de vídeo usando a paleta.

[0014] Em um outro exemplo, esta divulgação descreve um meio de armazenamento legível por computador que armazena as instruções que, quando executadas, fazem com que um ou mais processadores de um dispositivo configurado para decodificar os dados de vídeo recebam um vetor de predição binário codificado para um bloco atual dos dados de vídeo, decodifiquem o vetor de predição binário codificado usando uma técnica de decodificação run length, gerem uma paleta para o bloco atual de dados de vídeo com base no vetor de predição binário, o vetor de predição binário compreendendo entradas indicando se ou não entradas de paleta previamente utilizadas são reutilizadas para a paleta para o bloco atual de dados de vídeo e decodifiquem o bloco atual de dados de vídeo usando a paleta.

[0015] Em um outro exemplo, esta divulgação descreve um meio de armazenamento legível por computador que armazena as instruções que, quando executadas, fazem com que um ou mais processadores de um dispositivo configurado para codificar os dados de vídeo gerem uma

paleta para um bloco atual dos dados de vídeo, gerem um vetor de predição binário para a paleta para o bloco atual de dados de vídeo, o vetor de predição binário compreendendo entradas indicando se ou não entradas de paleta previamente utilizadas são reutilizadas para a paleta para o bloco atual de dados de vídeo, codifiquem o vetor de predição binário usando uma técnica de codificação run-length, e codifiquem o bloco atual de dados de vídeo usando a paleta.

[0016] Os detalhes de um ou mais exemplos são apresentados nos desenhos acompanhantes e na descrição abaixo. Outras características, objetos e vantagens serão evidentes a partir da descrição e desenhos, e a partir das reivindicações.

BREVE DESCRIÇÃO DOS DESENHOS

[0017] A figura 1 é um diagrama de blocos que ilustra um sistema de codificação de vídeo exemplar que pode implementar as técnicas descritas nesta divulgação.

[0018] A figura 2 é um diagrama de blocos que ilustra um codificador de vídeo exemplar que pode implementar as técnicas descritas nesta divulgação.

[0019] A figura 3 é um diagrama de blocos que ilustra um exemplo de decodificador de vídeo que pode implementar as técnicas descritas nesta divulgação.

[0020] A figura 4 é um diagrama de blocos que ilustra uma unidade de codificação baseada em paleta exemplar do codificador de vídeo da figura 2.

[0021] A figura 5 é um diagrama conceitual ilustrando uma técnica de predição de paleta exemplar de acordo com as técnicas da presente descrição.

[0022] A figura 6 é um diagrama conceitual que ilustra um exemplo de técnica de codificação de vetor de

predição binário de acordo com as técnicas da presente descrição.

[0023] A figura 7 é um diagrama de blocos que ilustra uma unidade de decodificação baseada em paleta exemplar do codificador de vídeo da figura 3.

[0024] A figura 8 é um fluxograma que ilustra um exemplo do método de codificação da divulgação.

[0025] A figura 9 é um fluxograma que ilustra um exemplo do método de decodificação da divulgação.

[0026] A figura 10 é um diagrama conceitual que ilustra um exemplo de determinar as entradas de paleta para codificação de vídeo baseada em paleta, de acordo com as técnicas desta divulgação.

DESCRIÇÃO DETALHADA

[0027] Esta divulgação inclui técnicas de codificação e compressão de vídeo. Em particular, esta divulgação descreve técnicas para a codificação baseada em paleta de dados de vídeo. Na codificação de vídeo tradicional, assume-se que as imagens são de tons contínuos e espacialmente suaves. Com base nessas premissas, várias ferramentas foram desenvolvidas, tais como o transformada baseada em bloco, filtragem, etc., e essas ferramentas têm mostrado bom desempenho para vídeos de conteúdo naturais.

[0028] No entanto, em aplicações como desktop remoto, trabalho colaborativo e exibição sem fio, o conteúdo de tela gerado por computador (por exemplo, como texto ou gráficos de computador) pode ser o conteúdo dominante a ser compactado. Este tipo de conteúdo tende a ter tom discreto e linhas nítidas de recursos, e limites de objetos de alto contraste. A suposição de tom contínuo e suavidade já não pode ser aplicada ao conteúdo de tela, e assim as técnicas de codificação de vídeo tradicionais

podem não ser formas eficientes para comprimir dados de vídeo que incluem o conteúdo de tela.

[0029] Técnicas de codificação baseada em paleta podem ser particularmente adequadas para a codificação de conteúdo gerado de tela. Por exemplo, assumindo que uma área particular de dados de vídeo tem um número relativamente pequeno de cores. Um codificador de vídeo (um codificador de vídeo ou decodificador de vídeo) pode codificar uma assim chamada "paleta", tal como uma tabela de cores para representar os dados de vídeo da área particular (por exemplo, um dado bloco). Cada pixel pode ser associado com uma entrada na paleta que representa a cor do pixel. Por exemplo, o codificador de vídeo pode codificar um índice que mapeia o valor de pixel para a entrada apropriada na paleta.

[0030] As técnicas de codificação baseada em paleta de dados de vídeo podem ser utilizadas com uma ou mais outras técnicas de codificação, tais como as técnicas de codificação inter- ou intrapreditivas. Por exemplo, como descrito em maior detalhe abaixo, um codificador ou decodificador, ou codificador-decodificador combinado (codec), pode ser configurado para executar a codificação inter e intrapreditiva, assim como a codificação baseada em paleta.

[0031] Em alguns exemplos, as técnicas de codificação baseada em paleta podem ser configuradas para uso com as uma ou mais normas de codificação de vídeo. Alguns exemplos de normas de codificação de vídeo incluem ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 ou ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (também conhecido como ISO/IEC MPEG-4 AVC), incluindo a suas extensões de Codificação de Vídeo Escalonável (SVC) e Codificação de Vídeo Multivista (MVC). Em outro exemplo,

técnicas de codificação baseada em paleta podem ser configuradas para uso com Codificação de Vídeo de Alta Eficiência (HEVC). HEVC é um novo padrão de codificação de vídeo desenvolvido pela Joint Collaboration Team on Video Coding (JCT-VC) de ITU-T Video Coding Experts Group (VCEG) e ISO/IEC Motion Picture Experts Group (MPEG).

[0032] Recentemente, o projeto de HEVC foi finalizado pela Joint Collaboration Team on Video Coding (JCT-VC) de ITU-T Video Coding Experts Group (VCEG) e ISO/IEC Motion Picture Experts Group (MPEG). A especificação mais recente de HEVC, referida como HEVC Versão 1 a seguir, é descrita em "ITU-T H.265 (V1)", que como de 24 de Março de 2015 está disponível a partir de http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11885&lang=en. Documento ITU-T H.265, SÉRIE H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS, Infraestrutura de Serviços Audiovisuais - Codificação de Vídeo em Movimento, "High Efficiency Video Coding", abril de 2013 também descreve o padrão HEVC. Uma especificação recente de extensões de distribuição, referida como RExt a seguir, é descrita em "ITU-T H.265 (V2)", que como de 24 de Março de 2015 está disponível a partir de http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=12296&lang=en.

[0033] Ao utilizar técnicas de codificação baseada em paleta exemplares, um codificador de vídeo pode codificar um bloco de dados de vídeo através da determinação de uma paleta para o bloco (por exemplo, codificando a paleta explicitamente, predizendo-a, ou uma combinação dos mesmos), localizar uma entrada na paleta para representar o valor de cada pixel, e codificar o bloco com os valores de índice para os pixels em relação ao valor de pixel para a paleta. Um decodificador de vídeo pode

obter, a partir de um fluxo de bits codificado, uma paleta para um bloco, assim como os valores de índice para os pixels do bloco. O decodificador de vídeo pode relacionar os valores de índice dos pixels de entradas da paleta para reconstruir os valores de pixel do bloco. O exemplo acima é destinado a prover uma descrição geral da codificação baseada em paleta.

[0034] Além de sinalização de valores de índice na paleta, um codificador de vídeo também pode transmitir a paleta no fluxo de bits de vídeo codificado. Técnicas para a transmissão da paleta podem incluir sinalizar explicitamente os valores da paleta, bem como predizer as entradas de paleta para um bloco atual a partir de entradas de paleta de um ou mais blocos previamente codificados. Técnicas de predição de paleta podem incluir sinalizar uma sequência de elementos de sintaxe (por exemplo, um vetor de predição binário) em que o valor de cada elemento de sintaxe no vetor de predição binário indica se ou não a entrada de paleta de um bloco previamente codificado de dados de vídeo pode ser reutilizada para um bloco atual de dados de vídeo. Tais vetores de predição binários podem ser longos para grandes tamanhos de paleta e/ou para casos em que um grande número de entradas de paleta é armazenado para possível utilização no processo de predição de paleta. Grandes vetores de predição binários podem resultar em níveis de taxa de bits inaceitáveis. Tendo em conta estas desvantagens, esta divulgação descreve técnicas para codificação e decodificação de vetores de predição binários em um processo de predição de paleta.

[0035] A figura 1 é um diagrama de blocos que ilustra um exemplo do sistema de codificação de vídeo 10 que podem utilizar as técnicas desta divulgação. Tal como aqui utilizado, o termo "codificador de vídeo" refere-se

genericamente a ambos os codificadores de vídeo e decodificadores de vídeo. Nesta divulgação, os termos "codificação de vídeo" ou "codificação" podem referir-se, genericamente, à codificação de vídeo ou decodificação de vídeo. O codificador de vídeo 20 e o decodificador de vídeo 30 do sistema de codificação de vídeo 10 representam exemplos de dispositivos que podem ser configurados para realizar as técnicas de codificação de vídeo baseada em paleta de acordo com vários exemplos descritos nesta divulgação. Por exemplo, o codificador de vídeo 20 e o decodificador de vídeo 30 podem ser configurados para codificar seletivamente vários blocos de dados de vídeo, tal como CU ou PU em codificação HEVC, usando a codificação baseada em paleta codificação ou não baseada em paleta. Modos de codificação não baseada em paleta podem referir-se a diferentes modos de codificação temporal preditiva interpreditiva ou modos de codificação espacial intrapreditiva, tal como os vários modos de codificação especificados por HEVC versão 1.

[0036] Como mostrado na figura 1, sistema de codificação de vídeo 10 inclui um dispositivo de origem 12 e um dispositivo de destino 14. O dispositivo de origem 12 gera dados de vídeo codificados. Por conseguinte, o dispositivo de origem 12 pode ser referido como um dispositivo de codificação de vídeo ou um aparelho de codificação de vídeo. O dispositivo de destino 14 pode decodificar os dados de vídeo codificados, gerados pelo dispositivo de origem 12. Consequentemente, o dispositivo de destino 14 pode ser referido como um dispositivo de decodificação de vídeo ou um aparelho de decodificação de vídeo. O dispositivo de origem 12 e o dispositivo de destino 14 podem ser exemplos de dispositivos de codificação de vídeo ou aparelhos de codificação de vídeo.

Várias implementações de dispositivo de origem 12, do dispositivo de destino 14, ou ambos, podem incluir um ou mais processadores e memória acoplados a um ou mais processadores. A memória pode incluir, mas não está limitada a RAM, ROM, EEPROM, memória flash, ou qualquer outro meio que possa ser utilizado para armazenar o código de programa desejado sob a forma de instruções ou estruturas de dados, e que pode ser acessado por um computador, como aqui descrito.

[0037] O dispositivo de origem 12 e dispositivo de destino 14 podem compreender uma ampla gama de dispositivos, incluindo computadores desktop, dispositivos de computação móvel, computadores notebook (por exemplo, laptop), computadores tablets, set-top boxes, aparelhos de telefone tal como os chamados "smartphones", televisores, câmeras, dispositivos de exibição, reprodutores de mídia digital, consoles de jogos de vídeo, computadores no carro, ou algo semelhante.

[0038] O dispositivo de destino 14 pode receber dados de vídeo codificados do dispositivo de origem 12 através de um canal 16. O canal 16 pode compreender uma ou mais mídias ou dispositivos capazes de retirar os dados de vídeo codificados do dispositivo de origem 12 para o dispositivo de destino 14. Em um exemplo, canal 16 pode compreender uma ou mais mídias de comunicação que permitem que dispositivo de origem 12 transmita dados de vídeo codificados diretamente para o dispositivo de destino 14 em tempo real. Neste exemplo, o dispositivo de origem 12 pode modular os dados de vídeo codificados de acordo com um padrão de comunicação, tal como um protocolo de comunicação sem fio, e pode transmitir os dados de vídeo modulados para o dispositivo de destino 14. As uma ou mais mídias de comunicação podem incluir meios de comunicação sem fio e/ou

com fio, como um espectro de frequência de rádio (RF) ou uma ou mais linhas de transmissão físicas. Os um ou mais meios de comunicação podem formar parte de uma rede baseada em pacotes, tal como uma rede de área local, uma rede de área ampla, ou uma rede global (por exemplo, a Internet). Os um ou mais meios de comunicação podem incluir roteadores, comutadores, estações base, ou outros equipamentos que facilitam a comunicação do dispositivo de origem 12 para o dispositivo de destino 14.

[0039] Em outro exemplo, o canal 16 pode incluir um meio de armazenamento que armazena dados de vídeo codificados gerados pelo dispositivo de origem 12. Neste exemplo, o dispositivo de destino 14 pode acessar o meio de armazenamento via acesso ao disco ou acesso ao cartão. O meio de armazenamento pode incluir uma variedade de meios de armazenamento de dados localmente acessados, tais como discos Blu-ray, DVDs, CD-ROMs, memória flash, ou outras mídias de armazenamento digital adequadas para o armazenamento de dados de vídeo codificados.

[0040] Em um outro exemplo, o canal 16 pode incluir um servidor de arquivos ou outro dispositivo de armazenamento intermediário que armazena dados de vídeo codificados gerados pelo dispositivo de origem 12. Neste exemplo, o dispositivo de destino 14 pode acessar dados de vídeo codificados armazenados no servidor de arquivos ou outro dispositivo de armazenamento intermediário via streaming ou download. O servidor de arquivos pode ser um tipo de servidor capaz de armazenar dados de vídeo codificados e transmitir os dados de vídeo codificados para o dispositivo de destino 14. Servidores de arquivos exemplares incluem servidores de web (por exemplo, para um site), servidores de protocolo de transferência de arquivos

(FTP), dispositivos de armazenamento ligados à rede (NAS) e unidades de disco locais.

[0041] O dispositivo de destino 14 pode acessar os dados de vídeo codificados por meio de uma conexão de dados padrão, tal como uma ligação à Internet. Exemplo de tipos de conexões de dados podem incluir canais sem fio (por exemplo, conexões Wi-Fi), conexões com fio (por exemplo, DSL, cabo modem, etc.), ou combinações de ambos que são adequadas para o acesso a dados de vídeo codificados armazenados em um servidor de arquivos. A transmissão de dados de vídeo codificados a partir do servidor de arquivo pode ser uma transmissão por streaming, uma transmissão de download, ou uma combinação de ambos.

[0042] As técnicas desta divulgação não se limitam a aplicações com ou sem fio. As técnicas podem ser aplicadas a codificação de vídeo em apoio a uma variedade de aplicações multimídia, tais como transmissões de televisão através do ar, transmissões de televisão por cabo, transmissões de televisão por satélite, transmissões de vídeo em streaming, por exemplo, através da Internet, codificação de dados de vídeo para armazenamento em um meio de armazenamento de dados, decodificação de dados de vídeo armazenados em um meio de armazenamento de dados, ou outras aplicações. Em alguns exemplos, o sistema de codificação de vídeo 10 pode ser configurado para suportar transmissão de vídeo de uma via ou de duas vias para suportar aplicativos como streaming de vídeo, reprodução de vídeo, broadcast de vídeo, e/ou vídeo telefonia.

[0043] A figura 1 é apenas um exemplo e as técnicas desta divulgação podem ser aplicadas às configurações de codificação de vídeo (por exemplo, codificação de vídeo ou decodificação de vídeo) que não necessariamente incluem qualquer comunicação de dados entre

os dispositivos de codificação e decodificação. Em outros exemplos, os dados são recuperados a partir de uma memória local, transmitidos através de uma rede, ou algo semelhante. Um dispositivo de codificação de vídeo pode codificar e armazenar dados na memória, e/ou um dispositivo de decodificação de vídeo pode recuperar e decodificar os dados da memória. Em muitos exemplos, a codificação e decodificação é executada por meio de dispositivos que não se comunicam um com o outro, mas simplesmente codificam os dados de memória e/ou recuperam e decodificam dados a partir da memória.

[0044] No exemplo da figura 1, o dispositivo de origem 12 inclui uma fonte de vídeo 18, um codificador de vídeo 20, e uma interface de saída 22. Em alguns exemplos, a interface de saída 22 pode incluir um modulador / demodulador (modem) e/ou um transmissor. A fonte de vídeo 18 pode incluir um dispositivo de captura de vídeo, por exemplo, uma câmara de vídeo, um arquivo de vídeo que contém dados de vídeo capturados previamente, uma interface de alimentação de vídeo para receber dados de vídeo a partir de um provedor de conteúdo de vídeo, e/ou um sistema de computação gráfica para gerar dados de vídeo, ou uma combinação de tais fontes de dados de vídeo.

[0045] O codificador de vídeo 20 pode codificar os dados de vídeo a partir de fonte de vídeo 18. Em alguns exemplos, o dispositivo de origem 12 transmite diretamente os dados de vídeo codificados para o dispositivo de destino 14 através de uma interface de saída 22. Em outros exemplos, os dados de vídeo codificados podem também ser armazenados em um meio de armazenamento ou um servidor de arquivos para acesso posterior pelo dispositivo de destino 14 para decodificação e/ou reprodução.

[0046] No exemplo da figura 1, o dispositivo de destino 14 inclui uma interface de entrada 28, um decodificador de vídeo 30, e um dispositivo de exibição 32. Em alguns exemplos, a interface de entrada 28 inclui um receptor e/ou um modem. A interface de entrada 28 pode receber dados de vídeo codificados pelo canal 16. O dispositivo de vídeo 32 pode ser integrado ao ou pode ser externo ao dispositivo de destino 14. Em geral, dispositivo de exibição 32 exibe dados de vídeo decodificados. O dispositivo de exibição 32 pode compreender uma variedade de dispositivos de vídeo, tais como um display de cristal líquido (LCD), um display de plasma, um display de diodo emissor de luz orgânica (OLED), ou outro tipo de dispositivo de exibição.

[0047] O codificador de vídeo 20 e decodificador de vídeo 30 cada um pode ser implementado como qualquer um de uma variedade de circuitos adequados, tais como um ou mais microprocessadores, processadores de sinal digitais (DSPs), circuitos integrados de aplicação específica (ASICs), arranjos de porta programáveis em campo (FPGAs), lógica discreta, hardware, ou quaisquer combinações dos mesmos. Se as técnicas são implementadas parcialmente em software, um dispositivo pode armazenar instruções no software em um meio de armazenamento legível por computador não transitório apropriado e pode executar as instruções em hardware utilizando um ou mais processadores para executar as técnicas desta divulgação. Qualquer um dos anteriores (incluindo o hardware, software, uma combinação de hardware e software, etc.) pode ser considerado como sendo um ou mais processadores. Cada um codificador de vídeo 20 e decodificador de vídeo 30 pode ser incluído em um ou mais codificadores ou decodificadores, cada um dos quais pode

ser integrado como parte de um codificador combinado / decodificador (CODEC) de um respectivo dispositivo.

[0048] Essa divulgação pode geralmente se referir a codificador de vídeo 20 "sinalizando" ou "transmitindo" determinadas informações para outro dispositivo, como decodificador de vídeo 30. O termo "sinalização" ou "transmissão" pode geralmente se referir à comunicação de elementos de sintaxe e/ou outros dados utilizados para decodificar os dados de vídeo comprimidos. Tal comunicação pode ocorrer em tempo quase real ou real. Alternativamente, esta comunicação pode ocorrer ao longo de um período de tempo, tal como pode ocorrer quando do armazenamento de elementos de sintaxe em um meio de armazenamento legível por computador, em um fluxo de bits codificado no momento da codificação, que pode então ser recuperado por um dispositivo de decodificação, em qualquer momento após ser armazenado neste meio.

[0049 Em alguns exemplos, codificador de vídeo 20 e decodificador de vídeo 30 operam de acordo com um padrão de compressão de vídeo, como o padrão HEVC mencionado acima, e descrito em HEVC Versão 1. Em adição ao padrão HEVC, há esforços em curso para produzir extensões de codificação de vídeo escalável, codificação de vídeo multivista e codificação 3D para HEVC. Além disso, os modos de codificação baseada em paleta, por exemplo, como descrito nesta divulgação, podem ser preditos em uma extensão do padrão HEVC. Em alguns exemplos, as técnicas descritas nesta divulgação para a codificação baseada em paleta podem ser aplicadas a codificadores e decodificadores configurados para operação de acordo com outras normas de codificação de vídeo, como o padrão ITU-TH.264/AVC ou padrões futuros. Por conseguinte, a aplicação de um modo de codificação baseada em paleta para codificar

unidades de codificação (CUs) ou unidades de predição (PUs) em um codec HEVC é descrita para fins de exemplo.

[0050] Em vários exemplos, as técnicas descritas nesta divulgação podem incluir técnicas para várias combinações de determinar, predizer e/ou sinalizar paletas em codificação baseada em paleta. Tal como descrito em maiores detalhes a seguir, as técnicas podem incluir qualquer combinação de determinar preditores de paleta, gerar um vetor de predição binário, e codificar / decodificar vetor de predição binário. Enquanto certos exemplos podem ser descritos individualmente para fins de ilustração e de clareza, esta divulgação contempla qualquer combinação das técnicas de codificação baseada em paleta aqui descritas.

[0051] Como será explicado em maior detalhe abaixo com referência às figuras 4-9, o codificador de vídeo 20 pode ser configurado para gerar uma paleta para um bloco atual dos dados de vídeo, gerar um vetor de predição binário para a paleta para o bloco atual de dados de vídeo, o vetor de predição binário compreendendo entradas indicando se ou não entradas de paleta previamente utilizadas (ou seja, as entradas em paletas de blocos previamente codificados) são reutilizadas para a paleta para o bloco atual de dados de vídeo, codificar o vetor de predição binário usando uma técnica de codificação run length e codificar o bloco atual de dados de vídeo usando a paleta. Da mesma forma, decodificador de vídeo 30 pode ser configurado para receber um vetor de predição binário codificado para um bloco atual dos dados de vídeo, decodificar o vetor de predição binário codificado usando uma técnica de decodificação run length, gerar uma paleta para o bloco atual de dados de vídeo baseado no vetor de predição binário, o vetor de predição binário compreendendo

entradas indicando se ou não entradas de paleta previamente utilizadas são reutilizadas para a paleta para o bloco atual de dados de vídeo e decodificar o bloco atual de dados de vídeo usando a paleta.

[0052] Em HEVC e outros padrões de codificação de vídeo, uma sequência de vídeo normalmente inclui uma série de imagens. As imagens também podem ser referidas como "quadros". Uma imagem pode incluir três matrizes de amostra, denotadas SL, SCb e SCr. SL é uma matriz bidimensional (ou seja, um bloco) de amostras de luma. SCb é uma matriz bidimensional de amostras de croma Cb. SCr é uma matriz bidimensional de amostras de croma Cr. Amostras de croma podem também ser aqui referidas como amostras de "crominância". Em outros casos, uma imagem pode ser monocromática e apenas pode incluir uma variedade de amostras de luma.

[0053] Para gerar uma representação codificada de uma imagem, codificador de vídeo 20 pode gerar um conjunto de unidades de árvore de codificação (CTUs). Cada uma das CTUs pode ser um bloco de árvore de codificação de amostras de luma, dois blocos de árvores de codificação correspondentes de amostras de croma, e estruturas de sintaxe utilizadas para codificar as amostras dos blocos de árvores de codificação. Um bloco de árvore de codificação pode ser um bloco NxN de amostras. A CTU também pode ser referida como um "bloco de árvore" ou uma "unidade de codificação maior" (LCU). A CTU de HEVC pode ser amplamente análoga aos macroblocos de outros padrões, tais como o H.264/AVC. No entanto, uma CTU não é necessariamente limitada a um tamanho particular e pode incluir uma ou mais unidades de codificação (CU). Uma fatia pode incluir um número inteiro de CTUs ordenadas consecutivamente na varredura raster.

[0054] Para gerar uma CTU codificada, o codificador de vídeo 20 pode de forma recursiva executar o particionamento quadtree nos blocos de árvores de codificação de uma CTU para dividir os blocos de árvores de codificação em blocos de codificação, daí o nome "unidades de árvores de codificação". Um bloco de codificação é um bloco NxN de amostras. A CU pode ser um bloco de codificação de amostras de luma e dois blocos de codificação de amostras de croma correspondentes de uma imagem que tem uma matriz de amostra de luma, uma matriz de amostra Cb e uma matriz de amostra Cr e estruturas de sintaxe utilizadas para codificar as amostras dos blocos de codificação. O codificador de vídeo 20 pode particionar um bloco de codificação de uma CU em um ou mais blocos preditivos. Um bloco de predição pode ser bloco de amostras retangular (por exemplo, quadrado ou não quadrado) no qual é aplicada a mesma predição. Uma unidade de predição (PU) de uma CU pode ser um bloco de predição de amostras de luma, dois blocos preditivos correspondentes de amostras de croma de uma imagem, e as estruturas de sintaxe utilizadas para predizer as amostras de blocos preditivos. O codificador de vídeo 20 pode gerar luma preditiva, blocos preditivos de Cb e Cr para luma, e blocos preditivos de Cb e Cr de cada PU da CU.

[0055] O codificador de vídeo 20 pode usar a intrapredição ou interpredição para gerar os blocos preditivos para a PU. Se codificador de vídeo 20 utiliza intrapredição para gerar os blocos preditivos de uma PU, codificador de vídeo 20 pode gerar os blocos preditivos da PU com base em amostras decodificadas da imagem associada à PU.

[0056] Se o codificador de vídeo 20 utiliza interpredição para gerar os blocos preditivos de uma PU,

codificador de vídeo 20 pode gerar os blocos preditivos de uma PU com base em amostras decodificados de uma ou mais imagens diferentes da imagem associada com a PU. O codificador de vídeo 20 pode usar unipredição ou bipredição para gerar os blocos preditivos de uma PU. Quando codificador de vídeo 20 usa unipredição para gerar os blocos preditivos para a PU, a PU pode ter um único vetor de movimento (MV). Quando codificador de vídeo 20 usa bipredição para gerar os blocos preditivos para a PU, a PU pode ter dois MVs.

[0057] Depois de codificador de vídeo 20 gerar um ou mais blocos preditivos (por exemplo, blocos de luma preditiva, Cb e Cr) para uma ou mais PUs de uma CU, codificador de vídeo 20 pode gerar um bloco residual para a CU. Cada uma das amostras no bloco residual pode indicar uma diferença entre uma amostra de um dos blocos preditivos da CU e uma amostra correspondente de um bloco de codificação da CU. Por exemplo, codificador de vídeo 20 pode gerar um bloco residual de luma para a CU. Cada amostra no bloco residual de luma da CU indica uma diferença entre uma amostra de luma em um dos blocos de luma preditiva da CU e uma amostra correspondente no bloco de codificação de luma original da CU. Além disso, codificador de vídeo 20 pode gerar um bloco residual de Cb para a CU. Cada amostra no bloco residual de Cb da CU pode indicar uma diferença entre uma amostra de Cb em um dos blocos Cb preditivo da CU e uma amostra correspondente no bloco de codificação Cb original da CU. O codificador de vídeo 20 também pode gerar um bloco residual de Cr para a CU. Cada amostra no bloco residual de Cr da CU pode indicar uma diferença entre uma amostra de Cr em um dos blocos de Cr preditivo da CU e uma amostra correspondente no bloco de codificação de Cr original da CU.

[0058] Além disso, codificador de vídeo 20 pode usar particionamento quadtree para decompor a blocos residuais de luma, Cb e Cr de uma CU em um ou mais blocos de transformada de luma, Cb e Cr. Um bloco de transformada pode ser um bloco de amostras retangular em que a mesma transformada é aplicada. Uma unidade de transformada (TU) de uma CU pode ser um bloco de transformada de amostras de luma, dois blocos de transformada de amostras de croma correspondentes, e estruturas de sintaxe utilizadas para transformar a amostras de blocos de transformada. Assim, cada TU de uma CU pode ser associada com um bloco de transformada de luminância, um bloco de transformada Cb e um bloco de transformada Cr. O bloco de transformada de luma associada à TU pode ser um sub-bloco do bloco residual de luma da CU. O bloco de transformada Cb pode ser um sub-bloco do bloco residual Cb da CU. O bloco de transformada Cr pode ser um sub-bloco do bloco residual Cr da CU.

[0059] O codificador de vídeo 20 pode aplicar uma ou mais transformadas para um bloco de transformada de uma TU para gerar um bloco de coeficientes para a TU. Um bloco de coeficientes pode ser uma matriz bidimensional de coeficientes de transformada. Um coeficiente de transformada pode ser uma quantidade escalar. Por exemplo, codificador de vídeo 20 pode aplicar uma ou mais transformadas para um bloco de transformada de luma de uma TU para gerar um bloco de coeficientes de luma para a TU. O codificador de vídeo 20 pode aplicar uma ou mais transformadas para um bloco de transformada de Cb de uma TU para gerar um bloco de coeficientes de Cb para a TU. O codificador de vídeo 20 pode aplicar uma ou mais transformadas para um bloco de transformada de Cr de uma TU para gerar um bloco de coeficientes de Cr para a TU.

[0060] Depois de gerar um bloco de coeficientes (por exemplo, um bloco de coeficientes de luminância, um bloco de coeficientes de Cb ou um bloco de coeficientes de Cr), codificador de vídeo 20 pode quantizar o bloco de coeficientes. Quantização geralmente refere-se a um processo em que os coeficientes de transformada são quantizados para possivelmente reduzir a quantidade de dados utilizada para representar os coeficientes de transformada, provendo compressão adicional. Depois de codificador de vídeo 20 quantiza um bloco de coeficientes, codificador de vídeo 20 pode codificar por entropia elementos de sintaxe indicando coeficientes de transformada quantizados. Por exemplo, o codificador de vídeo 20 pode executar o codificador de vídeo 20 pode realizar Codificação Aritmética Binária Adaptativa de Contexto (CABAC) nos elementos de sintaxe indicando coeficientes de transformada quantizados. O codificador de vídeo 20 pode emitir os elementos de sintaxe codificados por entropia em um fluxo de bits.

[0061] Para aplicar codificação CABAC a um elemento de sintaxe, o codificador de vídeo pode binarizar o elemento de sintaxe para formar uma série de um ou mais bits, que são referidos como "faixas”. Além disso, o codificador de vídeo pode identificar um contexto de codificação. O contexto de codificação pode identificar probabilidades de codificação de faixas com valores particulares. Por exemplo, um contexto de codificação pode indicar uma probabilidade de codificação de 0,7 de uma faixa de valor 0 e uma probabilidade de codificação de 0,3 uma faixa de valor 1. Após a identificação do contexto de codificação, o codificador de vídeo 20 pode dividir um intervalo em um subintervalo inferior e um subintervalo superior. Um dos subintervalos pode ser associado com o

valor 0 e o outro subintervalo pode ser associado com o valor 1. As larguras dos subintervalos podem ser proporcionais às probabilidades indicadas para os valores associados pelo contexto de codificação identificado. Se uma faixa do elemento de sintaxe tem o valor associado com o menor subintervalo, o valor codificado pode ser igual ao limite inferior do subintervalo inferior. Se a mesma faixa do elemento de sintaxe tem o valor associado com o subintervalo superior, o valor codificado pode ser igual ao limite inferior do subintervalo superior. Para codificar a próxima faixa do elemento de sintaxe, o codificador de vídeo pode repetir estas etapas com o intervalo sendo o subintervalo associado com o valor do bit codificado. Quando o codificador de vídeo 20 repete estas etapas para a próxima faixa, o codificador de vídeo pode usar probabilidades modificadas com base nas probabilidades indicadas pelo contexto de codificação identificado e os valores reais de faixas codificadas.

[0062] Quando um decodificador de vídeo 30 executa decodificação CABAC em um elemento de sintaxe, o decodificador de vídeo pode identificar um contexto de codificação. O decodificador de vídeo pode então dividir um intervalo em um subintervalo inferior e um subintervalo superior. Um dos subintervalos pode ser associado com o valor 0 e o outro subintervalo pode ser associado com o valor 1. As larguras dos subintervalos podem ser proporcionais às probabilidades indicadas para os valores associados pelo contexto de codificação identificado. Se o valor codificado está dentro do subintervalo inferior, o decodificador de vídeo pode decodificar uma faixa tendo o valor associado com o subintervalo inferior. Se o valor codificado está dentro do subintervalo superior, o decodificador de vídeo pode decodificar uma faixa tendo o

valor associado com o subintervalo superior. Para decodificar uma próxima faixa do elemento de sintaxe, o decodificador de vídeo pode repetir estas etapas com o intervalo sendo o subintervalo que contém o valor codificado. Quando o decodificador de vídeo repete estas etapas para a próxima faixa, o decodificador de vídeo pode usar probabilidades modificadas com base nas probabilidades indicadas pelo contexto de codificação identificado e as faixas decodificadas. O decodificador de vídeo pode então de-binarizar as faixas para recuperar o elemento de sintaxe.

[0063] Ao invés de realizar a codificação CABAC regular sobre todos os elementos de sintaxe, o codificador de vídeo pode codificar alguns elementos de sintaxe (por exemplo, faixas) usando codificação CABAC de bypassagem. Pode ser computacionalmente menos oneroso realizar codificação CABAC de bypassagem em uma faixa do que executar codificação CABAC regular sobre a faixa. Além disso, a realização de codificação CABAC de bypassagem pode permitir um maior grau de paralelização e capacidade de vazão. Faixas codificadas usando codificação CABAC de bypassagem podem ser referidas como "faixas de bypassagem". Agrupamento de faixas de bypassagem em conjunto pode aumentar a capacidade de vazão do codificador de vídeo e decodificador de vídeo. A máquina de codificação CABAC de bypassagem pode ser capaz de codificar várias faixas em um único ciclo, enquanto que a máquina de codificação CABAC regular pode ser capaz de codificar apenas uma única faixa em um ciclo. A máquina de codificação CABAC de bypassagem pode ser mais simples porque a máquina de codificação CABAC de bypassagem não seleciona contextos e pode assumir uma probabilidade de ½ para ambos os símbolos (0 e 1). Por

conseguinte, em codificação CABAC de bypassagem, os intervalos são divididos diretamente no meio.

[0064] O codificador de vídeo 20 pode emitir um fluxo de bits, que inclui os elementos de sintaxe codificados por entropia. O fluxo de bits pode incluir uma sequência de bits, que forma uma representação de imagens codificadas e dados associados. Em outras palavras, o fluxo de bits pode compreender uma representação codificada de dados de vídeo. O fluxo de bits pode compreender uma sequência de unidades de camada de abstração de rede (NAL). Cada uma das unidades NAL inclui um cabeçalho de unidade NAL e encapsula uma carga útil de sequência de byte brutos (RBSP). O cabeçalho de unidade NAL pode incluir um elemento de sintaxe que indica um código de tipo de unidade NAL. O código de tipo de unidade NAL especificado pelo cabeçalho de unidade NAL de uma unidade NAL indica o tipo da unidade NAL. Um PAAB pode ser uma estrutura de sintaxe contendo um número inteiro de bytes que está encapsulado dentro de uma unidade NAL. Em alguns casos, um PAAB inclui zero bit.

[0065] Diferentes tipos de unidades NAL podem encapsular diferentes tipos de RBSPs. Por exemplo, um primeiro tipo de unidade NAL pode encapsular uma RBSP para um conjunto de parâmetros de imagem (PPS), um segundo tipo de unidade NAL pode encapsular uma RBSP para uma fatia codificada, um terceiro tipo de unidade NAL pode encapsular uma RBSP para SEI, e assim por diante. Unidades NAL que encapsulam RBSPs para codificação de dados de vídeo (em oposição a RBSPs para conjuntos de parâmetros e mensagens SEI) podem ser referidas como unidades NAL de camada de codificação de vídeo (VCL).

[0066] O decodificador de vídeo 30 pode receber um fluxo de bits gerado pelo codificador de vídeo 20. Além disso, o decodificador de vídeo 30 pode analisar o fluxo de

bits para decodificar elementos de sintaxe a partir do fluxo de bits. Decodificador de vídeo 30 pode reconstruir as imagens dos dados de vídeo com base pelo menos em parte nos elementos de sintaxe decodificados a partir do fluxo de bits. O processo para reconstruir os dados de vídeo pode ser geralmente recíproco ao processo realizado por um codificador de vídeo 20. Por exemplo, um decodificador de vídeo 30 pode usar MVs de PUs para determinar os blocos preditivos para a PU de uma CU atual. Além disso, decodificador de vídeo 30 pode inversamente quantizar o bloco de coeficiente de transformada associado a TUs da CU atual. Decodificador de vídeo 30 pode realizar transformadas inversas nos bloco de coeficiente de transformada para reconstruir bloco de transformada associado às TUs da CU atual. O decodificador de vídeo 30 pode reconstruir os blocos de codificação da CU atual, adicionando as amostras dos blocos preditivos para a PU da CU atual para as amostras de blocos de transformada de todas as TUs da CU atual correspondente. Ao reconstruir os blocos de codificação para cada CU de uma imagem, decodificador de vídeo 30 pode reconstruir a imagem.

[0067] Como descrito acima, as técnicas aqui descritas podem ser aplicadas a codificação baseada em paleta de dados de vídeo. A codificação baseada em paleta pode geralmente incluir uma ou mais das seguintes técnicas. Com a codificação de vídeo baseada em paleta, um codificador de vídeo (por exemplo, um codificador de vídeo 20 e/ou decodificador de vídeo 30) pode formar uma assim chamada "paleta", tal como uma tabela de cores para representar os dados de vídeo da área particular (por exemplo, um dado bloco). Cada pixel pode ser associado com uma entrada na paleta que representa a cor do pixel. Por exemplo, codificador de vídeo 20 e/ou decodificador de

vídeo 30 pode codificar um índice que relaciona o valor de pixel ao valor apropriado na paleta.

[0068] No que diz respeito à estrutura HEVC, como um exemplo, as técnicas de codificação baseada em paleta podem ser configuradas para serem utilizadas como um modo de unidade de codificação (CU). Em outros exemplos, as técnicas de codificação baseada em paleta podem ser configuradas para serem utilizadas como um modo de PU no âmbito de HEVC. Por conseguinte, todos os seguintes processos divulgados descritos no contexto de um modo de CU pode, adicionalmente ou alternativamente, aplicam-se as PUs. No entanto, estes exemplos baseados em HEVC não devem ser considerados uma restrição ou limitação das técnicas de codificação baseada em paleta aqui descritas, uma vez que tais técnicas podem ser aplicadas a trabalhar de forma independente ou como parte de outros sistemas / normas existentes ou ainda a serem desenvolvidos. Nestes casos, o aparelho para codificação de paleta podem ser blocos quadrados, blocos retangulares ou mesmo formas não retangulares.

[0069] O documento X. Guo e A. Saxena, "RCE4: Summary Report of HEVC Range Extension Core Experiments 4 (RCE4) on pallete coding for screen content", JCTVC-P0035, San Jose, EUA, 09-17 de janeiro de 2014 descreve resultados de dois testes de modos baseados em paleta, que foram notificados para alcançar a redução da taxa de distorção significativa de Bjontegaard (taxa BD), especialmente para o conteúdo de tela. Os dois métodos são brevemente resumidos abaixo.

[0070] Em um exemplo do método, tal como descrito, por exemplo, no documento X. Guo. Y. Lu, e S. Li, "RCE4: Teste 1. Major-color-based screen content coding", JCTVC-P0108, San Jose, EUA, 09-17 de janeiro de 2014, um

algoritmo baseado em histograma é utilizado para classificar os pixels. Em particular, os valores mais significativos dos picos N em um histograma são selecionados como cores principais para a codificação. Os valores de pixels que estão perto de uma cor principal serão quantizados para a cor principal. Outros pixels que não pertencem a qualquer um dos conjuntos de cores principais são pixels de escape, que também são quantizados antes da codificação. Para a codificação sem perdas, nenhuma quantização é utilizada.

[0071] Através da utilização de classificação, os pixels de uma unidade de codificação (CU) podem ser convertidos em índices de cor. Depois disso, o número e valores de cor são codificados. Em seguida, os índices de cor são codificados como se segue:

• Para cada linha de pixel, um flag é sinalizado para indicar o modo de codificação. Há três modos: modo horizontal, modo vertical e modo normal.

- Se o modo é o modo horizontal, a linha inteira compartilha o mesmo índice de cor. Neste caso, o índice de cor é transmitido.
- Se o modo é modo vertical, a linha inteira é a mesma com a linha acima. Neste caso, nada é transmitido. A linha atual copia os índices de cor da linha acima.
- Se o modo é o modo normal, um flag é sinalizado para cada posição de pixel para indicar se ele é o mesmo com um dos pixels esquerdos e acima. Se não, o próprio índice é transmitido.

Além disso, se o pixel é pixel de escape, o valor de pixel é transmitido.

[0072] Em um outro método exemplar, tal como descrito, por exemplo, no documento L. Guo, W. Pu, M. Karczewicz, J. Sole, R. Joshi, e F. Zou, "RCE4: Results of Test 2 on Pallete Mode for Screen Content Coding", JCTVC-P0198, San Jose, EUA, 9-17 de janeiro de 2014, um modo de codificação baseada em paleta é incluído como um modo de CU. O processo de codificação do segundo método pode incluir o seguinte:

• Transmissão da paleta: um esquema de predição entrada a entrada é utilizado para codificar a paleta atual com base na paleta da CU esquerda (a CU vizinha a CU atualmente sendo codificada para a esquerda). Depois disso, as entradas não preditas da paleta são transmitidas.

• Transmissão de valores de pixel: os pixels na CU são codificados em uma ordem de varredura raster utilizando os seguintes três modos:

- o "Modo de execução": Um índice de paleta é primeiro sinalizado, seguido de "palle_run" (M). Os seguintes índices de paleta M são os mesmos que o índice de paleta sinalizado primeiro sinalizado,
- o "Modo de cópia de cima": Um valor "copy_run" (N) é transmitido para indicar que, para os seguintes N índices de paleta são os mesmos que os seus vizinhos acima, respectivamente,
- o "Modo de pixel": Um flag de predição é primeiro transmitido. O valor do flag sendo igual a 1 indica que predição residual usando pixel vizinho superior reconstruído como um preditor é transmitida. Se o valor deste flag é 0, o valor do pixel é transmitido sem predição.

[0073] A paleta pode tornar-se uma parte relativamente importante dos bits para um bloco codificado

de paleta (por exemplo, CU). Deste modo, o codificador de vídeo pode predizer uma ou mais entradas de paleta com base em uma ou mais entradas de uma paleta previamente codificada (por exemplo, como referido acima em relação à "transmissão da paleta").

[0074] Em alguns exemplos, o codificador de vídeo pode gerar uma lista de preditor de paleta ao predizer entradas de paleta. Por exemplo, o documento C. Gisquet, G. Laroche e P. Onno, "AhG10: Pallete predicos stuffing", JCTVC-Q0063 divulga um processo exemplar para determinar preditores de paleta. Em alguns exemplos, o codificador de vídeo pode usar um vetor Booleano para indicar se cada item em uma lista de preditor de paleta é utilizado (ou não utilizado) para a predição de uma ou mais entradas na paleta para o bloco atualmente sendo codificado.

[0075] Em alguns exemplos, todos os itens da lista de preditor de paleta são derivados a partir da paleta previamente codificada (por exemplo, a paleta codificada com o bloco previamente codificado). No entanto, tais paletas podem ser espacialmente distantes da CU atual, o que pode fazer a correlação de paleta relativamente fraca. Em geral, a expansão da tabela de preditor de paleta pode ser útil (por exemplo, pode prover indicadores mais precisos, que podem resultar em um ganho de eficiência). No entanto, determinar e usar uma tabela de preditor de paleta um relativamente grande resulta em um vetor booleano relativamente mais longo.

[0076] Em um exemplo de codificação de paleta, codificador de vídeo 20 pode gerar um elemento de sintaxe, tal como um flag "PLT_Mode_flag", que indica se ou não uma paleta com base no modo de codificação é utilizada para uma região particular de um quadro de vídeo. Por exemplo, o PLT_Mode_flag pode ser gerado no nível da fatia, no nível

de CU, no nível de PU, ou qualquer outro nível de um quadro de vídeo. Por exemplo, o codificador de vídeo 20 pode gerar o PLT_Mode_flag ao nível de CU e sinalizar o PLT_Mode_flag em um fluxo de bits de vídeo codificado. Decodificador de vídeo 30 pode, em seguida, analisar o PLT_Mode_flag ao decodificar o fluxo de bits de vídeo codificado. Neste exemplo, um valor deste PLT_MOde_flag igual a 1 especifica que a CU atual é codificada utilizando um modo de paleta. Neste caso, decodificador de vídeo 30 pode aplicar o modo de codificação baseada em paleta para decodificar a CU. Em alguns exemplos, um elemento de sintaxe pode indicar um de uma pluralidade de diferentes modos de paleta para a CU.

[0077] Um valor deste PLT_Mode_flag igual a 0 especifica que a CU atual é codificada usando um modo diferente do modo de paleta. Por exemplo, pode ser utilizado qualquer um de uma variedade de modos de codificação interpreditiva, intrapreditiva, ou outras. Quando um valor de PLT_Mode_flag é 0, informação adicional pode ser transmitida para sinalizar que o modo específico é utilizado para a codificação da respectiva CU, em que tal modo específico, tipicamente, pode ser um modo de codificação HEVC. A utilização do PLT_Mode_flag é descrita para fins de exemplo. Em outros exemplos, no entanto, outros elementos de sintaxe, como códigos multbits podem ser utilizados para indicar se o modo de codificação baseada em paleta deve ser utilizado para uma CU (PU ou em outros exemplos) ou para indicar qual de uma pluralidade de modos devem ser utilizados.

[0078] O PLT_Mode_flag ou outro elemento de sintaxe também pode ser transmitido a um nível superior. Por exemplo, o PLT_Mode_flag pode ser transmitido ao nível de fatia. Neste caso, um valor de flag igual a 1 indica que todas as CUs na fatia serão codificadas utilizando o modo

de paleta (o que significa que não há informação de modo, por exemplo, para o modo de paleta ou outros modos, necessita de ser transmitida ao nível de CU). Da mesma forma, este flag pode ser sinalizado no nível de conjunto de parâmetros de imagem (PPS), conjunto de parâmetros de sequência (SPS) ou conjunto de parâmetros de vídeo (VPS). Além disso, um flag pode ser enviado em um destes níveis especificando se o modo de paleta é habilitado ou desabilitado para a imagem particular, fatia, etc., enquanto o PLT_Mode_flag indica se o modo de codificação baseada em paleta é utilizado para cada CU. Neste caso, se um flag ou outro elemento de sintaxe enviado no nível de fatia, PPS, SPS ou VPS indica que o modo de codificação de paleta é desabilitado, em alguns exemplos, pode não haver necessidade para sinalizar o PLT_Mode_flag para cada CU. Alternativamente, se um flag ou outro elemento de sintaxe enviado na fatia, nível PPS, SPS ou VPS indica que o modo de codificação de paleta é habilitado, o PLT_Mode_flag pode ser ainda sinalizado para indicar se o modo de codificação baseada em paleta deve ser utilizado para cada CU. Novamente, como acima referido, a aplicação destas técnicas para a indicação de codificação baseada em paleta de uma CU poderia adicionalmente ou em alternativa, ser utilizada para indicar codificação baseada em paleta de uma PU.

[0079] Um flag, tal como PLT_Mode_flag, também pode ou, alternativamente, ser condicionalmente transmitido ou inferido. As condições para a transmissão do PLT_Mode_flag ou inferir o flag pode ser um ou mais de, como exemplos, o tamanho da CU, o tipo de quadro, o espaço de cor, o componente de cor, o tamanho do quadro, a taxa de quadros, o id de camada em codificação de vídeo escalável ou o id de vista em codificação multivista.

[0080] As técnicas para a geração e transmissão de uma paleta serão agora discutidas. O codificador de vídeo 20 pode ser configurado para gerar e sinalizar um ou mais elementos de sintaxe e valores que podem ser utilizados pelo decodificador de vídeo 30 para a construção e/ou reconstrução da paleta utilizada pelo codificador de vídeo 20 para codificar um determinado nível do quadro de vídeo (por exemplo, uma CU). Em alguns exemplos, codificador de vídeo 20 pode indicar ou não sinalizar uma paleta para cada CU. Em outros exemplos, codificador de vídeo 20 pode indicar ou não sinalizar uma paleta que pode ser compartilhada entre várias CUs.

[0081] O tamanho da paleta, por exemplo, em termos do número de valores de pixels incluídos, pode ser um valor fixo ou pode ser sinalizado por um codificador de vídeo 20 em um fluxo de bits de vídeo codificado. Decodificador de vídeo 30 pode receber e decodificar a indicação do tamanho da paleta do fluxo de bits de vídeo codificado. A sinalização pode ser separada por componentes diferentes ou um único tamanho pode ser sinalizado para todos os componentes. Os diferentes componentes podem ser, por exemplo, componentes de luminância e crominância. A sinalização pode usar códigos unários ou códigos unários truncados (por exemplo, que trunca a um limite máximo do tamanho da paleta). Códigos Exponencial-Golomb ou Rice-Golomb também podem ser utilizados. Em alguns exemplos, a sinalização do tamanho pode ser feita da seguinte forma: depois de sinalização uma entrada na paleta, um flag de "parada" é sinalizado. Um valor deste flag igual a 1 especifica que a entrada atual é a última na paleta; um valor deste flag igual a 0 especifica que não há mais entradas na paleta. O flag de "parada" não pode ser transmitido pelo codificador se a paleta já construída

atinge o limite máximo do tamanho da paleta. Em alguns exemplos, o tamanho da paleta pode também ser transmitido condicionalmente ou inferido com base na informação lateral da mesma forma como descrito acima para "Transmissão de flag PLT_Mode_flag".

[0082] A paleta pode ser transmitida separadamente para cada componente de cor na CU. Por exemplo, pode haver uma paleta para o componente Y desta CU, outra paleta para o componente U da CU, e ainda outra paleta para o componente V desta CU. Para a paleta Y, a entrada pode (assumidamente) ser um valor Y representante nesta CU. O mesmo se aplica aos componentes U e V. É também possível que a paleta possa ser transmitida para todos os componentes de cor na CU. Neste exemplo, a i-ésima entrada na paleta é um triplo (Yi, Ui, Vi). Neste caso, a paleta inclui valores para cada um dos componentes.

[0083] Predição de uma paleta é uma abordagem alternativa para a "transmissão de paleta" descrita acima. Em alguns exemplos, as técnicas de predição de paleta podem ser utilizadas em conjunto com técnicas de sinalização de paleta. Isto é, o codificador de vídeo 20 pode ser configurado para sinalizar elementos de sintaxe que podem ser utilizados pelo decodificador de vídeo 30 para predizer uma porção do número total de entradas de paleta. Além disso, o codificador de vídeo 20 pode ser configurado para sinalizar explicitamente outra porção das entradas de paleta.

[0084] Em um exemplo de uma abordagem de predição de paleta, para cada CU, um flag "Pred_palette_flag" é transmitido. Um valor deste flag igual a 1 especifica que a paleta para a CU atual irá ser predita a partir de dados anteriores e, portanto, não há necessidade para a paleta ser transmitida. Um valor deste flag igual a 0 significa

que a paleta de CU atual deve ser transmitida. O flag pode ser separado para componentes de cor diferentes (por exemplo, de modo que os flags 3 precisam de ser transmitidos para uma CU de vídeo YUV), ou um único flag pode ser sinalizado para todos os componentes de cor. Por exemplo, um único flag pode indicar se as paletas são transmitidas para todos os componentes, ou se as paletas para todos os componentes irão ser preditas.

[0085] Em alguns exemplos, a predição pode ser realizada da seguinte maneira. Se o valor de flag de predição é igual a 1, para a CU atual, codificador de vídeo 20 copia a paleta de uma ou mais das CUs vizinhas já codificadas. A paleta das CUs vizinhas já codificadas pode ter sido transmitida ou predita. Por exemplo, a CU vizinha copiada pode ser a CU vizinha esquerda. No caso em que a paleta da CU esquerda não está disponível (como no caso em que a CU esquerda não é codificada utilizando o modo de paleta ou a CU atual está na primeira coluna da figura), a cópia da paleta pode ser da CU acima da CU atual. A paleta copiada também pode ser uma combinação das paletas de um número de CUs vizinhas. Por exemplo, uma ou mais fórmulas, funções, regras ou semelhantes podem ser aplicadas para gerar uma paleta com base em paletas de um ou de uma combinação de uma pluralidade de CUs vizinhas.

[0086] Também é possível que uma lista de candidatos possa ser construída e um índice ser transmitido por um codificador de vídeo 20 para indicar a CU candidata a partir da qual a CU atual copia a paleta. Decodificador de vídeo 30 pode construir a mesma lista de candidatos e, em seguida, usar o índice para selecionar a paleta da CU correspondente para uso com a CU atual. Por exemplo, a lista de candidatos pode incluir uma CU acima da CU e uma no lado esquerdo, em relação à CU atual sendo codificada

dentro de uma fatia ou imagem. Neste exemplo, um flag ou outro elemento de sintaxe pode ser sinalizado para indicar a seleção de candidatos. Por exemplo, um flag transmitido igual a 0 significa que a cópia é da CU esquerda, e um flag transmitido igual a 1 significa que a cópia é da CU superior. Decodificador de vídeo 30 seleciona a paleta para ser copiada da CU vizinha correspondente e copia-a para utilização na decodificação da CU atual. A predição também pode ser derivada utilizando os valores das amostras mais frequentes nos vizinhos causais da CU atual.

[0087] A predição de paletas também pode ser entrada a entrada. Para cada entrada na paleta, codificador de vídeo 20 gera e sinaliza um flag. Um valor de um flag igual a 1 para uma determinada entrada especifica que um valor predito (por exemplo, a entrada correspondente a partir de uma CU candidata selecionada como a CU esquerda) é utilizado como o valor de entrada. Um valor de um flag igual a 0 especifica que esta entrada não está predita e seu valor será transmitido para o decodificador de vídeo 30 a partir do codificador de vídeo 20, por exemplo, sinalizado em um fluxo de bits codificado pelo codificador de vídeo 20 para posterior decodificação pelo decodificador de vídeo 30.

[0088] O valor de "pred_palette_flag", a CU candidata cuja paleta é utilizada para predizer a paleta da CU atual ou as regras para construir os candidatos podem ser também condicionalmente transmitidos ou inferidos com base na informação lateral da mesma forma como acima descrito para "Transmissão de flag PLT_Mode_flag".

[0089] Em seguida, o codificador de vídeo 20 pode gerar e sinalizar um mapa que indica que a respectiva entrada de paleta está associada com cada pixel em uma CU. A i-ésima entrada no mapa é correspondente à i-ésima

posição na CU. Um valor da i-ésima entrada igual a 1 especifica que o valor do pixel nesta i-ésima posição na CU é um dos valores da paleta, e um índice de paleta é em seguida transmitido para que decodificador de vídeo 30 possa reconstruir o valor de pixel (no caso de haver apenas uma entrada na paleta, a transmissão de índice de paleta pode ser ignorada). Um valor da i-ésima entrada igual a 0 especifica que o valor de pixel na i-ésima posição na CU não é na paleta e, assim, o valor do pixel será transmitido ao decodificador de vídeo 30 explicitamente.

[0090] Se o valor de pixel em uma posição na CU é um valor na paleta, observa-se que existe uma alta probabilidade de que as posições vizinhas da CU tenham o mesmo valor de pixel. Então, após codificar um índice de paleta (diga-se j, que é correspondente ao valor de pixel s) para uma posição, codificador de vídeo 20 pode transmitir um elemento de sintaxe "run" para indicar o número de valores consecutivos do mesmo valor de pixel s na CU antes da varredura alcançar um valor de pixel diferente. Por exemplo, se o próximo imediato tem um valor diferente de s, em seguida, execução = 0 é transmitido. Se o próximo é s, mas um após não é s, em seguida, execução = 1.

[0091] No caso de uma execução não ser transmitida (por exemplo, *Derivação de Execução Implícita*), o valor da execução pode ser uma constante, por exemplo, 4, 8, 16, etc., ou o valor da execução pode igualmente ser dependente de informações laterais. Por exemplo, o valor da execução pode depender do tamanho do bloco, por exemplo, a execução é igual à largura do bloco atual, ou a altura do bloco atual, ou metade da largura (ou metade da altura) do bloco atual, ou uma fração da largura e da altura do bloco, ou um múltiplo da relação altura / largura do bloco. O valor da execução pode também ser dependente do QP, tipo de

quadro, componente de cor, formato de cor (por exemplo, 444, 422, 420) e/ou espaço de cores (por exemplo, YUV, RGB). O valor da execução também pode depender da direção da varredura. Em outros exemplos, o valor da execução pode depender de outros tipos de informação lateral. O valor da execução também pode ser sinalizado usando a sintaxe de alto nível (por exemplo, PPS, SPS).

[0092] Em alguns exemplos, o mapa pode não precisar de ser transmitido. A execução só pode começar em determinados locais. Por exemplo, a execução só pode começar no início de cada linha, ou o início de cada N linhas. A localização de partida pode ser diferente para diferentes direções de varredura. Por exemplo, se a varredura vertical é utilizada, a execução só pode começar no início de uma coluna ou no início de cada N colunas. A localização de partida pode depender de informações laterais. Por exemplo, a localização de partida pode ser o ponto médio de cada linha, ou de cada coluna, ou 1/n, 2/n, ... (n1)/n (isto é, frações) de cada linha / coluna. A localização de partida também pode depender do QP, do tipo de quadro, componente de cor, formato de cor (por exemplo, 444, 422, 420) e/ou espaço de cor (por exemplo, YUV, RGB). Em outros exemplos, a posição de início da execução pode depender de outros tipos de informações laterais. A posição de partida também pode ser sinalizada com a sintaxe de alto nível (por exemplo, PPS, SPS, etc.).

[0093] É também possível que a derivação de posição de partida implícita e a derivação de execução implícita devem ser combinadas. Por exemplo, a execução é igual à distância entre duas posições de início vizinhas. No caso em que o ponto de partida é o começo (ou seja, a primeira posição) de cada linha, o comprimento da execução é uma linha.

[0094] A direção de varredura pode ser vertical ou horizontal. É possível que um flag seja transmitido para cada CU para indicar a direção da varredura. Flags podem ser transmitidos separadamente para cada componente ou um único flag pode ser transmitido e a direção de varredura indicada aplica-se a todos os componentes de cor. É também possível que outras direções de varredura, como 45 graus ou 135 graus, sejam utilizadas. A ordem de verificação pode ser fixa ou pode ser dependente de informação lateral da mesma forma como descrito acima para "Transmissão de PLT_Mode_flag".

[0095] Acima, é explicado como transmitir uma paleta. Uma alternativa para os exemplos acima descritos é a de construir a paleta on-the-fly. Neste caso, no início da CU, não existe qualquer entrada na paleta, e conforme o codificador de vídeo 20 sinaliza novos valores dos pixels para as posições na CU, estes valores estão incluídos na paleta. Ou seja, codificador de vídeo 20 acrescenta valores de pixel na paleta conforme eles são gerados e transmitidos para as posições na CU. Em seguida, as posições posteriores na CU que têm os mesmos valores podem referir-se ao valores de pixel na paleta, por exemplo, com valores de índice, em vez de ter um codificador de vídeo 20 de transmitindo os valores de pixel. Da mesma forma, quando decodificador de vídeo 30 recebe um novo valor de pixel (por exemplo, sinalizado pelo codificador) para uma posição na CU, que inclui o valor de pixel na paleta construída pelo decodificador de vídeo 30. Quando posições posteriores na CU têm valores de pixel que foram adicionados à paleta, decodificador de vídeo 30 pode receber informação, tais como, por exemplo, valores do índice, que identificam os valores de pixel correspondentes na paleta para a reconstrução dos valores de pixel na CU.

[0096] Se o tamanho máximo de paleta é alcançado, por exemplo, como a paleta é construída de forma dinâmica on-the-fly, em seguida, o codificador e decodificador compartilham do mesmo mecanismo para remover uma entrada da paleta. Um método consiste em remover a entrada mais antiga na paleta (fila FIFO). Outro método consiste em remover a entrada menos utilizada na paleta. Outra é ponderar ambos os métodos (tempo em paleta e frequência de utilização) para decidir a entrada a ser substituída. Como um exemplo, se uma entrada de valor de pixel é removida da paleta, e o valor de pixel ocorre novamente a uma posição posterior na paleta, o codificador pode transmitir o valor do pixel em vez de incluir uma entrada na paleta. Adicionalmente, ou alternativamente, é possível que um tal valor de pixel possa ser reintroduzido na paleta, depois de ter sido removido, por exemplo, conforme o codificador e decodificador digitaliza as posições na CU.

[0097] Esta divulgação também considera combinar uma paleta inicial sinalizando com a derivação on-the-fly da paleta. Em um exemplo, a paleta inicial seria atualizada com a codificação dos pixels. Por exemplo, ao transmitir a paleta inicial, codificador de vídeo 20 pode adicionar valores na paleta inicial ou valores de mudança na paleta inicial como valores de pixel de localizações adicionais na CU são verificados. Da mesma forma, ao receber uma paleta inicial, decodificador de vídeo 30 pode adicionar valores aos valores da paleta inicial ou valores de mudança na paleta inicial conforme valores de pixel de locais adicionais na CU são verificados. Do mesmo modo, o codificador pode sinalizar se a CU atual utiliza a transmissão de toda a paleta, ou geração de paleta on-the-fly, ou uma combinação de transmissão de uma paleta inicial com a atualização da paleta inicial por derivação on-the-

fly. Em alguns exemplos, a paleta inicial pode ser uma paleta completa no tamanho máximo da paleta, neste caso os valores na paleta inicial podem ser alterados, ou uma paleta de tamanho reduzido, neste caso os valores são adicionados à paleta inicial e, valores opcionalmente sob a paleta inicial são alterados.

[0098] Acima, foi descrito como transmitir o mapa, identificando o valor de pixel. Juntamente com aquele método descrito acima, a transmissão do mapa pode ser realizada através da indicação de cópia de linha. Em um exemplo, a cópia de linha é sinalizada por um codificador de vídeo 20, de modo que o valor de pixel para uma entrada é igual ao valor do pixel da entrada de uma linha acima (ou na coluna do lado esquerdo, se a verificação for vertical). Em seguida, a 'execução' de entradas que são copiadas a partir da linha pode ser sinalizada. Além disso, a linha a partir da qual ela é copiada pode ser indicada; várias linhas acima podem ser armazenadas para esta finalidade. Por exemplo, as quatro linhas anteriores são armazenadas e qual linha é copiada pode ser sinalizado com um código truncado unário ou outros códigos, e, em seguida, quantas entradas desta linha são copiadas, ou seja, a execução, pode ser sinalizada. Assim, em alguns exemplos, o valor de pixel de uma entrada pode ser um sinal para ser igual a um valor de pixel de uma entrada em uma linha imediatamente acima ou duas ou mais linhas acima da linha atual.

[0099] No caso em que nenhuma execução é sinalizada, o valor da execução pode ser constante / fixo ou pode ser dependente de informação lateral (e derivado pelo decodificador) usando o método descrito acima.

[0100] É também possível que o mapa não necessite de ser transmitido. Por exemplo, a execução pode começar apenas em determinadas posições. A posição inicial pode ser

fixa ou pode ser dependente de informação lateral (e derivada pelo decodificador), então a sinalização da posição de partida pode ser ignorada. Em vez disso, uma ou mais das técnicas descritas acima, podem ser aplicadas. A derivação de posição inicial implícita e a derivação de execução implícita podem também ser combinadas, utilizando o mesmo método tal como descrito acima.

[0101] Se são utilizados ambos os métodos de transmissão de mapa, então, um flag pode indicar se o elemento de imagem é obtido a partir da paleta ou a partir das linhas anteriores, e, em seguida, um índice indica a entrada na paleta ou a linha, e, finalmente, a "Execução'.

[0102] A figura 2 é um diagrama de blocos ilustrando um codificador de vídeo exemplar 20 que pode aplicar técnicas de codificação de paleta desta divulgação. A figura 2 é provida para fins de explicação e não deve ser considerada como limitativa das técnicas como amplamente exemplificadas e descritas nesta divulgação. Para efeitos da explicação, esta divulgação descreve um codificador de vídeo 20, no contexto da codificação HEVC. No entanto, as técnicas desta divulgação podem ser aplicáveis a outras normas ou métodos de codificação.

[0103] O codificador de vídeo 20 representa um exemplo de um dispositivo que pode ser configurado para executar as técnicas de codificação de vídeo baseada em paleta de acordo com vários exemplos descritos nesta divulgação. Por exemplo, o codificador de vídeo 20 pode ser configurado para seletivamente codificar vários blocos de dados de vídeo, tais como CUs ou PUs em codificação HEVC, utilizando codificação baseada em codificação em paleta ou não baseada em paleta.

[0104] No exemplo da figura 2, um codificador de vídeo 20, inclui a unidade de processamento de predição

100, a unidade de geração residual 102, unidade de processamento de transformada 104, unidade de quantização 106, unidade de quantização inversa 108, unidade de processamento de transformada inversa 110, unidade de reconstrução 112, unidade de filtro 114, armazenador de imagem decodificada (DPB) 116, memória de vídeo 119 e unidade de codificação de entropia 118. A unidade de processamento de predição 100 inclui unidade de processamento de interpredição 120 e unidade de processamento de intrapredição 126. A unidade de processamento de interpredição 120 inclui uma unidade de estimação de movimento e uma unidade de compensação de movimento (não mostrada). O codificador de vídeo 20 inclui também unidade de codificação baseada em paleta 122 configurada para realizar vários aspectos das técnicas de codificação baseada em paleta descritas nesta divulgação. Em outros exemplos, o codificador de vídeo 20 pode incluir mais, menos ou diferentes componentes funcionais.

[0105] Como será explicado em maior detalhe abaixo com referência às figuras 4-9, unidade de codificação baseada em paleta 122 do codificador de vídeo 20 pode ser configurada para gerar uma paleta para um bloco atual dos dados de vídeo, gerar um vetor de predição binário para a paleta para o bloco atual de dados de vídeo, o vetor de predição binário compreendendo entradas que indicam se ou não entradas de paleta previamente utilizadas são reutilizadas para a paleta para o bloco atual de dados de vídeo, codificar o vetor de predição binário usando uma técnica de codificação run length e codificar o bloco atual de dados de vídeo usando a paleta.

[0106] O codificador de vídeo 20 pode receber dados de vídeo. O codificador de vídeo 20 pode codificar cada CTU em uma fatia de uma imagem dos dados de vídeo.

Cada uma das CTUs pode estar associada com blocos de árvores de codificação de luma de tamanho igual (CTBs) e CTBs correspondentes da imagem. Como parte de codificar uma CTU, unidade de processamento de predição 100 pode executar o particionamento quad-tree para dividir os CTBs da CTU em blocos progressivamente menores. O bloco menor pode ser blocos de codificação de CUs. Por exemplo, a unidade de processamento de predição 100 pode dividir um CTB associado com uma CTU em quatro sub-blocos de tamanho igual, particionar um ou mais dos sub-blocos em quatro sub-sub-blocos de tamanho igual, e assim por diante.

[0107] Como mostrado na figura 2, a memória de vídeo 119 recebe dados de vídeo que são utilizados para a codificação de um bloco de vídeo atual dentro de um quadro de vídeo. A memória de vídeo 119 pode armazenar dados de vídeo a serem codificados pelos componentes de um codificador de vídeo 20 (por exemplo, configurado para armazenar dados de vídeo). Os dados de vídeo armazenados na memória de vídeo 119 podem ser obtidos, por exemplo, a partir de fonte de vídeo 18 da figura 1. DPB 116 é um exemplo de DPB que armazena dados de vídeo de referência para utilização na codificação de dados de vídeo por um codificador de vídeo 20 (por exemplo, em modos de inter ou intracodificação, também referidos como modos de codificação intra ou interpreditivos). A memória de vídeo 119 e DPB 116 pode ser formada por qualquer um de uma variedade de dispositivos de memória, tais como a memória dinâmica de acesso aleatório (DRAM), DRAM síncrona incluindo (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM), ou outros tipos de dispositivos de memória. A memória de vídeo 119 e DPB 116 pode ser provida pelo mesmo dispositivo de memória ou dispositivos de memória separados. Em vários exemplos, memória de vídeo 119

pode ser sobre chip com outros componentes de um codificador de vídeo 20, ou fora do chip em relação a esses componentes.

[0108] O codificador de vídeo 20 pode codificar CUs de uma CTU para gerar representações codificadas das CUs (ou seja, CUs codificadas). Como parte de codificar uma de CU, unidade de processamento de predição 100 pode dividir os blocos de codificação associados com a CU entre uma ou mais PUs da CU. Assim, cada PU pode ser associada com um bloco de predição de luma e blocos de predição de croma correspondentes. O codificador de vídeo 20 e decodificador de vídeo 30 podem suportar PUs tendo vários tamanhos. Tal como indicado acima, o tamanho de uma CU pode referir-se ao tamanho do bloco de codificação de luma da CU e o tamanho de uma PU pode referir-se ao tamanho de um bloco de predição de luma da PU. Supondo-se que o tamanho de uma CU particular seja 2Nx2N, codificador de vídeo 20 e decodificador de vídeo 30 pode suportar tamanhos de PU de 2Nx2N ou NxN para a intrapredição e tamanhos de PU simétricas de 2Nx2N, 2NxN, Nx2N, NxN, ou similar para a interpredição. O codificador de vídeo 20 e decodificador de vídeo 30 pode também suportar particionamento assimétrico para tamanhos de PU de 2NxnU, 2NxnD, nLx2N e nRx2N para, interpredição.

[0109] A unidade de processamento de interpredição 120 pode gerar dados preditivos para uma PU pela realização de interpredição em cada PU de uma CU. Os dados preditivos para a PU podem incluir blocos preditivos da PU e informação de movimento para a PU. A unidade de interpredição 121 pode realizar diferentes operações para uma PU de uma CU dependendo de se a PU é uma fatia I, uma fatia P ou uma fatia B. Em uma fatia I, todas as PUs são intrapreditas. Por isso, se a PU está em uma fatia, a

unidade de interpredição 121 não realiza interpredição na PU. Assim, para os blocos codificados em modo I, o bloco predito é formado usando a predição espacial dos blocos vizinhos previamente codificados dentro do mesmo quadro.

[0110] Se uma PU está em uma fatia P, a unidade de estimação de movimento de unidade de processamento de interpredição 120 pode buscar as imagens de referência em uma lista de imagens de referência (por exemplo, "RefPicList0") para uma região de referência para a PU. A região de referência para a PU pode ser uma região, dentro de uma imagem de referência, que contém blocos de amostra que melhor correspondem aos blocos de amostras da PU. A unidade de estimação de movimento pode gerar um índice de referência que indica uma posição em RefPicList0 da imagem de referência contendo a região de referência para a PU. Além disso, a unidade de estimação de movimento pode gerar um MV que indica um deslocamento espacial entre um bloco de codificação da PU e uma localização de referência associada com a região de referência. Por exemplo, o MT pode ser um vetor bidimensional que provê um deslocamento a partir das coordenadas da imagem atual decodificada para coordenadas de uma imagem de referência. A unidade de estimação de movimento pode emitir o índice de referência e o MV como a informação de movimento da PU. A unidade de compensação de movimento da unidade de processamento de interpredição 120 pode gerar os blocos preditivos da PU com base em amostras reais ou interpoladas na localização de referência indicada pelo vetor de movimento da PU.

[0111] Se uma PU está em uma fatia B, a unidade de estimação de movimento pode realizar unipredição ou bipredição para a PU. Para executar unipredição para a PU, a unidade de estimação de movimento pode buscar as imagens de referência de RefPicList0 ou uma segunda lista de

imagens de referência ("RefPicList1") para uma região de referência para a PU. A unidade de estimação de movimento pode emitir, conforme a informação de movimento da PU, um índice de referência que indica uma posição em RefPicList0 ou RefPicList1 da imagem de referência que contém a região de referência, um MV que indica um deslocamento espacial entre um bloco de predição da PU e uma localização de referência associada com a região de referência, e um ou mais indicadores de direção de predição que indicam se a imagem de referência está em RefPicList0 ou RefPicList1. A unidade de compensação de movimento da unidade de processamento de interpredição 120 pode gerar os blocos preditivos da PU com base pelo menos em parte em amostras reais ou interpoladas na região de referência indicada pelo vetor de movimento da PU.

[0112] Para executar bidirecional interpredição para uma PU, a unidade de estimação de movimento pode buscar as imagens de referência em RefPicList0 para uma região de referência para a PU e pode também buscar as imagens de referência em RefPicList1 para uma outra região de referência para a PU. A unidade de estimação de movimento pode gerar índices de imagem de referência que indicam as posições em RefPicList0 e RefPicList1 das imagens de referência que contêm as regiões de referência. Além disso, a unidade de estimação de movimento podem gerar MVs que indicam deslocamentos espaciais entre a localização de referência associada com as regiões de referência e um bloco de amostras da PU. A informação de movimento da PU pode incluir os índices de referência e os MVs da PU. A unidade de compensação de movimento pode gerar os blocos preditivos da PU com base pelo menos em parte em amostras reais ou interpoladas nas regiões de referência indicadas pelos vetores de movimento da PU.

[0113] A unidade de processamento de intrapredição 126 pode gerar dados preditivos para uma PU realizando intrapredição na PU. Os dados preditivos para a PU podem incluir blocos preditivos para a PU e vários elementos de sintaxe. A unidade de processamento de intrapredição 126 pode executar intrapredição na PU nas fatias I, fatias P e fatias B.

[0114] Para executar intrapredição na PU, a unidade de processamento de intrapredição 126 pode usar vários modos de intrapredição para gerar vários conjuntos de dados preditivos para a PU. Unidade de processamento de intrapredição 126 pode usar amostras de blocos de amostras de PUs vizinhas para gerar um bloco de predição para uma PU. A PU vizinha pode estar acima, acima e à direita, acima e à esquerda, ou à esquerda da PU, assumindo uma ordem de codificação da esquerda para a direita, de cima para baixo para PUs, CUs e CTUs. A unidade de processamento de intrapredição 126 pode usar vários números de modos de intrapredição, por exemplo, 33 modos de intrapredição direcionais. Em alguns exemplos, o número de modos de intrapredição pode depender do tamanho da região associada com a PU.

[0115] A unidade de processamento de predição 100 pode selecionar os dados preditivos para PUs de uma CU entre os dados preditivos gerados pela unidade de processamento de interpredição 120 para as PUs ou os dados preditivos gerados pela unidade de processamento de intrapredição 126 para as PUs. Em alguns exemplos, a unidade de processamento de predição 100 seleciona os dados preditivos para as PUs da CU com base em métricas / taxa de distorção dos conjuntos de dados preditivos. Os blocos preditivos dos dados preditivos selecionados podem ser aqui referidos como os blocos preditivos selecionados.

[0116] unidade de geração residual 102 pode gerar, com base no bloco de codificação (por exemplo, um bloco de codificação de luma, Cb e Cr) de uma CU e os blocos preditivos selecionados (por exemplo, blocos preditivos de luma, Cb e Cr) das PUs da CU, um bloco residual (por exemplo, um bloco residual de luma, Cb e Cr) da CU. Por exemplo, a unidade de geração residual 102 pode gerar os blocos residuais da CU de tal forma que cada amostra nos blocos residual tem um valor igual a diferença entre uma amostra em um bloco de codificação da CU e uma amostra correspondente de um bloco de predição selecionado correspondente de uma PU da CU.

[0117] A unidade de processamento de transformada 104 pode executar o particionamento quad-tree para particionar os blocos residuais associados a CU para bloco de transformada associado com TUs da CU. Assim, uma TU pode corresponder a um bloco de transformada de luminância e dois blocos de transformada de croma. Os tamanhos e as posições de blocos de transformada de luma e croma de TUs de uma CU podem ou não podem basear-se nos tamanhos e posições dos blocos preditivos das PUs da CU. Uma estrutura quad-tree conhecida como "quad-tree residual" (RQT) pode incluir nós correspondentes a cada uma das regiões. As TUs de uma CU podem corresponder a nós de folha do RQT.

[0118] A unidade de processamento de transformada 104 pode gerar blocos de coeficientes de transformada para cada TU de uma CU através da aplicação de uma ou mais transformadas aos blocos de transformada da TU. A unidade de processamento de transformada 104 pode aplicar várias transformadas a um bloco de transformada associado a uma TU. Por exemplo, unidade de processamento de transformada 104 pode aplicar uma transformada de cosseno discreta (DCT), uma transformada direcional, ou uma transformada

conceitualmente similar a um bloco de transformada. Em alguns exemplos, unidade de processamento de transformada 104 não aplica transformadas a um bloco de transformada. Nestes exemplos, o bloco de transformada pode ser tratado como um bloco de coeficiente de transformada.

[0119] A unidade de quantização 106 pode quantizar os coeficientes de transformada em um bloco de coeficientes. O processo de quantização pode reduzir a profundidade do bit associado com alguns ou todos os coeficientes de transformada. Por exemplo, um coeficiente de transformada de N bits pode ser arredondado para baixo para um coeficiente de transformada de m-bit durante quantização, em que n é maior do que m. A unidade de quantização 106 pode quantizar um bloco de coeficientes associados com uma TU de uma CU com base em um valor de parâmetro de quantização (QP) associado com a CU. O codificador de vídeo 20 pode ajustar o grau de quantização aplicado aos blocos de coeficientes associados com uma CU, ajustando o valor QP associado com a CU. Quantização pode introduzir a perda de informações, assim coeficientes de transformada quantizados podem ter precisão menor do que os originais.

[0120] A unidade de quantização inversa 108 e unidade de processamento de transformada inversa 110 pode aplicar quantização inversa e transformada inversa a um bloco de coeficientes, respectivamente, para reconstruir um bloco residual do bloco de coeficientes. A unidade de reconstrução 112 pode adicionar o bloco residual reconstruído a partir de amostras correspondentes de um ou mais blocos preditivos gerados pela unidade de processamento de predição 100 para produzir um bloco de transformada reconstruído associado a uma TU. Ao reconstruir blocos de transformada para cada TU de uma CU

deste modo, o codificador de vídeo 20 pode reconstruir os blocos de codificação da CU.

[0121] A unidade de filtro 114 pode executar uma ou mais operações de desbloqueio para reduzir artefatos de bloqueio nos blocos de codificação associados a uma CU. O armazenador de imagem decodificada 116 pode armazenar os blocos de codificação reconstruídos após a unidade de filtro 114 realizar as uma ou mais operações de desbloqueio nos blocos de codificação reconstituídos. A unidade de processamento de interpredição 120 pode usar uma imagem de referência que contém os blocos de codificação reconstruídos para executar interpredição sobre PUs de outras imagens. Além disso, unidade de processamento de intrapredição 126 pode usar blocos de codificação reconstruídos em armazenador de imagem decodificada 116 para executar intrapredição em outras PUs na mesma imagem que a CU.

[0122] A unidade de codificação de entropia 118 pode receber dados de outros componentes funcionais do codificador de vídeo 20. Por exemplo, a unidade de codificação de entropia 118 pode receber blocos de coeficientes da unidade de quantização 106 e pode receber elementos de sintaxe da unidade de processamento de predição 100. A unidade de codificação de entropia 118 pode efetuar uma ou mais operações de codificação de entropia nos dados para gerar dados codificados por entropia. Por exemplo, unidade de codificação de entropia 118 pode executar uma operação de codificação de comprimento variável adaptativa de contexto (CAVLC), uma operação CABAC, uma operação de codificação de comprimento variável para variável (V2V), uma operação de codificação aritmética binária adaptativa de contexto baseada em sintaxe (SBAC), uma operação de codificação de Entropia de Particionamento

de Intervalo de Probabilidade (PIPE), uma operação de codificação Exponencial-Golomb, ou outro tipo de operação de codificação de entropia dos dados. O codificador de vídeo 20 pode emitir um fluxo de bits, que inclui dados codificados por entropia gerados pela unidade de codificação de entropia 118. Por exemplo, o fluxo de bits pode incluir dados que representam um RQT para uma CU.

[0123] A figura 3 é um diagrama de blocos que ilustra um decodificador de vídeo exemplar 30 que é configurado para implementar as técnicas desta divulgação. A figura 3 é provida para fins de explicação e não deve ser limitativa sobre as técnicas como amplamente exemplificadas e descritas nesta divulgação. Para efeitos da explicação, esta divulgação descreve decodificador de vídeo 30, no contexto da codificação HEVC. No entanto, as técnicas desta divulgação podem ser aplicáveis a outras normas ou métodos de codificação.

[0124] No exemplo da figura 3, decodificador de vídeo 30 inclui unidade de decodificação de entropia 150, unidade de processamento de predição 152, unidade de quantização inversa 154, unidade de processamento de transformada inversa 156, unidade de reconstrução 158, uma unidade de filtro 160, memória de vídeo 163, e armazenador de imagem decodificada (DPB) 162. Unidade de processamento de predição 152 inclui uma unidade de compensação de movimento 164 e uma unidade de processamento de intrapredição 166. O decodificador de vídeo 30 inclui também uma unidade de decodificação com base em paleta 165 configurada para realizar vários aspectos das técnicas de codificação baseada em paleta descritas nesta divulgação. Em outros exemplos, decodificador de vídeo 30 pode incluir mais, menos ou diferentes componentes funcionais.

[0125] Como será explicado em maior detalhe abaixo com referência às figuras 4-9, unidade de decodificação baseada em paleta 165 do decodificador de vídeo 30 pode ser configurada para receber um vetor de predição binário codificado (por exemplo, um vetor de predição binário codificado com codificação run length) para um bloco atual dos dados de vídeo, decodificar o vetor de predição binário codificado usando uma técnica de decodificação run length, gerar uma paleta para o bloco atual de dados de vídeo com base no vetor de predição binário, o vetor de predição binário compreendendo entradas indicando se ou não entradas de paleta previamente utilizadas são reutilizadas para a paleta para o bloco atual de dados de vídeo, e decodificar o bloco atual de dados de vídeo usando a paleta.

[0126] Um armazenador de imagem codificada (CPB), por exemplo, memória de vídeo 163, pode receber e armazenar dados de vídeo (por exemplo, unidades NAL) de um fluxo de bits. A unidade de decodificação de entropia 150 pode receber dados de vídeo codificados (por exemplo, unidades NAL) do CPB e analisar as unidades NAL para decodificar elementos de sintaxe. Unidade de decodificação de entropia 150 pode decodificar por entropia elementos de sintaxe codificados por entropia nas unidades NAL. Unidade de processamento de predição 152, uma unidade de quantização inversa 154, unidade de processamento de transformada inversa 156, unidade de reconstrução 158, e a unidade de filtro 160 podem gerar dados de vídeo decodificados com base nos elementos de sintaxe extraídos a partir do fluxo de bits.

[0127] Como mostrado na figura 3, a memória de vídeo 163 recebe dados de vídeo que são utilizados para a decodificação de um bloco de vídeo atual dentro de um

quadro de vídeo. A memória de vídeo 163 pode armazenar dados de vídeo para serem decodificados pelos componentes do decodificador de vídeo 30 (por exemplo, configurado para armazenar dados de vídeo). Os dados de vídeo armazenados na memória de vídeo 163 podem ser obtidos, por exemplo, a partir de um fluxo de bits de vídeo codificado produzido pelo codificador de vídeo 20. DPB 162 é um exemplo de DPB que armazena dados de vídeo de referência para utilização na decodificação de dados de vídeo pelo decodificador de vídeo 30 (por exemplo, nos modos de inter ou intracodificação, também conhecido como modos de codificação de intra ou interpredição). A memória de vídeo 163 e DPB 162 podem ser formados por qualquer um de uma variedade de dispositivos de memória, tais como a memória dinâmica de acesso aleatório (DRAM), incluindo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM), ou outros tipos de dispositivos de memória. A memória de vídeo 163 e DPB 162 podem ser providos pelo mesmo dispositivo de memória ou dispositivos de memória separados. Em vários exemplos, memória de vídeo 163 pode ser sobre chip com outros componentes do decodificador de vídeo 30, ou fora do chip em relação a esses componentes.

[0128] As unidades NAL do fluxo de bits podem incluir unidades NAL de fatia codificada. Como parte de decodificar o fluxo de bits, a unidade de decodificação de entropia 150 pode extrair e decodificar por entropia elementos de sintaxe das unidades NAL de fatia codificada. Cada uma das fatias codificadas pode incluir um cabeçalho de fatia de dados e fatias. O cabeçalho da fatia pode conter elementos de sintaxe que pertencem a uma fatia. Os elementos de sintaxe no cabeçalho da fatia podem incluir um

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

[0129] Além de decodificar elementos de sintaxe do fluxo de bits, decodificador de vídeo 30 pode executar uma operação de reconstrução em uma CU não particionada. Para executar a operação de reconstrução em uma CU não particionada, decodificador de vídeo 30 pode executar uma operação de reconstrução em cada TU da CU. Ao realizar a operação de reconstrução para cada TU da CU, decodificador de vídeo 30 pode reconstruir blocos residuais da CU.

[0130] Como parte de executar uma operação de reconstrução em uma TU de uma CU, unidade de quantização inversa 154 pode inverter quantização, ou seja, dequantizar, blocos de coeficiente associados à TU. A unidade de quantização inversa 154 pode utilizar um valor QP associado com a CU da TU para determinar um grau de quantização e, do mesmo modo, um grau de quantização inversa para unidade de quantização inversa 154 para aplicar. Isto é, a taxa de compressão, isto é, a razão entre o número de bits utilizados para representar sequência original e aquele comprimido, podem ser controlados ajustando o valor do QP utilizado ao quantizar os coeficientes de transformada. A taxa de compressão pode também depender do método de codificação de entropia empregue.

[0131] Após a unidade de quantização inversa 154 quantiza inversamente um bloco de coeficientes, unidade de processamento de transformada inversa 156 pode aplicar uma ou mais transformada inversa no bloco de coeficientes de forma a gerar um bloco residual associado à TU. Por exemplo, unidade de processamento de transformada inversa 156 pode aplicar uma DCT inversa, uma transformada de inteiro inversa, uma transformada de Karhunen-Loeve inversa

(KLT), uma transformada rotacional inversa, uma transformada direcional inversa, ou outra transformada inversa ao bloco de coeficientes.

[0132] Se uma PU é codificada utilizando a intrapredição, unidade de processamento de intrapredição 166 pode executar intrapredição para gerar blocos preditivos para a PU. Unidade de processamento de intrapredição 166 pode usar um modo de intrapredição para gerar a blocos de luma, Cb e Cr preditivos para a PU com base nos blocos preditivos de PUs espacialmente vizinhas. A unidade de processamento de intrapredição 166 pode determinar o modo de intrapredição para a PU com base em um ou mais elementos de sintaxe a partir do fluxo de bits decodificados.

[0133] A unidade de processamento de predição 152 pode construir uma primeira lista de imagem de referência (RefPicList0) e uma segunda lista de imagem de referência (RefPicList1) com base em elementos de sintaxe extraídos do fluxo de bits. Além disso, se uma PU é codificada utilizando interpredição, a unidade de decodificação de entropia 150 pode extrair informações de movimento para a PU. A unidade de compensação de movimento 164 pode determinar, com base na informação de movimento da PU, uma ou mais regiões de referência para a PU. A unidade de compensação de movimento 164 pode gerar, com base em amostras de blocos em um ou mais blocos de referência para a PU, blocos preditivos (por exemplo, blocos preditivos de luma, Cb e Cr) para a PU.

[0134] Unidade de reconstrução 158 podem usar os blocos de transformada (por exemplo, bloco de transformada de luma, Cb e Cr) associados com TUs de uma CU e os blocos preditivos (por exemplo, blocos preditivos de luma, Cb e Cr) das PUs da CU, ou seja, tanto os dados de intrapredição

quanto os dados de interpredição, conforme o caso, para reconstruir os blocos de codificação (por exemplo, blocos de codificação de luma, Cb e Cr) da CU. Por exemplo, a unidade de reconstrução 158 pode adicionar amostras aos blocos de transformada (por exemplo, blocos de transformada de luma, Cb e Cr) para amostras correspondentes dos blocos preditivos (por exemplo, blocos preditivos de luma, Cb e Cr) para reconstruir os blocos de codificação (por exemplo, blocos de codificação de luma, Cb e Cr) da CU.

[0135] A unidade de filtro 160 pode executar uma operação de desbloqueio para reduzir os artefatos de bloqueio associados com os blocos de codificação (por exemplo, blocos de codificação de luma, Cb e Cr) da CU. Decodificador de vídeo 30 pode armazenar os blocos de codificação (por exemplo, blocos de codificação de luma, Cb e Cr) da CU no armazenador de imagem decodificada 162. Armazenador de imagem decodificada 162 pode prover imagens de referência para compensação de movimento subsequente, intrapredição e apresentação em um dispositivo de exibição, como o dispositivo de exibição 32 da figura 1. Por exemplo, decodificador de vídeo 30 pode realizar, com base nos blocos (por exemplo, blocos de luma, Cb e Cr) no armazenador de imagem decodificada 162, a operações de intrapredição ou interpredição na PU de outra CU.

[0136] A figura 4 é um diagrama de blocos, mostrando unidade de codificação baseada em paleta 122 do codificador de vídeo 20 com mais detalhes. A unidade de codificação baseada em paleta 122 pode ser configurada para executar uma ou mais das técnicas exemplares desta divulgação para a codificação de um vetor de predição binário.

[0137] Como descrito acima, a unidade de codificação baseada em paleta 122 pode ser configurada para

codificar um bloco de dados de vídeo (por exemplo, uma CU ou PU) com um modo de codificação baseada em paleta. Em um modo de codificação baseada em paleta, uma paleta pode incluir entradas numeradas por um índice e representar valores de componentes de cor (por exemplo, RGB, YUV etc.) ou intensidades que podem ser utilizadas para indicar os valores de pixel. A unidade de geração de paleta 203 pode ser configurada para receber de pixel 212 para valores de um bloco atual de dados de vídeo e gerar uma paleta de valores de cor para o bloco atual de dados de vídeo. A unidade de geração de paleta 203 pode usar todas as técnicas para gerar uma paleta para um bloco atual de dados de vídeo, incluindo as técnicas baseadas em histograma acima discutidas. A unidade de geração de paleta 203 pode ser configurada para gerar uma paleta de qualquer tamanho. Em um exemplo, a unidade de geração de paleta 203 pode ser configurada para gerar 32 entradas de paleta, em que cada entrada de paleta inclui valores de pixel para os componentes Y, Cr, e Cb de um pixel. No primeiro exemplo, assume-se que cada entrada de paleta especifica os valores para todos os componentes de cor de uma amostra (pixel). No entanto, os conceitos descritos neste documento são aplicáveis à utilização de uma paleta separada para cada componente de cor.

[0138] Uma vez que uma paleta é gerada pela unidade de geração de paleta 203, a unidade de mapa 204 pode gerar um mapa para o bloco atual de dados de vídeo que indica se ou não um elemento de imagem particular no bloco atual de dados de vídeo pode ser representado por uma entrada na paleta gerada pela unidade de geração de paleta 203. A unidade de mapa 204 pode produzir um mapa 214 que inclui elementos de sintaxe que indicam como cada pixel usa (ou não usa) as entradas da paleta. Se o valor para um

pixel no bloco atual de dados de vídeo não é encontrado na paleta, e, portanto, não pode ser representado por um índice para a paleta, a unidade de mapa 204 pode explicitamente transmitir um valor de pixel para o pixel particular. Em alguns exemplos, a unidade de mapa 204 pode predizer o valor de pixel explícito de uma das entradas encontradas na paleta. Em outros exemplos, a unidade de mapa 204 pode quantizar o pixel e transmitir os valores quantizados.

[0139] Além disso, para sinalizar elementos de sintaxe que indicam os valores das cores utilizadas para cada um dos pixels de um bloco, unidade de codificação baseada em paleta 122 pode também ser configurada para sinalizar a paleta que deve ser utilizada para um bloco atual de dados de vídeo. De acordo com as técnicas desta divulgação, a unidade de codificação baseada em paleta 122 pode ser configurada para utilizar técnicas de predição de paleta para reduzir a quantidade de dados que é sinalizada para indicar os valores de uma paleta para um determinado bloco de dados de vídeo.

[0140] Como um exemplo de predição de paleta, como é descrito em JCTVC-Q0094, que é disponível a partir de 20 de junho de 2014 a partir de http://phenix.int-evry.fr/jct/doc_end_user/documents/17_Valencia/wg11/-JCTVC-Q0094-v1.zip, uma paleta pode incluir entradas que são copiadas a partir de uma paleta de preditor. Uma paleta de preditor pode incluir entradas de paleta de blocos previamente codificados que utilizam o modo de paleta ou outras amostras reconstruídas. Como mostrado na figura 4, unidade de codificação baseada em paleta 122 pode incluir um armazenador de paleta de preditor 210. O armazenador de paleta de preditor 210 pode ser configurado para armazenar uma série de entradas de paleta previamente utilizadas a

partir de blocos previamente codificados. Como um exemplo, armazenador de paleta de preditor 210 pode ser configurado como um armazenador de primeiro entrada e primeira saída (FIFO) de um tamanho predeterminado. O armazenador de paleta de preditor 210 pode ser de qualquer tamanho. Em um exemplo, armazenador de paleta de preditor 210 inclui até 64 entradas de paleta previamente utilizadas.

[0141] Em alguns exemplos, unidade de codificação baseada em paleta 122 pode ser configurada para podar as entradas em armazenador de paleta de preditor 210 de tal modo que todas as entradas de paleta no armazenador de paleta de preditor 210 são exclusivas. Ou seja, para cada nova entrada de paleta a ser adicionada ao Armazenador de paleta de preditor 210, a unidade de codificação baseada em paleta 122 pode ser configurada para primeiro verificar que não existem outras entradas idênticas já armazenadas no armazenador de paleta de preditor 210. Se não houver entradas idênticas, a nova entrada de paleta é adicionada ao armazenador de paleta de preditor 210. Se a nova entrada é uma duplicata de uma entrada existente, a nova entrada de paleta é adicionada ao armazenador de paleta de preditor 210 e as entradas duplicadas são removidas do armazenador de paleta de preditor 210.

[0142] A unidade de codificação baseada em paleta 122 pode incluir uma unidade de geração de vetor de predição binário 206 que é configurado para gerar e sinalizar um flag binário, para cada entrada em uma paleta para um bloco atual de dados de vídeo gerados pela unidade de geração de paleta 203, para indicar se uma entrada de paleta no armazenador de paleta de preditor 210 é copiada (ou reutilizada) para uma das entradas na paleta para o bloco atual de dados de vídeo (por exemplo, indicado pelo flag = 1). Isto é, um flag com um valor de 1 no vetor de

preditor binário indica que a entrada correspondente no armazenador de paleta de preditor 210 é reutilizada para a paleta para o bloco atual, enquanto um flag com um valor de 0, no vetor de predição binário indica que a entrada correspondente no armazenador de paleta de preditor 210 não é reutilizada para a paleta para o bloco atual. Além disso, a unidade de codificação baseada em paleta 122 pode ser configurada para sinalizar explicitamente alguns valores da paleta atual que não podem ser copiados a partir de entradas no armazenador de paleta de preditor 210. O número de novas entradas pode ser sinalizado como bem.

[0143] No pedido de US Provisório No. 61/970.257, depositado em 25 de março de 2014; pedido de US Provisório No. 61/981.105, depositado em 17 de abril de 2014; e pedido de US Provisório No. 62/002.668, depositado em 23 de maio de 2014, um método de sinalização baseado em árvore binária e métodos de sinalização baseados em fim de posição com foram propostos para codificação do vetor de preditor binário de paleta. No pedido de US Provisório No. 62/002.741, depositado em 23 de maio de 2014, um método de sinalização baseado em grupo foi proposto. Esta descrição propõe técnicas adicionais para a geração, codificação e decodificação do vetor de predição binário.

[0144] Alguns exemplos aqui descritos referem-se a métodos para codificar o vetor de predição de paleta para melhorar a eficiência de codificação. Por exemplo, suponha que o vetor de predição binário gerado pela unidade de geração de vetor de predição binário 206 é indicado por:

$$b = [b_0, b_1 \ldots b_{N-1}], N \geq 0, b_i \{0,1\}, 0 \leq i \leq N$$

Na equação acima, $b_i \{0,1\}, 0 \leq i \leq N$ denota um flag de predição (também chamado de um flag binário ou flag de predição binário). Se $N=0, b=$ (isto é, b é o vetor vazio), que não

precisa de ser sinalizado. Portanto, na descrição seguinte, pode ser assumido que $N>0$

[0145] A figura 5 mostra um exemplo de um armazenador de paleta de preditor 210 e uma paleta atual 220. Como pode ser visto na figura 5, paleta atual 220 reutiliza os valores de pixel do armazenador de preditor de paleta 210 associado com índices de entrada 1, 2, 5 e 9. Como tal, um vetor de preditor binário produzido pela unidade de geração de vetor de predição binário 206 da figura 4 seria b = [110010001000]. Como pode ser visto, neste exemplo, o vetor de predição binário b inclui flags com um valor de 1 corresponde ao 1°, 2°, 5°, e 9° índices no armazenador de paleta de preditor 210. Isto é, as 1°, 2°, 5° e 9° entradas no armazenador de preditor de paleta 210 são as únicas entradas reutilizadas para a paleta atual 220. Para índices entrada 5-8 na paleta atual 220, unidade de codificação baseada em paleta 122 pode ser configurada para sinalizar valores de entrada de paleta no fluxo de bits de vídeo codificado (por exemplo, usando a sinalização explícita ou outra técnica de predição).

[0146] De acordo com uma ou mais técnicas desta divulgação, um codificador de vídeo 20 pode ser configurado para codificar ou geralmente codificar o vetor preditor binário b, a fim de reduzir a quantidade de dados necessários para sinalizar uma paleta no fluxo de bits de vídeo codificado. Como mostrado na figura 4, a unidade de compressão de vetor de predição binário 209 pode ser configurada para gerar e sinalizar vetor de predição binário codificado 215. No entanto, deve ser entendido que as técnicas de compressão de vetor de predição binário desta divulgação podem ser aplicadas em outras estruturas do codificador de vídeo 20, incluindo unidade de codificação de entropia 118.

[0147] Em um exemplo da descrição, a unidade de compressão de vetor de predição binário 209 pode ser configurada para codificar o vetor de predição binário usando uma técnica de codificação run length. Por exemplo, a unidade de compressão de vetor de predição binário 209 pode ser configurada para codificar o vetor de predição binário através da indicação do número de ‘0s’ consecutivos entre ‘1s’ no vetor de predição binário utilizando um código de Exponencial-Golomb. Como um exemplo, suponha que novamente b = [110010001000]. Neste exemplo, como mostrado na figura 6, o vetor de predição binário (isto é, b) pode ser expresso como: 'zero consecutivos 0s’ –‘1’ - 'zero consecutivo 0s’- ‘1’ – ‘dois consecutivo 0s’ – ‘1’ - 'três consecutivo 0s’ – ‘1’ – e ‘quatro consecutivos 0s'. Porque é sabido que b_i {0,1}, exceto para o último grupo 'consecutivo 0', cada grupo 'consecutivo 0' deve ser seguido por um "1". Portanto, a unidade de compressão de vetor de predição binário 209 pode utilizar técnicas de codificação run-length baseada em zero para representar o vetor de predição binário b como 'zero consecutivo 0' - 'zero consecutivo 0' - 'dois consecutivo 0' - 'três consecutivo 0' - 'quatro consecutivo 0', que pode ser expresso como a sequência run length "0-0-2-3-4 '.

[0148] De acordo com um ou mais exemplos desta divulgação relacionada com sinalização baseada em run-length, para codificar a sequência run length, um código de Golomb-Rice, código Exponencial-Golomb de qualquer ordem, o código Exponencial-Golomb Truncado, código Rice truncado ou qualquer outro, incluindo binarizações truncadas, podem ser utilizados. Em um exemplo, vetor de predição binário de unidade de compressão 209 utiliza um código de Golomb Exponencial de 0-ésima ordem como a técnica de codificação run length.

[0149] Para a binarização truncada, o maxsymbol pode ser o valor máximo possível da execução, dependendo da posição de "1" no vetor binário e o tamanho do vetor binário, uma vez que, ao mover-se para o fim do vetor binário, o máximo valor possível de execução é reduzido do tamanho do vetor para 0, dependendo da posição dentro do vetor. Por exemplo, o símbolo max pode ser o comprimento do vetor binário ou o comprimento do vetor binário menos a posição do "1" a partir do qual a execução está zendo contada. Em outras palavras ele é o comprimento medido restante a partir da extremidade do vetor binário. Para o exemplo acima com o vetor binário b de um tamanho particular, por exemplo, 13, a sequência de run-length '0-0-2-3-4' pode ser codificada com a binarização truncada '0[13]-0[12]-2[11]-3[8]-4[4]', em que o símbolo máximo é dado entre parênteses.

[0150] Além disso, em alguns exemplos, binarização pode ser dependente da posição ou índice do elemento (0 ou 1) no vetor binário. Como um exemplo particular, se a posição é menor do que um certo limite, um tipo de binarização é utilizada; de outra forma, um outro tipo de binarização é aplicada. Em alguns exemplos, o tipo de binarização pode ser códigos de binarização diferentes, ou da mesma família de códigos, mas com ordem diferente, tal como Código Exponencial-Golomb.

[0151] Em um exemplo, o limite pode ser o comprimento de paleta a partir do bloco anterior ou paleta de bloco codificado anterior. Em outro exemplo, o limite pode ser fixado para um valor padrão ou sinalizado por bloco, imagens, fatia ou em outro local. Deve ser reconhecido que uma técnica correspondente pode, opcionalmente, ser utilizada para definir um contexto CABAC para codificar os valores de execução. Além disso, a

unidade de codificação baseada em paleta 122 pode ser configurada para parar sinalização run-length quando o número de elementos sinalizados "1" (ou seja, o número de entradas de paleta a partir de armazenador de paleta de preditor 210 indicado como sendo reutilizado para a paleta atual 220) atinge um número máximo possível. Em alguns exemplos, o número máximo possível é o tamanho máximo possível de paleta.

[0152] Alguns exemplos esta divulgação refere-se a codificação de posição final da sequência run length indicando o vetor de predição binário b. Em um ou mais exemplos desta divulgação, a unidade de compressão de vetor de predição binário 209 pode ser configurada para codificar o vetor de predição binário b utilizando um run length reservado L para codificar a posição final do vetor de predição binário. Em um exemplo, L = 1 é utilizado como a run length reservado. No codificador de vídeo 20, se a run length é igual a ou maior do que L, a unidade de compressão de vetor de predição binário 209 é configurada para adicionar q ao run length. Se o run length real é menor que L, unidade de compressão de vetor de predição binário 209 é configurada para sinalizar o run-length como ele é. Unidade de compressão de vetor de predição binário 209 pode sinalizar o run-length de posição final com o run-length reservado L.

[0153] Do mesmo modo, no decodificador de vídeo 30, se o valor decodificado de um run-length é maior do que L, 1 é subtraído do run-length real. Se o valor decodificado ou um run-length é menor do que L, o valor decodificado é utilizado como o run-length real. Se o valor decodificado é igual a L, as posições restantes no vetor de predição binário b são todas 0. Assim, se o valor

decodificado é igual a L, sinalização de execução não é mais necessária.

[0154] Utilizar o mesmo exemplo como acima (ou seja, b = [1100100001000]) e assumindo que L = 1, unidade de compressão de vetor de predição binário 209 está configurada para sinalizar a sequência de run-length ‘0-0-2-3-4' da figura 6 como ‘0-0-3-4-1’. Em seguida, aplicar as regras acima, decodificador de vídeo 30 pode ser configurado para recuperar a sequência de run-length, como ‘0-0-2-3-fim'. Ou seja, o primeiro valor run-length de 0 é decodificado como 0 e a próxima sequência de run-length de 0 é decodificada como 0, pois ambas das sequências de run-length 0 são menores do que o valor run length reservado de L = 1. A próxima sequência de run length é 3, e como tal, decodificador de vídeo 30 poderia ser configurado para subtrair 1 do valor de 3 para se obter 2, porque o valor recebido de 3 é maior do que o valor de run length reservado de L = 1. Da mesma forma, um decodificador de vídeo 30 poderia ser configurado para subtrair 1 do valor recebido de 4 para se obter 3 para a próxima sequência run length, porque o valor recebido de 4 é maior do que o valor de run length reservado de L = 1. Finalmente, o último valor run-length recebido 1 é igual ao valor run length reservado de L = 1. Por conseguinte, decodificador de vídeo 30 pode determinar que não há mais valores de ‘1’ presentes no vetor de predição binário.

[0155] Em um outro exemplo da descrição, a unidade de compressão de vetor de predição binário 209 pode ser configurado para aplicar somente a posição final de codificação quando o número total de "1s" no vetor de predição binário (isto é, o número de indicações de entradas de paleta reutilizadas a partir de armazenador de paleta de preditor 210) é menor do que o tamanho máximo

possível de paleta. Se o número total de "1s" no vetor de predição binário é igual a um tamanho máximo possível de paleta, a unidade de compressão de vetor de predição binário 209 pode ser configurada para ignorar a sinalização do último run length.

[0156] No exemplo acima, se o decodificador de vídeo 30 determina que o tamanho máximo possível de paleta é 4, em seguida, uma sequência run length de '0-0-2-3' (ou sequência run-length 0-0-3-4, de acordo com as regras de run length reservado descritas acima) pode ser suficiente para recuperar o vetor de predição binário b. Em alguns exemplos, o tamanho máximo possível de paleta pode ser predeterminado. Por conseguinte, o tamanho máximo possível de paleta pode ser previamente determinado por um codificador de vídeo 20 e decodificador de vídeo 30. Em outros exemplos, o tamanho máximo possível de paleta pode ser sinalizado por um codificador de vídeo 20 no fluxo de bits de vídeo codificado. Sinalização do tamanho máximo possível de paleta pode ser realizada utilizando quaisquer técnicas de comunicação de dados convencional adequadas ou quaisquer técnicas de comunicação de dados adequadas aqui descritas.

[0157] Em um outro exemplo da descrição, a unidade de compressão de vetor de predição binário 209 pode ser configurada para aplicar técnicas de codificação de posição final acima descritas apenas quando o última run length na sequência run length não é 0. Se o último run-length na sequência de run-length é 0, a posição final de codificação pode ser ignorada. Esta afirmação é o equivalente a quando o último flag no vetor de predição binário b é igual a 1. Nesta situação, a codificação de posição final pode ser ignorada. Por exemplo, se b = [100000000001], então a sequência de run-length é '0-10-0'.

A sequência de run-length de '0-10' (ou sequência de valor sinalizado 0-11, de acordo com as regras de run-length reservado descritas acima) pode ser suficiente para recuperar o vetor de predição binário b. Além disso, a posição final pode ser sinalizado sem sinalizar quaisquer execuções para indicar que o vetor de predição binário b é um vetor zero, significando que nenhuma entrada de paleta é predita.

[0158] Em alguns exemplos, em vez de usar um valor fixo L para sinalizar que os flags binários restantes são 0, um mínimo de L e o tamanho do vetor preditor binário pode ser utilizado para indicar a posição final. Isto é porque o valor da execução é sempre menor do que o tamanho do vetor preditor binário.

[0159] A figura 7 é um diagrama de blocos, mostrando um exemplo de unidade de decodificação baseada em paleta 165 do decodificador de vídeo 30. A unidade de decodificação baseada em paleta 165 pode ser configurada para executar de uma forma recíproca a unidade de codificação baseada em paleta 122 da figura 4. A unidade de codificação baseada em paleta 165 pode ser configurada para receber um mapa 312 que indica, para cada pixel em um bloco atual, se ou não entradas para uma paleta serão utilizadas para os pixels no bloco atual. Além disso, o mapa 312 pode ainda indicar que entradas de paleta estão sendo utilizadas para um dado pixel. A unidade de mapa 302 pode decodificar o bloco atual de dados de vídeo usando o mapa 312 e uma paleta gerada pela unidade de geração de paleta 304 para produzir dados de vídeo decodificado 314.

[0160] De acordo com as técnicas desta divulgação, unidade de decodificação baseada em paleta 165 pode também receber um vetor de predição binário codificado 316. Como discutido acima, vetor de predição binário 316

pode ser codificado utilizando uma técnica de codificação run length que codifica uma sequência run length indicando uma série de valores de zero no vetor de predição de binário. Vetor de predição binário da unidade de descompressão 306 pode ser configurado para decodificar o vetor de predição binário codificado utilizando qualquer combinação de técnicas de codificação run length das sequências descritas acima com referência às figuras 4-6. Uma vez um vetor de predição binário é recuperado pela unidade de descompressão de vetor de predição binário 306, a unidade de geração de paleta 304 pode gerar uma paleta para o bloco atual de dados de vídeo com base no vetor de predição binário e entradas de paleta previamente utilizadas armazenadas no armazenador de paleta de preditor 310. Unidade de decodificação baseada em paleta 165 pode ser configurada para armazenar entradas de paleta previamente utilizadas no armazenador de paleta de preditor 310 da mesma maneira que unidade de codificação baseada em paleta 122 armazenou entradas de paleta previamente utilizadas no armazenador de paleta de preditor 210.

[0161] A figura 8 é um fluxograma que ilustra um exemplo do método de codificação da divulgação. As técnicas da figura 8 podem ser implementadas por um ou mais componentes estruturais de um codificador de vídeo 20, incluindo a unidade de codificação baseada em paleta 122. Em um exemplo da descrição, unidade de codificação baseada em paleta 122 pode ser configurada para gerar uma paleta para o bloco atual de dados de vídeo (800), e gerar um vetor de predição binário para a paleta para o bloco atual de dados de vídeo (802). O vetor de predição binário compreende entradas que indicam se ou não entradas de paleta previamente utilizadas são reutilizadas para a paleta para o bloco atual de dados de vídeo. A unidade de

codificação baseada em paleta 122 pode ser adicionalmente configurada para codificar o vetor de predição binário usando uma técnica de codificação run length (804), e codificar o bloco atual de dados de vídeo, utilizando a paleta (806). Em um exemplo, a técnica de codificação run length compreende codificar um run length de zeros.

[0162] Em um exemplo da descrição, a unidade de codificação baseada em paleta 122 pode ser configurada para codificar o vetor de predição binário usando uma técnica de codificação Exponencial-Golomb. Em um exemplo, a técnica de codificação exponencial Golomb é uma técnica de codificação Exponencial-Golomb de 0-ésima ordem.

[0163] Em um outro exemplo da divulgação, unidade de codificação baseada em paleta 122 pode ser configurada para codificar o vetor de predição binário utilizando a técnica de codificação run length e um valor de run length reservado L, o run length reservado L indicando um valor de posição final do vetor de predição binário. Em um exemplo, o valor de run length reservado L é 1.

[0164] Em um outro exemplo da divulgação, unidade de codificação baseada em paleta 122 pode ser configurada para codificar o vetor de predição binário utilizando a técnica de codificação run length, um valor de run length reservado L, e um tamanho máximo de paleta. O valor run length reservado L indica uma posição final do vetor de predição binário. Neste exemplo, o valor de run length reservado L não é utilizado, se um número total de entradas no vetor de predição binário que indica que entradas de paleta previamente utilizadas são reutilizadas para a paleta para o bloco atual de dados de vídeo é igual ao tamanho máximo de paleta.

[0165] Em outro exemplo da divulgação, unidade de codificação baseada em paleta 122 pode ser configurada para

codificar o vetor de predição binário usando a técnica de codificação run length, e um valor run-length reservado L, o valor run length reservado L indicando uma posição final do vetor de predição binário. Neste exemplo, o valor run length reservado L não é utilizado se um último run length no vetor de predição binário codificado não indica um run length de zero.

[0166] Em um outro exemplo da divulgação, unidade de codificação baseada em paleta 122 pode ser configurada para armazenar entradas de paleta previamente utilizadas para um ou mais previamente blocos de dados de vídeo codificados no armazenador. A unidade de codificação baseada em paleta 122 pode ser adicionalmente configurada para remover entradas duplicadas das entradas de paleta previamente utilizadas armazenadas no armazenador. Em um outro exemplo da divulgação, as entradas de paleta previamente utilizadas para os um ou mais blocos de dados de vídeo previamente codificados compreendem entradas de paleta previamente utilizadas para uma linha de pixels acima do bloco atual de dados de vídeo e entradas de paleta previamente utilizadas para uma linha de pixels à esquerda do bloco atual de dados de vídeo.

[0167] A figura 9 é um fluxograma que ilustra um exemplo do método de decodificação da divulgação. As técnicas da figura 9 podem ser implementadas por um ou mais componentes estruturais do decodificador de vídeo 30, incluindo a unidade de decodificação baseada em paleta 165.

[0168] Em um exemplo da divulgação, unidade de decodificação baseada em paleta 165 pode ser configurada para receber um vetor de predição binário codificado para um bloco atual de dados de vídeo (900), e decodificar o vetor de predição binário codificado usando uma técnica de decodificação run length (902). Em um exemplo, a técnica de

decodificação run length compreende codificar um run length de zeros. A unidade de decodificação baseada em paleta 165 pode ser adicionalmente configurada para gerar uma paleta para o bloco atual de dados de vídeo com base no vetor de predição binário (904). O vetor de predição binário compreendendo entradas indicando se ou não entradas de paleta previamente utilizadas são reutilizadas para a paleta para o bloco atual de dados de vídeo. A unidade de decodificação baseada em paleta 165 pode ainda ser configurada para decodificar o bloco atual de dados de vídeo usando a paleta (906).

[0169] Em um outro exemplo da divulgação, unidade de decodificação baseada em paleta 165 pode ser adicionalmente configurada para decodificar o vetor de predição binário codificado usando uma técnica de decodificação Exponencial-Golomb. Em um exemplo, a técnica de decodificação de Exponencial Golomb é uma técnica de decodificação Exponencial-Golomb de 0-ésima ordem.

[0170] Em um outro exemplo da divulgação, unidade de decodificação baseada em paleta 165 pode ser adicionalmente configurada para decodificar o vetor de predição binário codificado utilizando a técnica de decodificação run length e um valor de run length reservado G, o valor de run length reservado L indicando uma posição final do vetor de predição binário. Em um exemplo, o valor de run length reservado L é 1.

[0171] Em um outro exemplo da divulgação, unidade de decodificação baseada em paleta 165 pode ser adicionalmente configurada para decodificar o vetor de predição binário codificado utilizando a técnica de decodificação run length, um valor de run length reservado L, e um tamanho máximo de paleta. O valor run length reservado L indica uma posição final do vetor de predição

binário. O valor run length reservado L não é utilizado se um número total de entradas no vetor de predição binário que indica que entradas de paleta previamente utilizadas são reutilizadas para a paleta para o bloco atual de dados de vídeo é igual ao tamanho máximo de paleta.

[0172] Em um outro exemplo da divulgação, unidade de decodificação baseada em paleta 165 pode ser adicionalmente configurada para decodificar o vetor de predição binário codificado utilizando a técnica de decodificação run length, e um valor run length reservado L, o valor de run length reservado L indicando uma posição final do vetor de predição binário. O valor run length reservado L não é utilizado se um último run length no vetor de predição binário codificado não indica um run length de zero.

[0173] Em um outro exemplo da divulgação, unidade de decodificação baseada em paleta 165 pode ser adicionalmente configurada para copiar, a partir de um armazenador, entradas de paleta previamente utilizadas para a paleta, que são indicadas como sendo reutilizadas para a paleta pelo vetor de predição binário, e receber, no caso em que o número de entradas de paleta previamente utilizadas copiadas para a paleta é inferior a um tamanho máximo de paleta, entradas de paleta adicionais.

[0174] Em um outro exemplo da divulgação, unidade de decodificação baseada em paleta 165 pode ser adicionalmente configurada para armazenar entradas de paleta previamente utilizadas para um ou mais dos blocos de dados de vídeo previamente decodificados no armazenador. A unidade de decodificação com base em paleta 165 pode ser adicionalmente configurada para remover entradas duplicadas das entradas de paleta previamente utilizadas armazenadas no armazenador. Em um exemplo, as entradas de paleta

previamente utilizadas para os um ou mais blocos de dados de vídeo previamente decodificados compreendem entradas de paleta previamente utilizadas para uma linha de pixels acima do bloco atual de dados de vídeo e entradas de paleta previamente utilizadas para uma linha de pixels à esquerda do bloco atual de dados de vídeo.

[0175] As seções seguintes descrevem técnicas exemplares adicionais de divulgação. Em um primeiro exemplo, o codificador de vídeo 20 pode ser configurado para primeiro sinalizar o número total de uns ('núm-de-um'), isto é, o número de entradas de paleta reutilizadas a partir de armazenador de paleta de preditor 210, no vetor de predição binário b usando um código de Golomb-Rice, um código de Exponencial-Golomb, ou um código Rice truncado, ou sua combinação. Então, uma vez que o número de flags de predição binários diferentes de zero decodificados atinge o 'núm-de-um' sinalizado, decodificador de vídeo 30 pode determinar que os flags restantes no vetor de predição binário b são 0 e sinalização destes flags foi contornada pelo codificador de vídeo 20. Em outras palavras, neste primeiro exemplo, o procedimento de codificação de posição final acima descrito pode ser contornado. Neste caso, os run lengths reais são sinalizados. Em um exemplo, não é necessário ajuste para cima, para os valores de run-length.

[0176] Em um segundo exemplo, tal como descrito acima, apenas o run-length de valores zero no vetor de predição binário é codificado. Como uma técnica alternativa, a codificação run length pode incluir (por exemplo, permitir) ambos run lenghts de valores zero no vetor de predição binário e/ou run lengths de valores um do vetor de predição binário. A execução inicial pode ser predefinida para ser um "run length de um' (ou 'run length de zero"). Em seguida, o codificador de vídeo 20 pode

alternar codificação das execuções de zeros e execuções de uns.

[0177] Por exemplo, utilizando o mesmo exemplo como acima (ou seja, b = [1 100100010000]) e assumindo que a primeira execução é 'run length de um', b = [1 100100010000] pode ser expresso como:

'Run = 2' - 'Run = 2' - 'Run = 1' -'Run = 3'-'Run = 1' - 'Run = 4'

No entanto, sabe-se que 'execução zero' deve ser seguido 'execução de um' e vice-versa; Por conseguinte, a expressão anterior pode ser simplificada como:

'Run = 2' - 'Run = 1' -'Run = 0' - 'Run = 2' - 'Run = 0' - 'Run = 3'

que pode ser expressa como sequência run length como '210203'. codificação Golomb-Rice, codificação Exponencial-Golomb, ou codificação Rice truncada podem ser utilizadas para codificar esta sequência. 'Execução zero' e 'execução um' podem ter diferentes parâmetros de um código de Golomb-Rice, código Exponencial-Golomb, ou código Rice truncado. Diferentes contextos podem ser utilizado para 'execução zero' e 'execução um' se faixas regulares são utilizadas para codificá-los. As técnicas de codificação de posição final descritas em outra parte nesta divulgação podem também ser aplicadas no presente exemplo.

[0178] Além disso, o codificador de vídeo 20 pode ser configurado para parar sinalização run-length quando o número de elementos sinalizados '1' atinge um número máximo possível, o que pode ser o tamanho máximo possível de paleta.

[0179] Em um terceiro exemplo, b0, o primeiro elemento no vetor de predição binário b, pode ser sinalizado diretamente para que o decodificador de vídeo 30

possa determinar se a primeira execução é 'execução zero' ou 'execução um' com base no valor de decodificação b0.

[0180] Em um quarto exemplo, o codificador de vídeo 20 pode ser configurado para codificar os primeiros M flags, isto é, [b0 , b1, ..., b_{M-1}] no vetor de predição binário usando uma primeira técnica de codificação e o resto, ou seja, [bM, b_{M+1}, ..., B_{N-1}] são codificados usando uma segunda técnica de codificação. A segunda técnica de codificação é o método baseado em run-length especificado acima. Em um exemplo, M = 4. A primeira técnica de codificação pode ser de qualquer técnica de codificação convencional adequada. Em alguns exemplos, a primeira técnica de codificação pode ser um método baseado em codificação não run length. Em outros exemplos, as primeira e segunda técnicas de codificação podem ser ambas técnicas de codificação baseada em run-length, mas a primeira técnica de codificação e a segunda técnica de codificação podem ser diferentes umas das outras.

[0181] Em um exemplo sinalização baseada em run length para ambos os elementos binários, como mencionado acima, o codificador de vídeo 20 pode ser configurado para sinalizar o run length m para '0' ou '1' para o vetor de predição binário b. No entanto, os dois podem ser combinados onde o run-length é sinalizado para ambos os elementos binários. Isto pode ser realizado através das seguintes etapas: (1) no inicio, um valor de faixa é selecionado, por exemplo, faixa = 0; (2) sinalizar o run length para o elemento de faixa, (3) se faixa é igual a 1, o último sinal de flag indicando se a faixa igual a 1 é o último elemento '1' no vetor binário. Por exemplo, última igual a 1 indica que é a última '1', e igual a 0, caso contrário. Se última é igual a 1, parar sinalização de elementos do vetor binários e sair; (4) valor de faixa é

trocado, ou seja, faixa = 1 – faixa e ir para a etapa (2) acima.

[0182] Se o primeiro valor de faixa é selecionado para ser 0, então pode haver uma necessidade de indicar o caso quando o vetor binário é zero. Essa divulgação pode referir-se a um flag que indica se o vetor binário é zero como zero_flag. Por exemplo, pode ser utilizado um flag separado no início da sinalização de run lengths. O flag separado pode ser igual a 1, indicando que o vetor binário é zero. Neste caso, não há nenhuma sinalização run length depois de zero_flag. Se flag zero é 0, então os valores de run-length são sinalizados. Se o primeiro faixa é escolhido para ser igual a 1, então pode não haver necessidade para sinalizar este zero_flag, porque a faixa igual a 1 tem o last_flag indicando o fim de sinalização.

[0183] Em um exemplo, o algoritmo básico discutido acima pode ser ainda melhorado usando uma ou mais das técnicas na lista a seguir. Cada item na lista pode ser referido como um "produto melhorado" na descrição abaixo seguinte a esta lista:

1. A seleção de faixa pode ser fixa para ser 0 ou 1, ou ser adaptativa (ou seja, ser diferente para vários blocos). Por exemplo, faixa pode ser escolhida (isto é, selecionada), com base nos blocos anteriores ao identificar qual elemento é o primeiro elemento ou mais frequente, que é o mais frequente nos blocos codificados previamente. Em alternativa, a faixa pode ser sinalizada para cada bloco, ou pode ser sinalizada para um grupo de blocos (por exemplo, para cada mosaico ou no cabeçalho de fatia. Em alternativa, a faixa pode ser sinalizada por imagem em uma sequência de vídeo, por exemplo, em pelo menos um do conjunto de parâmetros de imagem (PPS), conjunto de

parâmetros de sequência (SPS), conjunto de parâmetros de vídeo (VPS), ou em outro lugar.

2. O último flag não pode ser sinalizado se o elemento de vetor binário ‘1’ para o qual este último flag é sinalizado é o último elemento do vetor binário b.

3. Após o processamento do primeiro elemento no vetor binário, o valor de run length sinalizado pode ser o valor de run length real menos 1, porque se o elemento não é o último, então sabe-se que ele é seguido por pelo menos um outro elemento binário.

4. O valor de run length para o último elemento '1' no vetor binário pode não ser sinalizado se o valor de run length para o último elemento '1' no vetor binário é precedido pelo elemento ‘0’, uma vez que se o valor run-length '0' é interrompido antes do último elemento do vetor binário, é sabido que este elemento é ‘1’ e é o ‘último’ elemento no vetor.

5. Alternativamente, para a sinalização zero_flag, o valor de run length para o último elemento '1' no vetor binário pode ser incluído no primeiro valor de run length em que o valor específico é atribuído a este. Por exemplo, se o primeiro valor de run length é zero, o vetor binário é zero e não existe qualquer outra sinalização depois disso. Para vetor binário diferente de zero, o primeiro valor run-length sinalizado é necessário ser aumentado em 1 após aquele valor específico, 0 neste exemplo, para evitar ambiguidades.

6. O zero_flag e last_flag pode ser codificado por codificação aritmética binária adaptativa de contexto (CABAC), onde, por exemplo, o contexto separado pode ser atribuído para cada flag. Contextos podem ser dependentes do tamanho do bloco, o tipo de fatia e assim por diante.

7. Os valores de run-length podem ser codificados de acordo com vários métodos de binarização, tais como código unário, código Golomb-Rice ou Exponencial-Golomb, ou versões truncadas dos códigos, tendo em conta que o valor de run length não pode ser maior do que o tamanho do vetor binário.

8. A sinalização run length pode ser parada quando o número de elementos sinalizados '1' atingiu o número máximo possível, o que pode representar o tamanho máximo possível de paleta.

[0184] Por exemplo, com um vetor de predição binário b = {11001110100001], o algoritmo básico acima descrito produz: [0]0-2(0)-2-3(0)-1-1(0)-4-1(1). No início, a faixa é definida como 0, e o primeiro run-length é sinalizado para aquela faixa e ele é 0, neste exemplo, então a cada '-' a faixa é trocada par 1-faixa. O último flag sinalizado está indicado nas chaves. Nos colchetes é o zero_flag. Por exemplo, o seguinte é o resultado de um algoritmo aperfeiçoado: 1-1(0)-1-2(0)-0-0 (0)-3~~-0(1)~~.

[0185] Após o primeiro processamento de elemento refletir o valor de run length 1 devido ao item No.5, acima, os outros valores de run-length são reduzidos em 1 como mencionado no item melhorado No.3, acima. O riscado indica nenhuma sinalização do último flag para o último ‘1’ no vetor mencionado no item 2 acima, e nenhuma sinalização de valor run length para o último elemento ‘1’ sugerida no No.4, acima.

[0186] Para outro exemplo: b = {11001110100011} e um exemplo melhorado do algoritmo resulta no seguinte: 11(0)-1-2(0)-0-0(0)-3~~-1(1)~~.

[0187] Alguns métodos exemplares podem usar um código de Exponencial-Golomb para codificar o run-length da execução acima da cópia, execução à esquerda da cópia, ou

ambas na codificação de bloco de índice de modo de paleta. Por exemplo, um método que utiliza um código Exponencial-Golomb para codificar comprimento de execução acima da cópia também pode aplicar um código de Exponencial-Golomb para codificar o run-length de acima da cópia, execução de cópia da esquerda, ou ambas na codificação de bloco de índice de modo de paleta de. Em outro exemplo, após a sinalização os flags "maior que 0”, "maior que 1”, e "maior que 2", uma segunda ordem de código de Golomb-Rice, pode ser utilizada para codificar os valores restantes (por exemplo, run length - 3) se o (por exemplo, run length - 3) > = 0.

[0188] Em alguns exemplos, os flags "maior que 0", "maior que 1" e "maior que 2" podem ser utilizados para indicar que um run length é "maior que 0”, "maior que 1", ou "maior que 2", respectivamente. Um run-length pode ser codificado utilizando um vetor de predição binário run-length codificado.

[0189] De acordo com alguns exemplos adicionais desta divulgação, um codificador de vídeo (por exemplo, codificador de vídeo 20 ou decodificador de vídeo 30) pode determinar um ou mais novos preditores de paleta. De acordo com estes exemplos adicionais desta divulgação, o codificador de vídeo pode determinar um novo vetor booleano para sinalização de preditores de paleta. Estas e outras técnicas são descritas em maior detalhes, por exemplo, com respeito à figura 10 abaixo.

[0190] A figura 10 é um diagrama conceitual que ilustra um exemplo da determinação de uma paleta para a codificação de dados de vídeo, de acordo com as técnicas desta divulgação. O exemplo da figura 10 inclui um quadro 178 que possui uma primeira unidade de codificação (CU) 180 que é codificada utilizando o modo de paleta (PAL)

associada com primeiras paletas 184 e uma segunda CU 188 que está associada com segundas paletas 192. Tal como descrito em maior detalhe abaixo e de acordo com as técnicas desta divulgação, segundas paletas 192 baseiam-se em primeiras paletas 184. A figura 178 também inclui o bloco 196 codificado com um modo de codificação de intrapredição e o bloco 200 que está codificado com um modo de codificação de interpredição.

[0191] As técnicas da figura 10 são descritas no contexto de desempenho por um codificador de vídeo 20 (figura 1 e figura 2) e decodificador de vídeo 30 (figura 1 e figura 3) e no que diz respeito ao padrão de codificação de vídeo HEVC para fins de explicação. No entanto, deve ser entendido que as técnicas desta divulgação não se limitam, desta forma, e podem ser aplicadas por outros processadores e/ou dispositivos de codificação de vídeo em outros processos e/ou padrões de codificação de vídeo.

[0192] De um modo geral, uma paleta refere-se a uma série de valores de pixel que são dominantes e/ou representativos para uma CU atualmente sendo codificada, CU 188 no exemplo da figura 10. Primeiras paletas 184 e segundas paletas 192 são mostradas como incluindo várias paletas. Em alguns exemplos, de acordo com aspectos da presente divulgação, um codificador de vídeo (por exemplo, um codificador de vídeo 20 ou decodificador de vídeo 30) pode codificar paletas separadamente para cada componente de cor de uma CU. Por exemplo, o codificador de vídeo pode codificar 20 uma paleta para um componente de luma (Y) de uma CU, outra paleta para uma componente de croma (U) da CU, e ainda outra paleta para o componente de croma (V) da CU. Neste exemplo, as entradas de paleta Y podem representar valores Y de pixels da CU, entradas da paleta U

podem representar valores U de pixels da CU, e entradas da paleta V podem representar os valores V de pixels da CU.

[0193] Em outros exemplos, o codificador de vídeo 20 pode codificar uma única paleta para todos os componentes de cor de uma CU. Neste exemplo, o codificador de vídeo pode codificar 20 uma paleta com uma i-ésima entrada que é um valor triplo, incluindo Yi, Ui, e Vi. Neste caso, a paleta inclui valores para cada um dos componentes dos pixels. Por conseguinte, a representação de paletas 184 e 192 como um conjunto de paletas possuindo várias paletas individuais é meramente um exemplo e não pretende ser limitativa.

[0194] No exemplo da figura 10, as primeiras paletas 184 incluem três entradas 202-206 que têm valor de índice de entrada 1, valor de índice de entrada 2 e valor de índice de entrada 3, respectivamente. Entradas 202-206 relacionam os valores de índice a valores de pixel, incluindo valor de pixel A, valor de pixel B, e valor de pixel C, respectivamente. Tal como aqui descrito, em vez de codificar os valores de pixel reais da primeira CU 180, um codificador de vídeo (por exemplo, um codificador de vídeo 20 ou decodificador de vídeo 30) pode utilizar uma codificação baseada em paleta para codificar os pixels do bloco utilizando os índices 1-3. Isto é, para cada posição de pixel da primeira CU 180, o codificador de vídeo 20 pode codificar um valor de índice para o elemento de imagem, em que o valor do índice é associado com um valor de pixel em uma ou mais das primeiras paletas 184. O decodificador de vídeo 30 pode obter os valores de índice a partir de um fluxo de bits e reconstruir os valores de pixel usando os valores de índice e uma ou mais das primeiras paletas 184. Assim, primeiras paletas 184 são transmitidas pelo codificador de vídeo 20 em um fluxo de bits de dados de

vídeo codificado para utilização pelo decodificador de vídeo 30 na decodificação baseada em paleta.

[0195] Em alguns exemplos, codificador de vídeo 20 e decodificador de vídeo 30 podem determinar segundas paletas 192 com base em primeiras paletas 184. Por exemplo, o codificador de vídeo 20 e/ou decodificador de vídeo 30 podem localizar um ou mais blocos a partir dos quais as paletas preditivas, em neste exemplo, primeiras paletas 184, são determinadas. Em alguns exemplos, tais como o exemplo ilustrado na figura 10, codificador de vídeo 20 e/ou decodificador de vídeo 30 podem localizar a CU previamente codificada, como uma CU vizinha esquerda (primeira CU 180) ao determinar uma paleta preditiva para segunda CU 188.

[0196] No exemplo da figura 10, segundas paletas 192 incluem três entradas 208-212 tendo valor de índice de entrada 1, valor do índice de entrada 2 e valor de índice de entrada 3, respectivamente. Entradas 208-212 relacionam os valores de índice a valores de pixel, incluindo valor de pixel A, valor de pixel B, e valor de pixel D, respectivamente. Neste exemplo, o codificador de vídeo 20 pode codificar um ou mais elementos de sintaxe indicando que as entradas de primeiras paletas 184 estão incluídas nas segundas paletas 192. No exemplo da figura 10, os um ou mais elementos de sintaxe são ilustrados como vetor 216 (por exemplo, um vetor de predição binário). Vetor 216 tem um número de faixas associadas (ou bits); com cada faixa indicando se o preditor de paleta associado a essa faixa é utilizado para predizer uma entrada da paleta atual. Por exemplo, o vetor 216 indica que as duas primeiras entradas de primeiras paletas 184 (202 e 204) estão incluídas nas segundas paletas 192 (um valor de "1" no vetor 216), enquanto que a terceira entrada de primeiras paletas 184

não está incluída em segundas paletas 192 (um valor de "0" no vetor 216). No exemplo da figura 10, o vetor é um vetor booleano.

[0197] Em alguns exemplos, o codificador de vídeo 20 e o decodificador de vídeo 30 podem determinar uma lista de preditor de paleta (que também pode ser referida como uma tabela de preditor de paleta) ao realizar a predição de paleta. A lista de preditor de paleta pode incluir entradas de paletas de um ou mais blocos vizinhos que são utilizados para predizer uma ou mais entradas de uma paleta para a codificação de um bloco atual. O codificador de vídeo 20 e decodificador de vídeo 30 podem construir a lista da mesma maneira. O codificador de vídeo 20 e decodificador de vídeo 30 podem codificar dados (como vetor 216) para indicar que as entradas da lista de preditor de paleta devem ser incluídas em uma paleta para a codificação de um bloco atual.

[0198] O documento C. Gisquet, G. Laroche e P. Onno, "AhG10: Pallete predictor stuffing", JCTVC-Q0063 divulga um processo exemplar para determinar uma lista de preditor de paleta. Em alguns exemplos, como notado acima, o codificador de vídeo 20 ou decodificador de vídeo 30 podem usar um vetor booleano (tal como o vetor 216) para indicar se cada item na lista de paleta de preditor é utilizado (ou não utilizado) para predizer uma ou mais entradas na paleta para o bloco atualmente sendo codificado.

[0199] Em alguns exemplos, todos os itens da lista de preditor de paleta são derivados da paleta previamente codificada (por exemplo, a paleta codificada com o bloco previamente codificado). No entanto, tais paletas podem ser espacialmente distantes da CU atual, o que pode fazer a correlação de paleta relativamente fraca.

Em geral, expandir a tabela de preditor de paleta (ou lista) pode ser útil (por exemplo, pode prover indicadores mais precisos, que pode resultar em um ganho de eficiência). No entanto, determinar e usar uma tabela de preditor de paleta relativamente grande (ou lista) resulta em um vetor booleano relativamente mais longo.

[0200] Em vários exemplos, as técnicas descritas nesta divulgação podem incluir técnicas para várias combinações de determinar, predizer e/ou sinalizar paletas em codificação baseada em paleta. As técnicas podem incluir qualquer combinação de determinar preditores de paleta, adicionar preditores a uma lista de candidatos de preditores, podar preditores de uma lista de candidatos de preditores, codificar uma indicação do uso de preditores candidatos, ou quaisquer outras técnicas aqui descritas. Enquanto certos exemplos podem ser descritos individualmente para fins de ilustração e de clareza, esta divulgação contempla qualquer combinação das técnicas de codificação baseada em paleta aqui descritas.

[0201] De acordo com alguns aspectos desta divulgação, um codificador de vídeo (como codificador de vídeo 20 ou decodificador de vídeo 30) pode determinar um ou mais novos preditores de paleta. Por exemplo, certas técnicas desta descrição incluem a determinação de um ou mais preditores de paleta espaciais. Em um exemplo, se a CU acima da CU atual (referida como uma "CU acima" ou "CU superior" e mostrada como CU intracodificada 196 no exemplo da figura 10) utiliza a codificação baseada em paleta, a paleta da CU superior é eficaz como um preditor de paleta para a CU atual (segunda CU 188). Isto é, cada uma das entradas da CU superior pode ser incluída em uma lista de preditor de paleta (por exemplo, que pode estar associada

com um vetor que indica que as entradas são utilizadas como indicadores) para predizer uma paleta atual.

[0202] No entanto, se a CU acima faz parte de um CTU diferente, usar a paleta a partir da CU acima para a predição pode necessitar de armazenamento adicional. Por exemplo, para um bloco interpredito, o codificador de vídeo pode ter de acessar à memória (por exemplo, uma memória externa ao codificador de vídeo), para reconstituir valores de pixel, que pode resultar em latência. No entanto, o codificador de vídeo pode, em alguns exemplos, considerar as amostras reconstruídas associadas com um bloco intrapredito, por este meio denotado como amostras de referência. Tais amostras reconstituídas podem ser armazenadas localmente e disponíveis para o codificador de vídeo 20 e decodificador de vídeo 30.

[0203] Em um exemplo, de acordo com aspectos desta divulgação, as amostras de referência são adicionadas à lista de paleta de preditor. Por exemplo, os valores de pixel de um bloco espacialmente vizinho (que pode ser paleta codificada ou intracodificada) podem ser identificados como indicadores de paleta candidata e associados com um vetor (como mencionado acima). Em alguns exemplos, pode ser utilizado apenas um subconjunto das amostras de referência. Por exemplo, se a lista de preditor de paleta já inclui a paleta a partir de uma CU particular, tal como a UC vizinha esquerda (primeira CU 180), as amostras a partir daquela CU não podem ser incluídas na lista de predição de paleta. Além disso, em alguns exemplos, o codificador de vídeo pode aplicar um processo de poda para remover os preditores de paleta duplicados na lista, à custa de uma complexidade adicional para o codificador de vídeo 20 e especialmente decodificador de vídeo 30. Estes novos preditores de paleta podem ser

inseridos, quer no início da lista de preditor ou no fim da lista. Em outros exemplos, os novos indicadores de paleta (por exemplo, preditores de paleta espaciais) podem ser inseridos de forma adaptativa na lista de acordo com uma determinada regra, tal como o tamanho da lista de preditor, tamanho de CU, ou semelhantes.

[0204] Em outros exemplos, todos os outros pixels tais como pixels reconstruídos em qualquer CU em uma coluna à esquerda da CU atual podem também ser adicionados à lista de paleta de preditor.

[0205] Deste modo, o codificador de vídeo 20 e/ou decodificador de vídeo 30 podem determinar um ou mais preditores de paleta espaciais para a predição de uma ou mais entradas de uma paleta para um bloco atual de dados de vídeo, em que cada entrada de uma ou mais entradas indica um valor de pixel, e codificar um ou mais elementos de sintaxe indicando se cada respectivo preditor de paleta de um ou mais preditores de paleta são utilizados para predizer as entradas da paleta para o bloco atual de dados de vídeo.

[0206] De acordo com outros aspectos desta divulgação, um codificador de vídeo (como codificador de vídeo 20 ou decodificador de vídeo 30) pode determinar um novo vetor booleano para sinalização de preditores de paleta. Em alguns exemplos, as técnicas da presente divulgação podem ser utilizadas para codificar o vetor booleano para a predição de paleta.

[0207] Por exemplo, de acordo com aspectos da presente divulgação, o codificador de vídeo 20 ou decodificador de vídeo 30 podem determinar uma árvore binária para a predição de paleta. Em um exemplo para fins de ilustração, assume-se que o vetor booleano 0-1 que representa o uso de preditores de paleta é denotado por N.

Para eficientemente codificar este vetor, uma árvore binária é utilizada. Especificamente, vetor N é dividido em várias regiões contíguas, com cada região contendo um ou mais de um elemento de vetor. Um bit é sinalizado para indicar se todos os elementos dentro dessa região são zero ou não. Se a região não é todo zero, ela é adicionalmente dividida em sub-regiões da mesma maneira.

[0208] O particionamento pode ser interrompido de acordo com as regras. Por exemplo, uma regra pode ser "se o tamanho da região é menor do que um limite, o processo de partição para". Quando o particionamento para, se o tamanho da região é maior do que um, e ele contém elementos diferentes de zero, cada elemento na região é sinalizado no fluxo de bits. Neste caso, partindo do princípio de que o tamanho da região é X, se os primeiros elementos X-1 são todos zero, o último elemento deve ser um, de modo que não podem ser incluídos no fluxo de bits.

[0209] Em um outro exemplo, quando o processo de particionamento para e o tamanho da região é maior do que um, mas menor do que um segundo limite, o codificador de vídeo pode ignorar a sinalização de um bit para indicar se todos os elementos naquela região são zero. Em vez disso, para cada elemento na região, o codificador de vídeo pode sinalizar um bit para indicar se o respectivo elemento é zero ou um.

[0210] Em um outro exemplo, em vez de uma árvore binária, pode ser utilizada uma árvore com mais de duas ramificações em cada nó. Além disso, em alguns exemplos, a codificação dos primeiros vários níveis a partir da raiz da árvore pode ser ignorada. Em tais exemplos, os valores podem ser implicitamente assumidos como sendo um.

[0211] De acordo com aspectos da presente divulgação, a posição da última entrada diferente de zero

da lista de paleta de preditor (isto é, o último 1 no vetor booleano) pode ser explicitamente codificada no fluxo de bits, antes de sinalizar se as entradas individuais são 0 ou 1. Neste caso, os elementos a partir da última posição (inclusive) até o fim do vetor booleano não são sinalizados no fluxo de bits. Em outro exemplo, a posição do primeiro preditor de paleta que é utilizado para a predição, ou seja, o primeiro 1 no vetor booleano, pode também ser sinalizado explicitamente no fluxo de bits.

[0212] De acordo com aspectos da presente divulgação, o número de entradas que são utilizadas a partir da lista de preditor de paleta pode ser explicitamente codificado no fluxo de bits. Por exemplo, as técnicas da presente invenção incluem sinalizar aquele número de uns ("é") no vetor Booleano discutido acima. Em alguns exemplos, codificador de vídeo 20 pode inicialmente sinalizar o número de "1s" {TotalOnes) em um vetor booleano. O codificador de vídeo 20 pode, em seguida, sequencialmente, sinalizar (ou sinalizar de alguma outra maneira) cada item no vetor booleano (0 ou 1) até que o número de 1s sinalizados seja igual a TotalOnes.

[0213] Do mesmo modo, decodificador de vídeo 30 pode primeiro decodificar uma indicação do número de uns que estão incluídos em um vetor booleano recebido. Decodificador de vídeo 30 pode então decodificar cada elemento do vetor recebido até que o decodificador de vídeo 30 tenha decodificado o número de "1s" no vetor correspondente ao número de "1s" indicado. Em alguns exemplos, sinalização item a item (por exemplo, para os elementos do vetor) pode ser sequencial a partir do início do vetor. Em outros exemplos, os elementos do vetor podem ser sinalizados de acordo com alguma outra ordem de

leitura. Em ainda outros exemplos, a sinalização item a item pode usar tecnologia de codificação run-length.

[0214] Em alguns exemplos, quando o número de uns no vetor booleano é inicialmente sinalizado (por exemplo, TotalOnes é sinalizado no fluxo de bits, antes do vetor booleano), cada elemento de sintaxe (por exemplo, o flag) no vetor booleano pode ser codificado por contexto utilizando CABAC. Em alguns casos, podem ser utilizados vários contextos. Em alguns exemplos, todos os elementos de sintaxe de vetor booleano podem ser codificados por contexto. Em outros exemplos, apenas um subconjunto dos elementos de sintaxe de vetor booleano pode ser codificado por contexto.

[0215] De acordo com aspectos da presente divulgação, o contexto para um elemento de sintaxe do vetor booleano pode ser dependente da posição do elemento de sintaxe (por exemplo, um flag binário) dentro do vetor booleano e o número sinalizado de uns {TotalOnes) que é inicialmente sinalizado. Em um exemplo para fins de ilustração, se ((posição>3) && ((posição/2+1)>TotalOnes)) um codificador de vídeo (por exemplo, um codificador de vídeo 20 ou decodificador de vídeo 30) pode utilizar um primeiro contexto; de outra forma (quando a condição não é verdadeira) o codificador de vídeo pode usar outro contexto, em que a posição indica uma posição relativa do elemento de sintaxe sendo codificado no vetor booleano. Neste caso, o codificador de vídeo utiliza dois contextos para codificar os elementos de sintaxe (por exemplo, flags binários) no vetor booleano, e seleciona um contexto para um elemento de sintaxe particular do vetor booleano com base na posição relativa do elemento de sintaxe particular no vetor booleano. Em outro exemplo, o codificador de vídeo pode usar um contexto se uma condição ((posição>3) &&

((posição/4+1)>=TotalOnes)) é verdadeira e outro contexto, se a condição não é verdadeira. Podem também ser utilizadas outras condições para a determinação de contextos.

[0216] Nos exemplos acima, o contexto CABAC para um elemento de sintaxe particular de um vetor booleano depende da posição relativa do elemento de sintaxe e do número total de uns no vetor booleano {TotalOnes). Em outros exemplos, em vez de dependência sobre a posição e TotalOnes, o codificador de vídeo pode determinar contexto para elementos de sintaxe com base em uma posição relativa do elemento de sintaxe a ser codificada e o número total uns que foram sinalizados a medida que (por exemplo, o número total de uns que ocorreram antes da sintaxe de elemento a ser codificado no vetor booleano). Em ainda outros exemplos, um codificador de vídeo pode utilizar qualquer combinação das condições acima descritas para determinar contexto para elementos de sintaxe de codificação CABAC de um vetor booleano.

[0217] De acordo com aspectos da presente divulgação, um tamanho de paleta é sinalizado por um codificador e recebido por um decodificador. Por exemplo, uma série de entradas de uma paleta pode ser sinalizada para indicar o tamanho da paleta. Em alguns exemplos, o codificador de vídeo 20 pode, inicialmente, sinalizar o tamanho da paleta (PLTSize). O codificador de vídeo 20 pode, em seguida, sinalizar cada item no vetor booleano (0 ou 1) até que o número 1s sinalizados seja igual a PLTSize. O decodificador de vídeo 30 pode receber uma indicação do tamanho da paleta, e pode continuar a adicionar à paleta (por exemplo, usando preditores de paleta identificados por um vetor) até que o número de entradas na paleta alcance o tamanho de paleta sinalizada.

[0218] Em alguns exemplos, codificador de vídeo 20 pode inicialmente sinalizar um tamanho de paleta (PLTSize). O codificador de vídeo 20 pode, em seguida, sinalizar o número de itens de paleta não predita (nonPredPLTSize). O codificador de vídeo 20 pode, em seguida, sinalizar cada item no vetor booleano (0 ou 1) até que o número de "1s" sinalizados seja igual a uma diferença entre o tamanho de paleta e o número de itens de paleta não predita (PLTSize- nonPredPLTSize). Em alguns exemplos, decodificador de vídeo 30 pode receber uma indicação do tamanho da paleta e o número de entradas na paleta que não são preditas usando um preditor de paleta (por exemplo, o número de "0s" de um vetor). O decodificador de vídeo 30 pode também receber um vetor indicando qual paleta de preditores deve ser incluída em uma paleta (por exemplo, tal como o vetor booleano descrito acima). Decodificador de vídeo 30 pode então decodificar cada item do vetor até que o número de "1s" seja igual a uma diferença entre o tamanho de paleta e o número de itens de paleta não predita (PLTSize- nonPredPLTSize).

[0219] Deste modo, o codificador de vídeo 20 e/ou decodificador de vídeo 30 pode, em um exemplo, determinar um ou mais preditores de paleta para predizer uma ou mais entradas de uma paleta para um bloco atual de dados de vídeo, em que cada entrada das uma ou mais entradas indica um valor de pixel, determinar um vetor tendo uma pluralidade de faixas, em que cada faixa do vetor indica se um respectivo preditor de paleta dos um ou mais preditores de paleta é utilizado para predizer uma entrada de paleta para o bloco atual de dados vídeo, particionar o vetor em uma ou mais regiões, e determinar um ou mais valores das faixas do vetor com base no particionamento.

[0220] Embora as técnicas da figura 10 sejam descritas acima elas são descritas no contexto de CUs (HEVC), deve entender-se que as técnicas podem também ser aplicadas às unidades de predição (PUs) ou em outros processos e/ou padrões de codificação de vídeo.

[0221] Deve ser reconhecido que, dependendo do exemplo, certos atos ou eventos de qualquer uma das técnicas aqui descritas podem ser realizados em uma sequência diferente, podem ser adicionados, fundidos, ou deixados de fora (por exemplo, nem todos atos ou eventos descritos são necessários para a prática das técnicas). Além disso, em certos exemplos, atos ou eventos podem ser realizados simultaneamente, por exemplo, através do processamento multissequenciado, processamento de interrupção, ou múltiplos processadores, em vez de sequencialmente. Além disso, enquanto certos aspectos da presente divulgação são descritos como sendo realizados por um único módulo ou unidade para fins de clareza, deve-se compreender que as técnicas desta divulgação podem ser realizadas por uma combinação de unidades ou módulos associados com um codificador de vídeo.

[0222] Certos aspectos desta divulgação foram descritos no que diz respeito ao padrão HEVC desenvolvimento para fins de ilustração. No entanto, as técnicas descritas nesta divulgação podem ser úteis para outros processos de codificação de vídeo, incluindo outros processos de codificação de vídeo padrão ou proprietários ainda não desenvolvidos.

[0223] As técnicas descritas acima podem ser realizadas por um codificador de vídeo 20 (figuras 1 e 2) e/ou decodificador de vídeo 30 (figuras 1 e 3), ambos os quais podem ser geralmente referidos como um codificador de vídeo. Do mesmo modo, a codificação de vídeo pode referir-

se a codificação de vídeo ou decodificação de vídeo, como aplicável.

[0224] As técnicas descritas nesta divulgação podem incluir técnicas para várias combinações de um ou mais aspectos diferentes de codificação de paleta.

[0225] Em um ou mais exemplos, as funções descritas podem ser implementadas em hardware, software, firmware, ou qualquer combinação dos mesmos. Se implementadas em software, as funções podem ser armazenadas ou transmitidas, como uma ou mais instruções de código, ou um meio legível por computador e executadas por uma unidade de processamento com base em hardware. Meios legíveis por computador podem incluir meios de armazenamento legíveis por computador, o que correspondem a um meio tangível, tal como mídia de armazenamento de dados ou mídia de comunicação, incluindo qualquer meio que facilite a transferência de um programa de computador de um lugar para outro, por exemplo, de acordo com um protocolo de comunicação. Deste modo, meios legíveis por computador podem geralmente corresponder a (1) um meio de armazenamento legível por computador tangível que é não transitório ou (2) um meio de comunicação, tal como um sinal ou onda de portadora. Mídia de armazenamento de dados pode ser qualquer material disponível que pode ser acessado por um ou mais computadores ou um ou mais processadores para recuperar instruções de código, e/ou estruturas de dados para a implementação das técnicas descritas nesta divulgação. Um produto de programa de computador pode incluir um meio legível por computador.

[0226] A título de exemplo, e não como limitação, tais meios de armazenamento legíveis por computador podem compreender RAM, ROM, EEPROM, CD-ROM ou outro armazenamento em disco óptico, armazenamento em disco magnético, ou

outros dispositivos de armazenamento magnéticos, memória flash, ou qualquer outro meio que pode ser usado para armazenar o código de programa desejado sob a forma de instruções ou estruturas de dados, e que pode ser acessado por um computador. Além disso, qualquer ligação é denominada adequadamente um meio legível por computador. Por exemplo, se as instruções são transmitidas de um site, servidor ou outra fonte remota utilizando um cabo coaxial, cabo de fibra óptica, par trançado, linha de assinante digital (DSL) ou tecnologias sem fio, tais como infravermelhos, rádio e micro-ondas, então o cabo coaxial, cabo de fibra óptica, par trançado, DSL, ou tecnologias sem fio, tais como infravermelho, rádio e micro-ondas estão incluídos na definição de meio. Deve ser entendido, no entanto, que mídia de armazenamento legível por computador e mídia de armazenamento de dados não incluem conexões, ondas de portadoras, sinais ou outra mídia transitória, mas são direcionados para mídia de armazenamento tangível não-transitória. Disco e disquete, como aqui utilizado, incluem disco compacto (CD), disco laser, disco óptico, disco versátil digital (DVD), disquete e disco Blu-ray, onde discos geralmente reproduzem dados magneticamente, enquanto que discos reproduzem dados opticamente com lasers. Combinações dos anteriores também devem ser incluídas dentro do âmbito dos meios legíveis por computador.

[0227] As instruções podem ser executadas por um ou mais processadores, tais como um ou mais processadores de sinal digital (DSPs), microprocessadores de uso geral, circuitos integrados de aplicação específica (ASIC), arranjos de porta programáveis em campo (FPGA), ou outra equivalente integrado ou conjunto de circuitos lógicos discretos. Por conseguinte, o termo "processador" tal como aqui utilizado pode referir-se a qualquer da estrutura

precedente ou qualquer outra estrutura adequada para aplicação das técnicas aqui descritas. Além disso, em alguns aspectos, a funcionalidade aqui descrita pode ser provida dentro de módulos de hardware e/ou software dedicado configurados para a codificação e decodificação, ou incorporada em um codec combinado. Além disso, as técnicas podem ser totalmente implementadas em um ou mais circuitos ou elementos lógicos.

[0228] As técnicas da presente divulgação podem ser implementadas em uma vasta variedade de dispositivos ou aparelhos, incluindo um aparelho telefônico sem fio, um circuito integrado (IC) ou um conjunto de ICs (por exemplo, um conjunto de chips). Vários componentes, módulos ou unidades são descritos nesta divulgação para enfatizar os aspectos funcionais de dispositivos configurados para executar as técnicas divulgadas, mas não precisam necessariamente de realização por diferentes unidades de hardware. Em vez disso, tal como descrito acima, várias unidades podem ser combinadas de uma unidade de hardware codec ou providas por um conjunto de unidades de hardware interoperativo, incluindo um ou mais processadores, conforme descrito acima, em conjunto com o software e/ou firmware adequado.

REIVINDICAÇÕES

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

receber (900) um vetor de predição binário codificado para um bloco atual de dados de vídeo;

decodificar (902) o vetor de predição binário codificado utilizando uma técnica de decodificação run length, em que a técnica de decodificação run length compreende:

decodificar o vetor de predição binário codificado para obter valores run length decodificados;

para cada valor run length decodificado:

se o valor run length decodificado for maior do que um valor run length reservado L, subtrair 1 do valor run length decodificado para gerar um valor run length real que especifica um número de zeros consecutivos que precedem um valor não zero,

se o valor run length decodificado for menor do que o valor run length reservado L, usar o valor run length decodificado como o valor run length real, ou

se o valor run length decodificado for igual ao valor run length reservado L, determinar que nenhum valor não zero adicional está presente no vetor de predição binário;

gerar (904) uma paleta para o bloco atual de dados de vídeo com base no vetor de predição binário, o vetor de predição binário compreendendo entradas indicando se ou não entradas de paleta utilizadas previamente são reutilizadas para a paleta para o bloco atual de dados de vídeo; e

decodificar (906) o bloco atual de dados de vídeo utilizando a paleta.

2. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que a geração da paleta para o

bloco atual de dados de vídeo com base no vetor de predição binário compreende:

copiar, de um armazenador, entradas de paleta previamente utilizadas na paleta que são indicadas como sendo reutilizadas para a paleta pelo vetor de predição binário; e

receber, no caso de o número de entradas de paleta utilizado previamente copiadas para a paleta ser inferior a um tamanho máximo de paleta, entradas de paleta adicionais.

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

gerar (800) uma paleta para o bloco atual de dados de vídeo;

gerar (802) um vetor de predição binário para a paleta para o bloco atual de dados de vídeo, o vetor de predição binário compreendendo entradas indicando se ou não entradas de paleta utilizadas previamente são reutilizadas para a paleta para o bloco atual de dados de vídeo;

codificar (804) o vetor de predição binário utilizando uma técnica de codificação run length, em que a técnica de codificação run length compreende:

para cada valor não zero no vetor de predição binário:

se um número de zeros consecutivos precedendo o valor não zero for igual ou maior a um valor run length reservado L, adicionar 1 ao número de zeros consecutivos, e codificar o resultado como um valor run length, ou

se o número de zeros consecutivos precedendo o valor não zero for menor do que o valor run length reservado L, codificar o número de zeros consecutivos como o valor run length; e

para um número de zeros consecutivos que sucedem um valor não zero final no vetor de predição

binário, codificar o valor run length reservado L como o valor run length; e

codificar (806) o bloco atual de dados de vídeo utilizando a paleta.

4. Método, de acordo com a reivindicação 1 ou 3, **caracterizado** pelo fato de que o valor run length reservado L é igual a 1.

5. Método, de acordo com a reivindicação 2 ou reivindicação 3, **caracterizado** pelo fato de que compreende adicionalmente:

armazenar as entradas de paleta utilizadas previamente para um ou mais blocos de dados de vídeo codificados previamente no armazenador.

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

remover as entradas duplicadas das entradas de paleta utilizadas previamente armazenadas no armazenador.

7. Método, de acordo com a reivindicação 5, **caracterizado** pelo fato de que as entradas de paleta utilizadas previamente para os um ou mais blocos codificados previamente de dados de vídeo compreendem as entradas de paleta utilizadas previamente para uma linha de pixels acima do bloco atual de dados de vídeo e entradas de paleta utilizadas previamente para uma linha de pixels para a esquerda do bloco atual de dados de vídeo.

8. Aparelho configurado para decodificar dados de vídeo, **caracterizado** pelo fato de que compreende:

meios para receber um vetor de predição binário codificado para um bloco atual de dados de vídeo;

meios para decodificar o vetor de predição binário codificado utilizando uma técnica de decodificação run length, em que os meios para decodificar o vetor de

predição binário codificado compreendem meios para decodificar o vetor de predição binário codificado para obter valores run length decodificados tal que, para cada valor run lenth decodificado:

se o valor run length decodificado for maior do que um valor run length reservado L, subtrair 1 do valor run length decodificado para gerar um valor run length real que especifica um número de zeros consecutivos que precedem um valor não zero,

se o valor run length decodificado for menor do que o valor run length reservado L, usar o valor run length decodificado como o valor run length real, ou

se o valor run length decodificado for igual ao valor run length reservado L, determinar que nenhum valor não zero adicional está presente no vetor de predição binário;

meios para gerar uma paleta para o bloco atual de dados de vídeo com base no vetor de predição binário, o vetor de predição binário compreendendo entradas indicando se ou não entradas de paleta utilizadas previamente são reutilizadas para a paleta para o bloco de dados de vídeo atual, e

meios para decodificar o bloco atual de dados de vídeo utilizando a paleta.

9. Aparelho configurado para codificar dados de vídeo, **caracterizado** pelo fato de compreender:

meios para gerar uma paleta para o bloco atual de dados de vídeo;

meios para gerar um vetor de predição binário para a paleta para o bloco de dados de vídeo atual, o vetor de predição binário compreendendo entradas indicando se ou não entradas de paleta previamente utilizadas são reutilizadas para a paleta para o bloco atual de dados de vídeo;

meios para codificar o vetor de predição binário utilizando uma técnica de codificação run length, em que a técnica de codificação run length compreende:

para cada valor não zero no vetor de predição binário:

se um número de zeros consecutivos precedendo o valor não zero for igual ou maior a um valor run length reservado L, adicionar 1 ao número de zeros consecutivos, e codificar o resultado como um valor run length, ou

se o número de zeros consecutivos precedendo o valor não zero for menor do que o valor run length reservado L, codificar o número de zeros consecutivos como o valor run length; e

para um número de zeros consecutivos que sucedem um valor não zero final no vetor de predição binário, codificar o valor run length reservado L como o valor run length; e

meios para codificar o bloco atual de dados de vídeo utilizando a paleta.

10. Memória legível por computador, **caracterizada** pelo fato de armazenar instruções que, quando executadas, fazem com que um ou mais processadores de um dispositivo configurado para decodificar dados de vídeo realize o método conforme definido em qualquer uma das reivindicações 1 a 7.

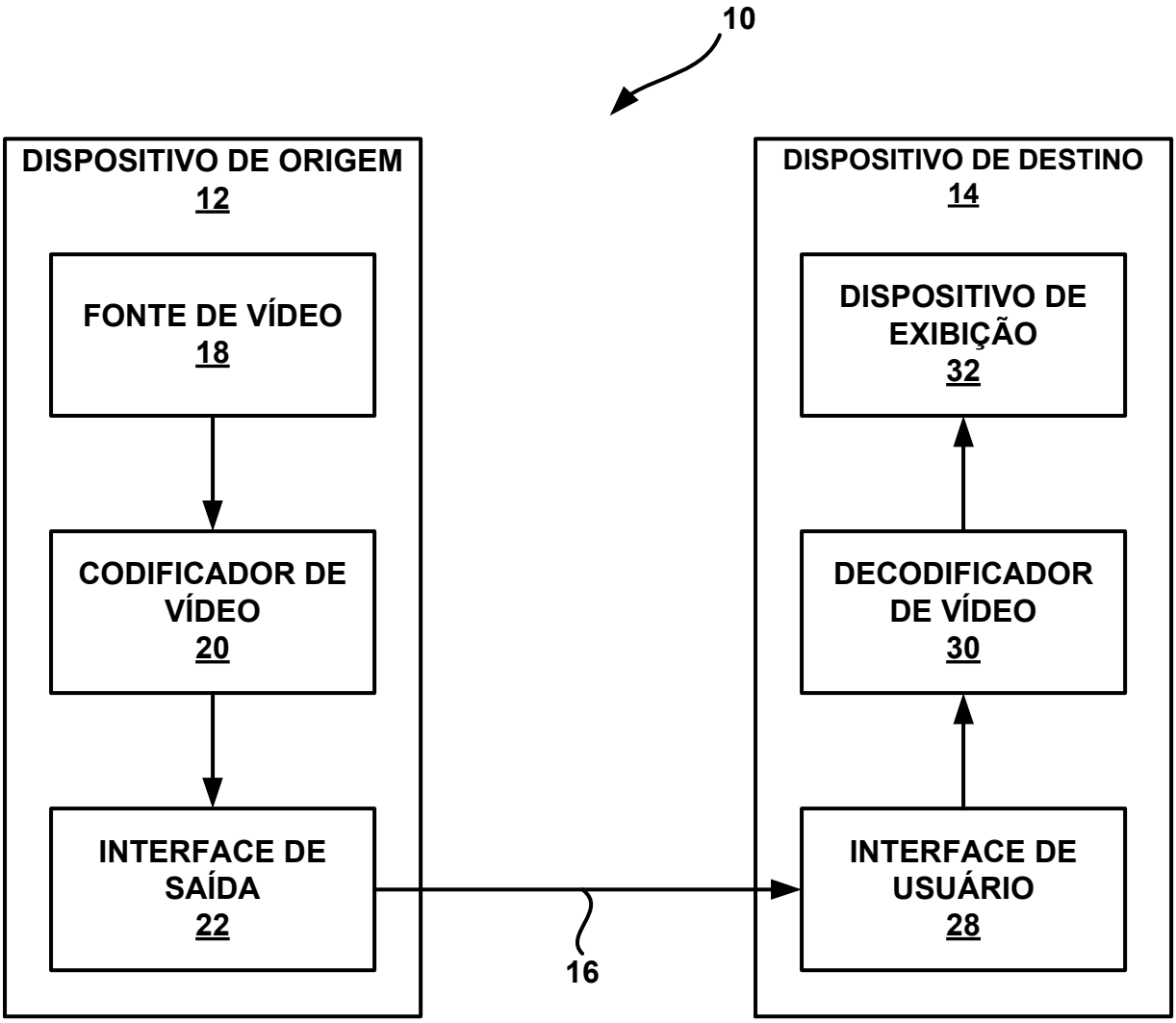
10
DISPOSITIVO DE ORIGEM
12
FONTE DE VÍDEO
18
CODIFICADOR DE VÍDEO
20
INTERFACE DE SAÍDA
22
16
DISPOSITIVO DE DESTINO
14
DISPOSITIVO DE EXIBIÇÃO
32
DECODIFICADOR DE VÍDEO
30
INTERFACE DE USUÁRIO
28

FIG. 1

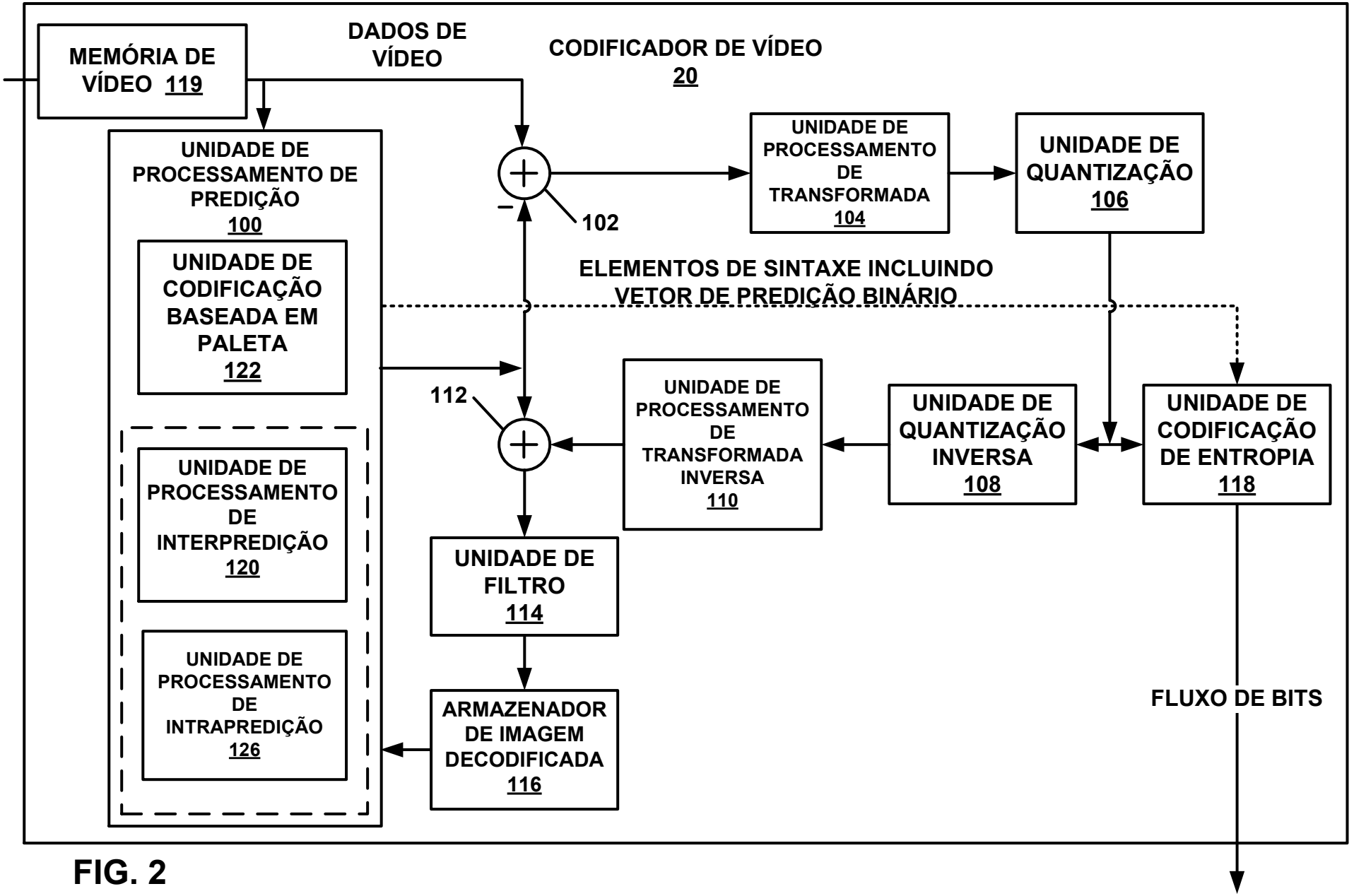
MEMÓRIA DE VÍDEO 119
DADOS DE VÍDEO
CODIFICADOR DE VÍDEO 20
UNIDADE DE PROCESSAMENTO DE PREDIÇÃO 100
UNIDADE DE CODIFICAÇÃO BASEADA EM PALETA 122
UNIDADE DE PROCESSAMENTO DE INTERPREDIÇÃO 120
UNIDADE DE PROCESSAMENTO DE INTRAPREDIÇÃO 126
102
UNIDADE DE PROCESSAMENTO DE TRANSFORMADA 104
UNIDADE DE QUANTIZAÇÃO 106
ELEMENTOS DE SINTAXE INCLUINDO VETOR DE PREDIÇÃO BINÁRIO
112
UNIDADE DE PROCESSAMENTO DE TRANSFORMADA INVERSA 110
UNIDADE DE QUANTIZAÇÃO INVERSA 108
UNIDADE DE CODIFICAÇÃO DE ENTROPIA 118
UNIDADE DE FILTRO 114
ARMAZENADOR DE IMAGEM DECODIFICADA 116
FLUXO DE BITS

FIG. 2

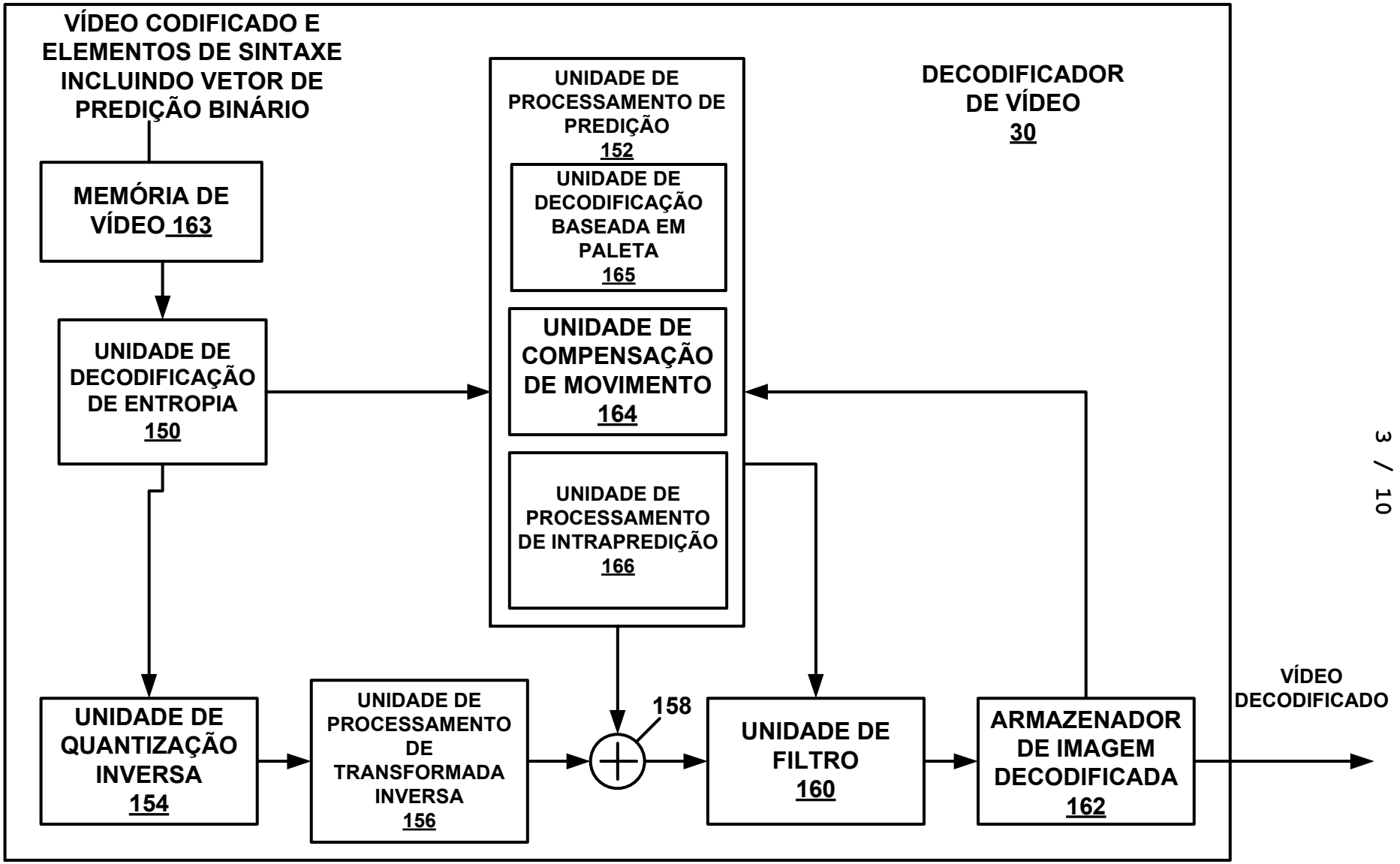
VÍDEO CODIFICADO E ELEMENTOS DE SINTAXE INCLUINDO VETOR DE PREDIÇÃO BINÁRIO
MEMÓRIA DE VÍDEO 163
UNIDADE DE DECODIFICAÇÃO DE ENTROPIA 150
UNIDADE DE QUANTIZAÇÃO INVERSA 154
UNIDADE DE PROCESSAMENTO DE TRANSFORMADA INVERSA 156
UNIDADE DE PROCESSAMENTO DE PREDIÇÃO 152
UNIDADE DE DECODIFICAÇÃO BASEADA EM PALETA 165
UNIDADE DE COMPENSAÇÃO DE MOVIMENTO 164
UNIDADE DE PROCESSAMENTO DE INTRAPREDIÇÃO 166
158
UNIDADE DE FILTRO 160
ARMAZENADOR DE IMAGEM DECODIFICADA 162
DECODIFICADOR DE VÍDEO 30
VÍDEO DECODIFICADO

FIG. 3

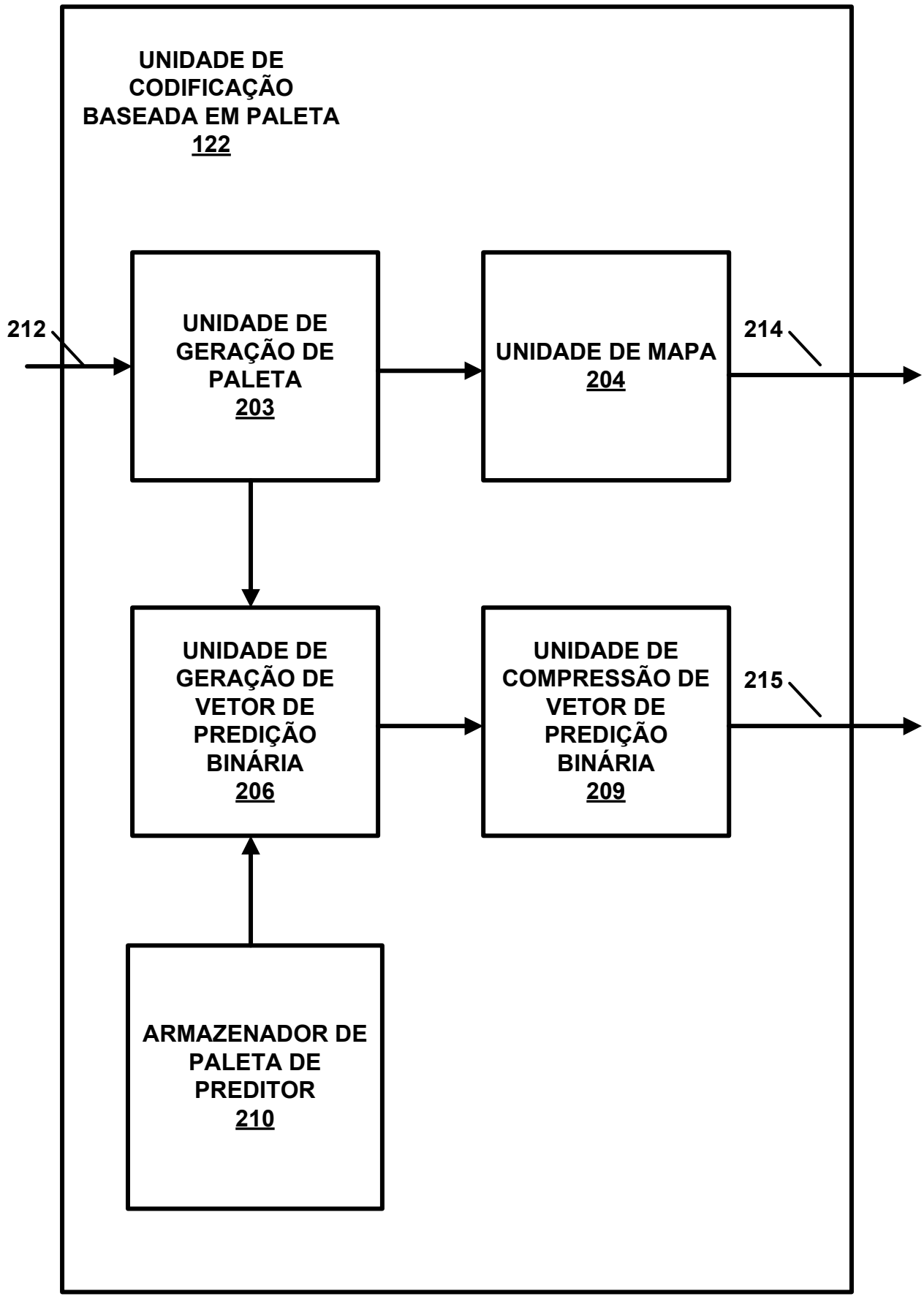
UNIDADE DE CODIFICAÇÃO BASEADA EM PALETA
122
212
UNIDADE DE GERAÇÃO DE PALETA
203
UNIDADE DE MAPA
204
214
UNIDADE DE GERAÇÃO DE VETOR DE PREDIÇÃO BINÁRIA
206
UNIDADE DE COMPRESSÃO DE VETOR DE PREDIÇÃO BINÁRIA
209
215
ARMAZENADOR DE PALETA DE PREDITOR
210

FIG. 4

ARMAZENADOR DE PALETA DE PREDIÇÃO
210

ÍNDICE DE ENTRADA	VALOR DE PIXEL
1	VALOR A
2	VALOR B
3	VALOR C
4	VALOR D
5	VALOR E
6	VALOR F
7	VALOR G
8	VALOR H
9	VALOR I
10	VALOR J
11	VALOR K
12	VALOR L
13	VALOR M
...	...

PALETA ATUAL
220

ÍNDICE DE ENTRADA	VALOR DE PIXEL
1	VALOR A
2	VALOR B
3	VALOR E
4	VALOR I
5	NOVO VALOR
6	NOVO VALOR
7	NOVO VALOR
8	NOVO VALOR

b = [1100100010000]

FIG. 5

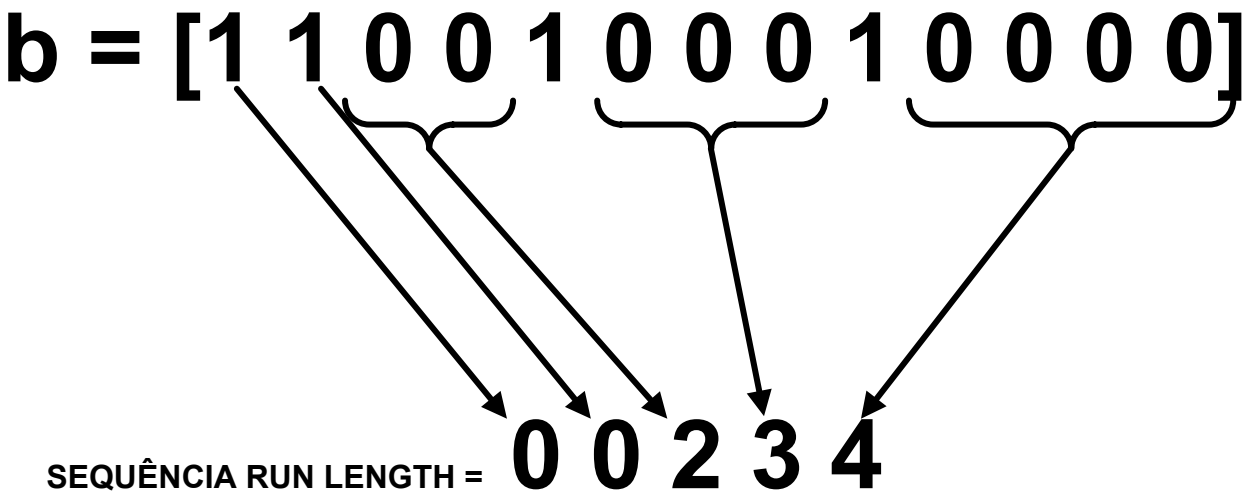

FIG. 6

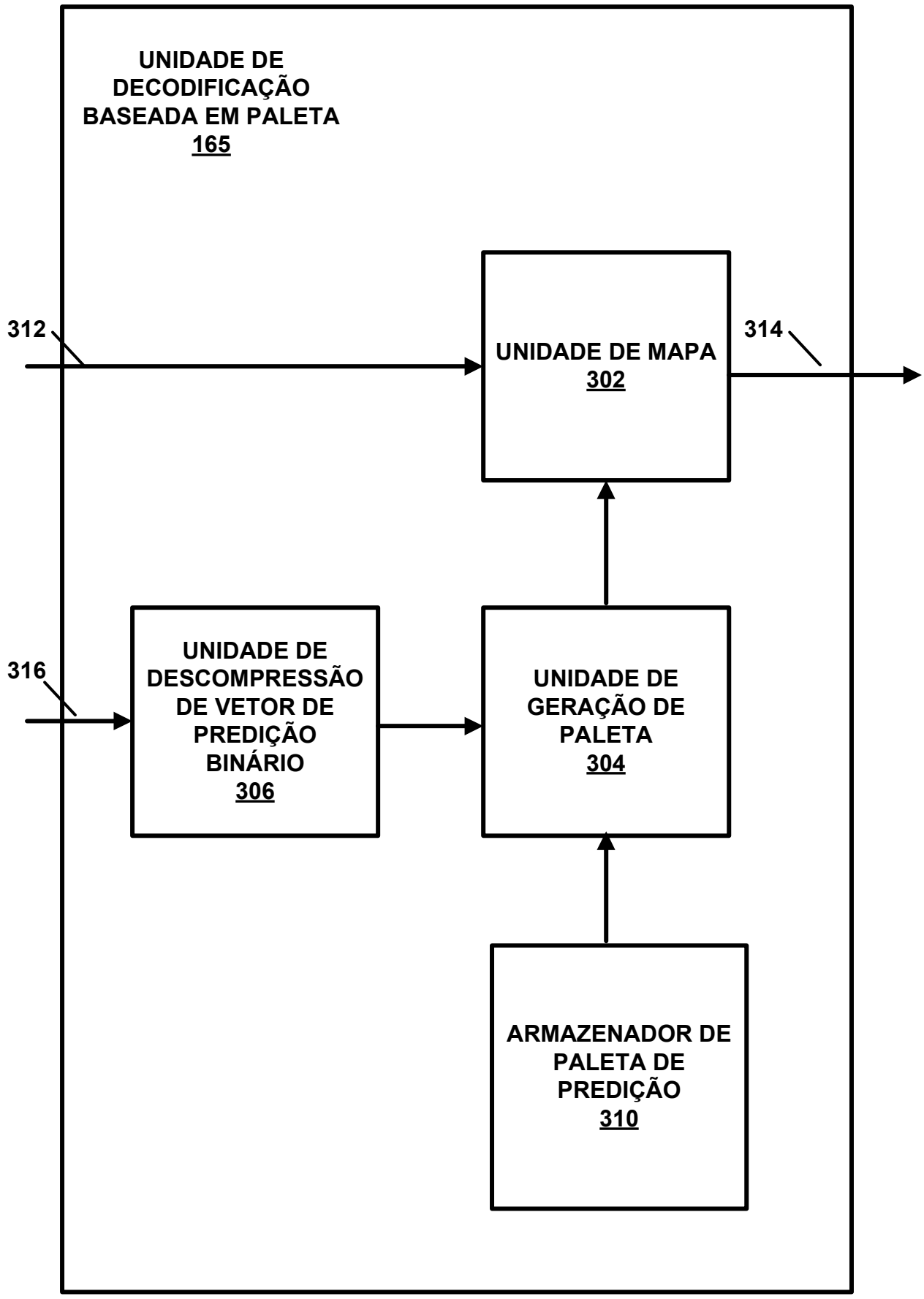
UNIDADE DE DECODIFICAÇÃO BASEADA EM PALETA
165
312
UNIDADE DE MAPA
302
314
316
UNIDADE DE DESCOMPRESSÃO DE VETOR DE PREDIÇÃO BINÁRIO
306
UNIDADE DE GERAÇÃO DE PALETA
304
ARMAZENADOR DE PALETA DE PREDIÇÃO
310

FIG. 7

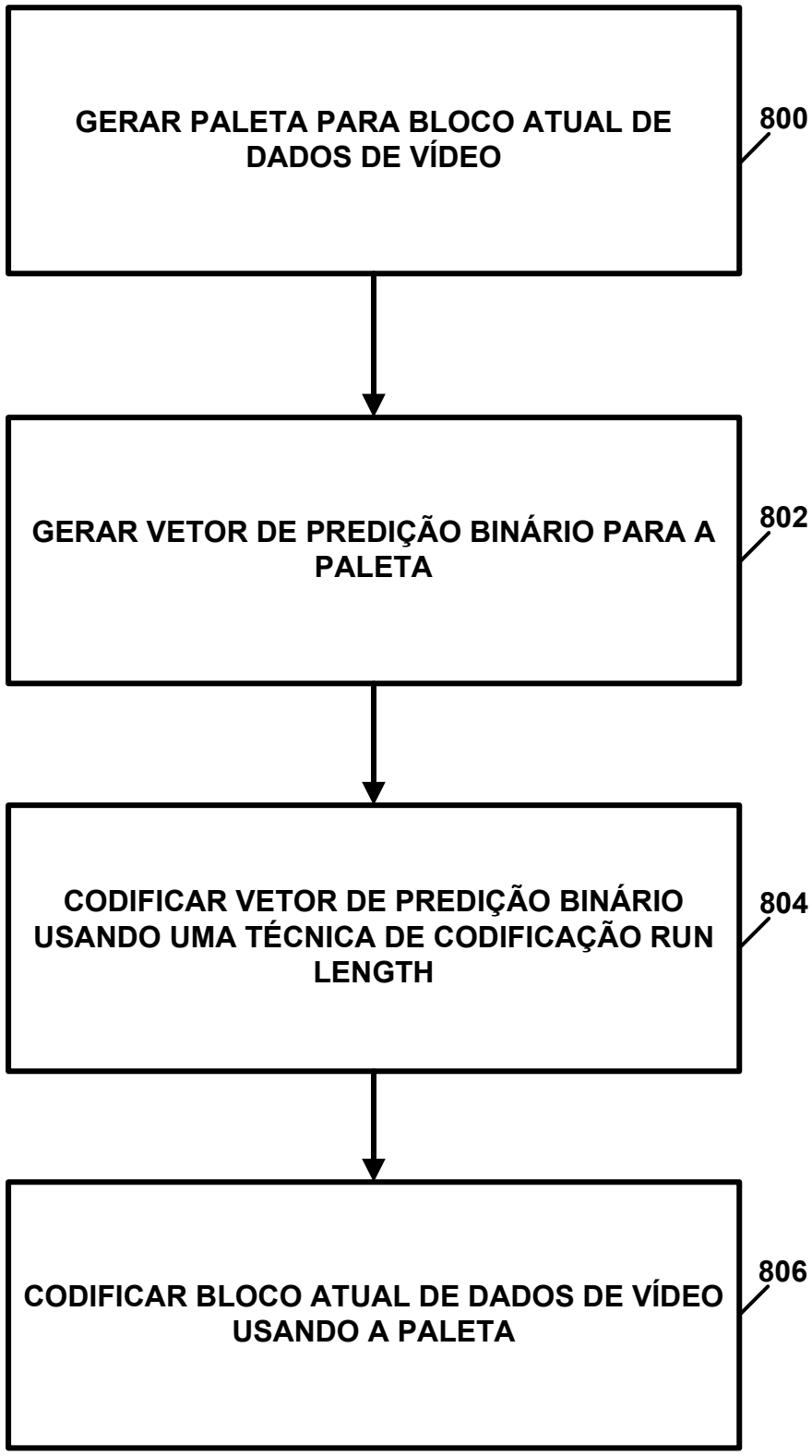
GERAR PALETA PARA BLOCO ATUAL DE DADOS DE VÍDEO
800
GERAR VETOR DE PREDIÇÃO BINÁRIO PARA A PALETA
802
CODIFICAR VETOR DE PREDIÇÃO BINÁRIO USANDO UMA TÉCNICA DE CODIFICAÇÃO RUN LENGTH
804
CODIFICAR BLOCO ATUAL DE DADOS DE VÍDEO USANDO A PALETA
806

FIG. 8

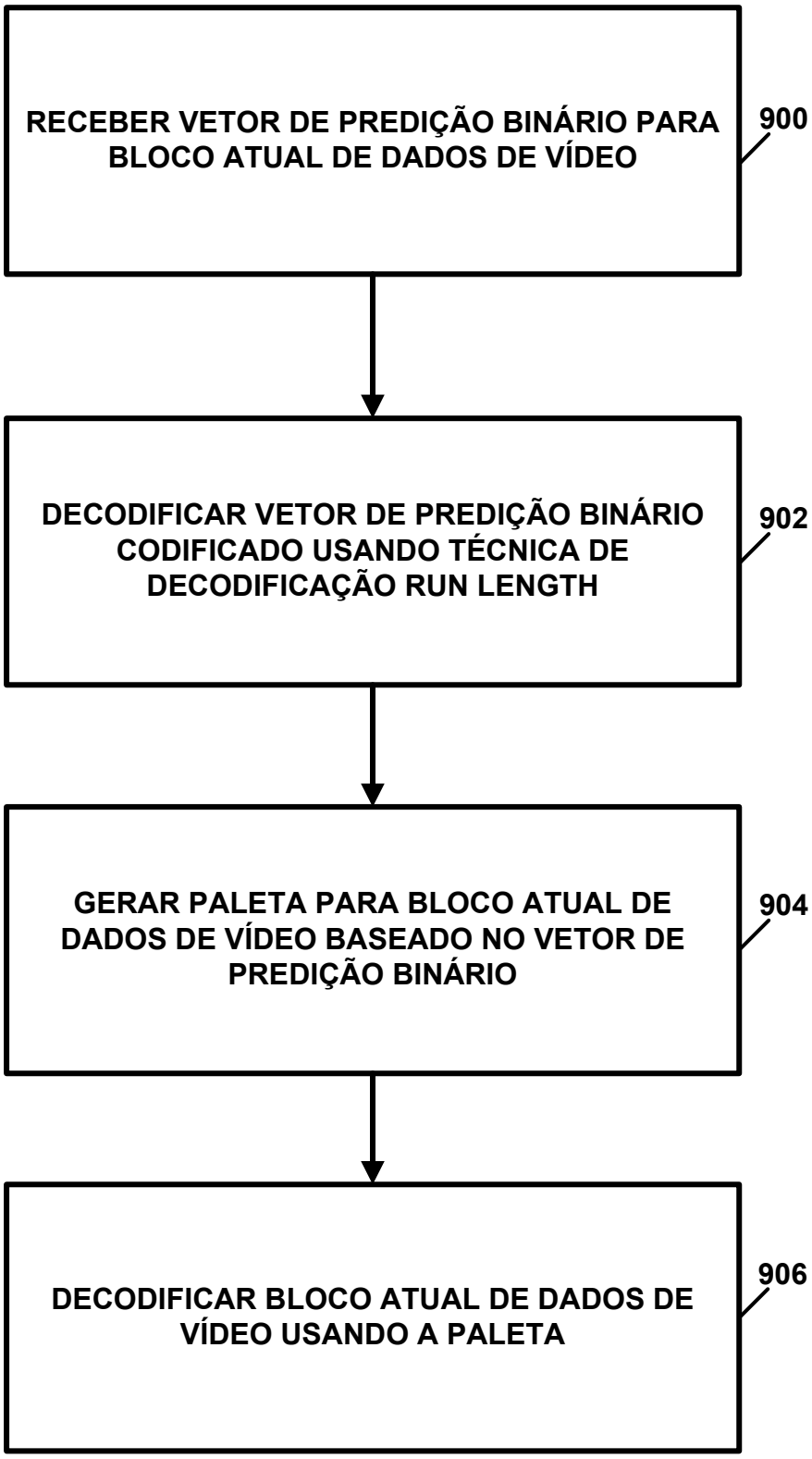
RECEBER VETOR DE PREDIÇÃO BINÁRIO PARA BLOCO ATUAL DE DADOS DE VÍDEO
900
DECODIFICAR VETOR DE PREDIÇÃO BINÁRIO CODIFICADO USANDO TÉCNICA DE DECODIFICAÇÃO RUN LENGTH
902
GERAR PALETA PARA BLOCO ATUAL DE DADOS DE VÍDEO BASEADO NO VETOR DE PREDIÇÃO BINÁRIO
904
DECODIFICAR BLOCO ATUAL DE DADOS DE VÍDEO USANDO A PALETA
906

FIG. 9

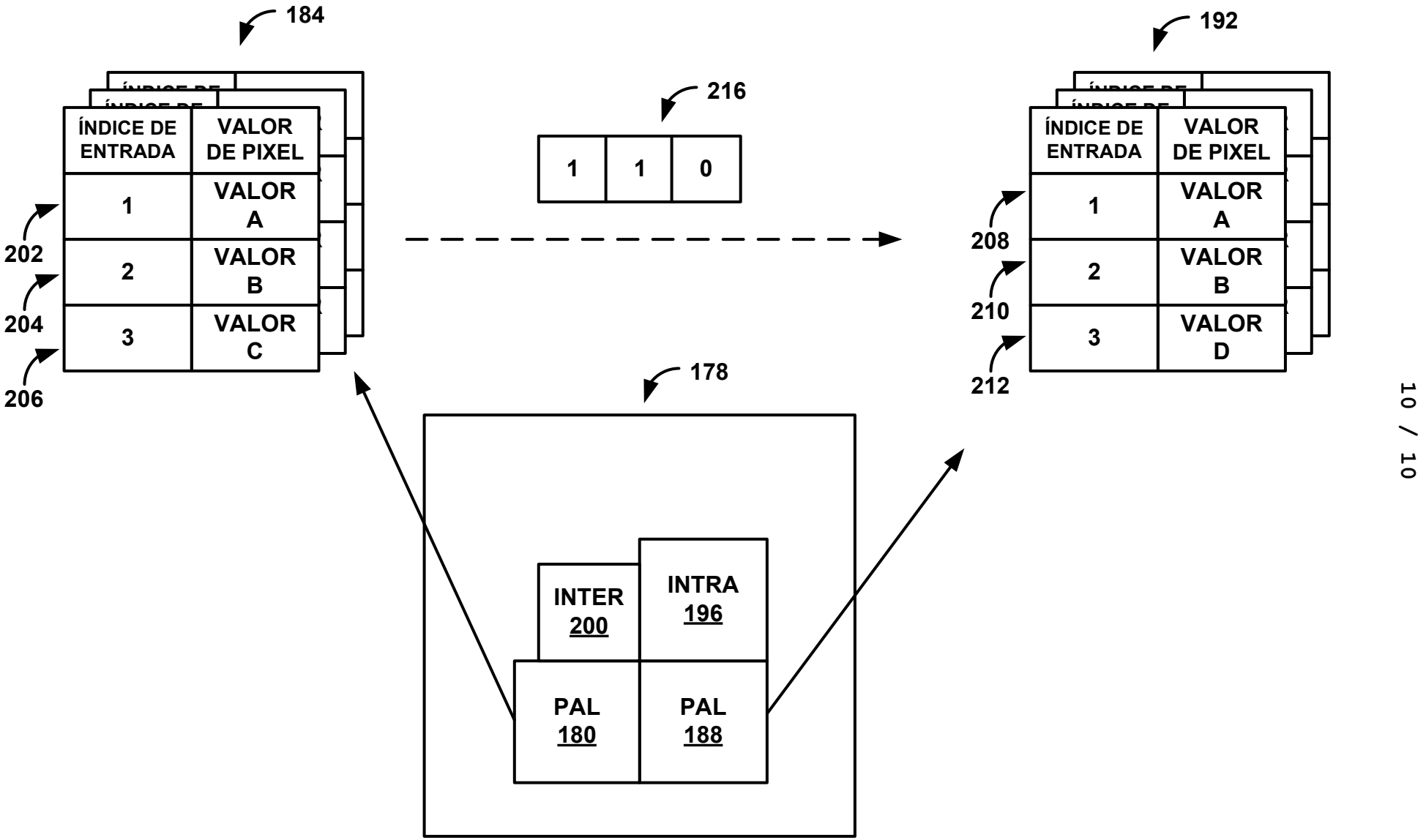
184
ÍNDICE DE ENTRADA
VALOR DE PIXEL
1
VALOR A
2
VALOR B
3
VALOR C
202
204
206
216
1
1
0
192
ÍNDICE DE ENTRADA
VALOR DE PIXEL
1
VALOR A
2
VALOR B
3
VALOR D
208
210
212
178
INTER 200
INTRA 196
PAL 180
PAL 188

FIG. 10