



República Federativa do Brasil
Ministério da Economia
Instituto Nacional da Propriedade Industrial

(11) BR 122014018140-0 B1



(22) Data do Depósito: 31/01/2012

(45) Data de Concessão: 07/06/2022

(54) Título: MÉTODO DE DECODIFICAÇÃO DE IMAGEM

(51) Int.Cl.: H04N 19/426.

(52) CPC: H04N 19/426.

(30) Prioridade Unionista: 01/07/2011 KR 10-2011-0065707; 03/03/2011 KR 10-2011-0019166; 27/05/2011 KR 10-2011-0050853; 31/01/2011 KR 10-2011-0009636; 31/01/2012 KR 10-2011-0010096.

(73) Titular(es): ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE.

(72) Inventor(es): SUNG CHANG LIM; HUI YONG KIM; SE YOON JEONG; SUK HEE CHO; DONG SAN JUN; JONG HO KIM; HA HYUN LEE; JIN HO LEE; JIN SOO CHOI; JIN WOONG KIM.

(86) Pedido PCT: PCT KR2012000770 de 31/01/2012

(87) Publicação PCT: WO 2012/105807 de 09/08/2012

(85) Data do Início da Fase Nacional: 23/07/2014

(62) Pedido Original do Dividido: BR112013019495-2 - 31/01/2012

(57) Resumo: MÉTODO DE DECODIFICAÇÃO DE IMAGEM A presente invenção refere-se a um método para codificação e decodificação de imagem utilizando um vetor de movimento. De acordo com uma modalidade da presente invenção, um método de decodificação de imagem é fornecido. O método de decodificação de imagem compreende as seguintes etapas: aparar um vetor de movimento de uma imagem de referência com uma faixa dinâmica predeterminada de modo a gerar um vetor de movimento aparado; o armazenamento do vetor de movimento aparado em um armazenador, a derivação de um vetor de movimento de um bloco a ser decodificado utilizando o vetor de movimento armazenado no armazenador; e realização da decodificação interpretável utilizando o vetor de movimento do bloco a ser decodificado. De acordo com a presente invenção, o tamanho de um espaço de memória necessário para armazenar um vetor de movimento pode ser reduzido.

"MÉTODO DE DECODIFICAÇÃO DE IMAGEM"

Divido do BR 11 2013 019495-2, depositado em 31/01/2012.

Campo Técnico

[001] A presente invenção refere-se a processamento de imagem, e mais particularmente, a um método de codificação/decodificação de vídeo utilizando um vetor de movimento cortado (*clipped*).

Técnica Fundamental

[002] Recentemente, de acordo com a expansão de um sistema de difusão suportando resolução de alta definição (HD) no país e em todo o mundo, muitos usuários têm se acostumado a uma imagem de alta resolução e definição, de modo que muitas organizações têm conduzido muitas tentativas de desenvolver os dispositivos de vídeo de próxima geração. Adicionalmente, à medida que o interesse em HDTV e definição ultra alta (UHD) possuindo uma resolução quatro vezes maior do que a HDTV tem aumentado, uma tecnologia de compressão por um vídeo de resolução mais alta e maior definição tem sido exigida.

[003] Para compressão de imagem, uma tecnologia interpredição de predição de valores de pixel incluídos em uma imagem atual a partir de uma imagem anterior e/ou posterior à imagem atual, uma tecnologia de intrapredição de predição de valores de pixel utilizando informação de pixel na imagem, uma tecnologia de codificação por entropia de alocação de um código curto para um símbolo possuindo uma alta frequência de surgimento e um código longo para um símbolo possuindo uma baixa frequência de surgimento, ou similares, pode ser utilizada.

Descrição

Problema Técnico

[004] A presente invenção fornece um método de codificação e decodificação de vídeo utilizando um vetor de movimento cortado e um aparelho para tal.

[005] A presente invenção também fornece um método de cortar um vetor de movimento de uma imagem de referência.

[006] A presente invenção também fornece um método de transmissão de informação em um vetor de movimento.

Solução Técnica

[007] 1. Uma modalidade da presente invenção se refere a um método de codifi-

cação de uma imagem. O método inclui cortar um vetor de movimento de uma imagem de referência em uma faixa dinâmica predetermined para gerar um vetor de movimento cortado, armazenando o vetor de movimento cortado em um armazenador e codificando um vetor de movimento de um bloco de árvore de codificação utilizando um vetor de movimento armazenado no armazenador.

[008] 2. Em 1, a faixa dinâmica pode ser definida por um nível de um codec (codificador/decodificador) de vídeo.

[009] 3. Em 1, a faixa dinâmica pode ser determinada por uma profundidade predetermined de bit, e a profundidade de bit pode ser definida por um nível de um codec de vídeo.

[010] 4. Em 1, o componente X e o componente Y do vetor de movimento da imagem de referência podem ser cortado em diferentes faixas dinâmicas.

[011] 5. Outra modalidade da presente invenção se refere a um método de decodificação de uma imagem. O método inclui cortar um vetor de movimento de uma imagem de referência em uma faixa dinâmica predetermined para gerar um vetor de movimento cortado, o armazenamento do vetor de movimento cortado em um armazenador, a derivação de um vetor de movimento de um bloco de árvore de codificação utilizando o vetor de movimento armazenado no armazenador, e realizando o processo de decodificação de interpredição utilizando o vetor de movimento do bloco de árvore de codificação.

[012] 6. Em 5, a faixa dinâmica pode ser definida por um nível de um codec de vídeo.

[013] 7. Em 5, a faixa dinâmica pode ser determinada por uma profundidade de bit predetermined, e a profundidade de bit pode ser definida por um nível de um codec de vídeo.

[014] 8. Em 5, a faixa dinâmica pode ser determinada por uma profundidade de bit predetermined, e a profundidade de bit pode ser obtida através de um parâmetro de sequência determinado transmitido a partir de um aparelho de codificação de uma imagem.

[015] 9. Em 8, o parâmetro de sequência determinado pode incluir um indicador indicando se o vetor de movimento da imagem de referência é cortado e um parâmetro para obtenção da profundidade de bit.

[016] 10. Em 9, o método de decodificação de uma imagem pode incluir a compressão do vetor de movimento da imagem de referência, onde o parâmetro de sequência determinado inclui um indicador indicando se o vetor de movimento da imagem de referência é comprimida e um parâmetro para obtenção de uma razão de compressão do vetor de movimento da imagem de referência.

[017] 11. Em 5, o método de decodificação de uma imagem pode incluir a limitação de uma resolução de representação do vetor de movimento da imagem de referência.

[018] 12. Em 5, o vetor de movimento cortado pode ser armazenado de acordo com uma prioridade.

[019] 13. Em 5, o vetor de movimento cortado pode ser um vetor de movimento de um bloco codificado no modo interpredição.

[020] 14. Em 5, o método de decodificação de uma imagem pode incluir a realização do escalonamento do vetor de movimento da imagem de referência.

[021] 15. Em 5, o componente X e o componente Y do vetor de movimento da imagem de referência pode ser cortado em diferentes faixas dinâmicas.

[022] 16. Em 15, a faixa dinâmica do componente X e a faixa dinâmica do componente Y podem ser definidas por um nível de um codec de vídeo.

[023] 17. Outra modalidade da presente invenção se refere a um aparelho de decodificação de uma imagem. O aparelho inclui um armazenador de imagem de referência armazenando uma imagem de referência e um compensador de movimento gerando um bloco predito utilizando a imagem de referência e um vetor de movimento da imagem de referência, onde o vetor de movimento da imagem de referência é cortado em uma faixa dinâmica predeterminada.

[024] 18. Em 17, a faixa dinâmica pode ser definida por um nível de um codec de vídeo.

[025] 19. Em 17, a faixa dinâmica pode ser determinada por uma profundidade de bit predeterminada, e a profundidade de bit pode ser definida por um nível de um codec de vídeo.

[026] 20. Em 17, a faixa dinâmica pode ser determinada por uma profundidade de bit predeterminada, e a profundidade de bit pode ser obtida através de um parâmetro de sequência determinado transmitido a partir de um aparelho de codificação de

uma imagem.

Efeitos Vantajosos

[027] De acordo com a modalidade ilustrativa da presente invenção, o vídeo pode ser codificado utilizando o vetor de movimento cortado.

[028] De acordo com a modalidade ilustrativa da presente invenção, um tamanho de uma memória necessária para o armazenamento de vetores pode ser reduzido.

[029] De acordo com a modalidade ilustrativa da presente invenção, uma largura de banda de acesso à memória necessária para se coletar dados da memória pode ser reduzida.

Descrição dos Desenhos

[030] A figura 1 é um diagrama em bloco ilustrando um exemplo de uma estrutura de um codificador de acordo com uma modalidade ilustrativa da presente invenção;

[031] A figura 2 é um diagrama em bloco ilustrando um exemplo de uma estrutura de um decodificador de acordo com uma modalidade ilustrativa da presente invenção;

[032] A figura 3 ilustra exemplos de uma imagem de objeto de codificação e decodificação e uma imagem de referência;

[033] A figura 4 ilustra um exemplo de limitação de uma faixa dinâmica de um vetor de movimento;

[034] As figuras de 5 a 8 são fluxogramas ilustrando um método de armazenamento de um vetor de movimento de uma imagem de referência;

[035] A figura 9 ilustra um exemplo de quantização de um vetor de movimento;

[036] As figuras de 10 a 13 ilustram exemplos de coleta de informação de movimento a partir de uma imagem de referência;

[037] A figura 14 é um fluxograma ilustrando um método de codificação de uma imagem de acordo com a modalidade ilustrativa da presente invenção;

[038] A figura 15 é um fluxograma ilustrando um método de decodificação de uma imagem de acordo com a modalidade ilustrativa da presente invenção.

Modo de Invenção

[039] Doravante, as modalidades ilustrativas da presente invenção serão descritas em detalhes com referência aos desenhos em anexo. No entanto, na descrição das modalidades ilustrativas da presente invenção, funções ou construções bem co-

nhecidas não serão descritas em detalhes visto que podem obscurecer desnecessariamente a compreensão da presente invenção.

[040] Deve-se compreender que quando qualquer elemento é referido como estando "conectado a" ou "acoplado a" outro elemento, o mesmo pode ser conectado diretamente a ou acoplado diretamente a outro elemento ou ser conectado a ou acoplado a outro elemento, possuindo o outro elemento intervindo entre os mesmos. Adicionalmente, na presente especificação, no caso de descrição "incluindo" um componente específico, deve-se compreender que componentes adicionais além de um componente correspondente não estão excluídos, mas podem ser incluídos nas modalidades ilustrativas ou escopo técnico da presente invenção.

[041] Os termos utilizados na especificação, "primeiro", "segundo", etc. podem ser utilizados para descrever vários componentes, mas os componentes não devem ser considerados como estando limitados aos termos. Isso é, os termos são utilizados para distinguir um componente de outro componente. Por exemplo, o "primeiro" componente pode ser chamado de "segundo" componente, e vice-versa, sem se distanciar do escopo da presente invenção.

[042] Adicionalmente, os componentes descritos nas modalidades ilustrativas da presente invenção são ilustrados independentemente apenas a fim de indicar que realizam funções características diferentes. Portanto, os componentes que são ilustrados independentemente não significam que cada um dos componentes pode não ser implementado como um hardware ou software. Isso é, cada um dos componentes é dividido por motivos de conveniência de explicação, uma pluralidade de componentes pode ser combinada um com o outro para, dessa forma, ser operado como um componente ou um componente pode ser dividido em uma pluralidade de componentes para, dessa forma, ser operado como uma pluralidade de componentes, que são incluídos no escopo da presente invenção desde que se distancie das características essenciais da presente invenção.

[043] Adicionalmente, alguns dos componentes podem não ser componentes indispensáveis realizando funções essenciais da presente invenção, mas os componentes seletivos aperfeiçoando apenas o desempenho dos mesmos. A presente invenção também pode ser implementada apenas por uma estrutura incluindo os componentes indispensáveis exceto pelos componentes seletivos, e a estrutura in-

cluindo apenas os componentes indispensáveis também é incluída no escopo da presente invenção.

[044] A figura 1 é um diagrama em bloco ilustrando um exemplo de uma estrutura de um codificador de acordo com uma modalidade ilustrativa da presente invenção.

[045] Com referência à figura 1, o codificador 100 inclui um preditor de movimento 111, um compensador de movimento 112, um intrapreditor 120, um comutador 115, um subtrator 125, um transformador 130, um quantizador 140, um codificador por entropia 150, um desquantizador 160, um transformador invertido 170, um somador 175, uma unidade de filtro 180 e um armazenador de imagem de referência 190.

[046] O codificador 100 codifica as imagens de entrada em um modo de intrapredição ou um modo de interpredição para a saída do codificador de uma sequência de bits. Intrapredição significa predição de intra imagem e interpredição significa predição inter imagem. O codificador 100 é comutado entre o modo de intrapredição e o modo de interpredição através da comutação do comutador 115. O codificador 100 gera um bloco predito para um bloco de entrada da imagem de entrada e então codifica um residual entre o bloco de entrada e o bloco predito.

[047] No caso do modo de intrapredição, o intrapreditor 120 realiza a predição espacial utilizando valores de pixel de blocos vizinhos que são codificados já para gerar os blocos preditos.

[048] No caso do modo de intrapredição, o preditor de movimento 111 busca um bloco de referência idealmente combinado com o bloco de entrada em uma imagem de referência armazenada no armazenador de imagem de referência 190 durante um processo de predição de movimento para obter um vetor de movimento. O compensador de movimento 112 realiza a compensação de movimento utilizando o vetor de movimento para gerar o bloco predito. Aqui, o vetor de movimento pode ser um vetor bidimensional utilizado para a interpredição e representa um desvio entre um bloco de árvore de codificação atual e o bloco de referência.

[049] O subtrator 125 gera um bloco residual com base no residual entre o bloco de entrada e o bloco predito, e o transformador 130 transforma o bloco residual para enviar um coeficiente de transformação. O quantizador 140 quantiza o coeficiente de transformação para enviar o coeficiente quantizado.

[050] O codificador por entropia 150 realiza a codificação por entropia com base

na informação obtida durante um processo de codificação e quantização para enviar a sequência de bits. A codificação por entropia representa símbolos gerados com frequência como um número pequeno de bits, reduzindo, assim, um tamanho de uma sequência de bits para um símbolo de codificação. Portanto, o desempenho de compressão de um vídeo pode ser aperfeiçoado através da codificação por entropia. O codificador por entropia 150 pode utilizar um método de codificação tal como golomb exponencial, codificação de comprimento variável adaptativo de contexto (CAVLC), codificação aritmética binária adaptativa de contexto (CABAC), ou similares, para a codificação por entropia.

[051] Uma imagem codificada precisa ser decodificada novamente e armazenada a fim de ser utilizada como uma imagem de referência para a realização da codificação interpredição. Portanto, o desquantizador 160 desquantiza o coeficiente quantizado, e o transformador invertido 170 transforma de forma invertida o coeficiente desquantizado para enviar um bloco residual reconstruído. O somador 175 soma o bloco residual reconstruído ao bloco predito para gerar um bloco reconstruído.

[052] A unidade de filtro 180 é chamada também de um filtro em circuito adaptativo e aplica pelo menos uma filtragem de desbloqueio, compensação de desvio adaptativo de amostra (SAO), filtragem de circuito adaptativo (ALF) ao bloco reconstruído. A filtragem de desbloqueio significa que a distorção de bloco ocorrida em um limite entre os blocos é removida, e a compensação SAO significa que um desvio adequado é adicionado a um valor de pixel a fim de compensar um erro de codificação. Adicionalmente, ALF significa que a filtragem é realizada com base em um valor de comparação entre uma imagem reconstruída e uma imagem original.

[053] Enquanto isso, o armazenador de imagem de referência 190 armazena o bloco reconstruído passando através da unidade de filtro 180.

[054] A figura 2 é um diagrama em bloco ilustrando um exemplo de uma estrutura de um decodificador de acordo com uma modalidade ilustrativa da presente invenção.

[055] Com referência à figura 2, um decodificador inclui um decodificador por entropia 210, um desquantizador 220, um transformador invertido 230, um intrapreditor 240, um compensador de movimento 250, um somador 255, uma unidade de filtro 260, e um armazenador de imagem de referência 270.

[056] O decodificador 200 decodifica a sequência de bits no modo intrapredição ou o modo de interpredição para enviar uma imagem reconstruída. O decodificador 200 é comutado entre o modo de intrapredição e o modo de interpredição através da comutação do comutador. O decodificador 200 obtém um bloco residual a partir da sequência de bits para gerar um bloco predito e então adiciona o bloco residual e o bloco predito um para outro para gerar um bloco reconstruído.

[057] O decodificador de entropia 210 realiza uma decodificação por entropia com base na distribuição de probabilidade. O processo de decodificação por entropia é um processo oposto ao processo de codificação por entropia mencionado acima. Isso é, o decodificador por entropia 210 gera um símbolo incluindo um coeficiente quantizado a partir da sequência de bits onde um símbolo gerado com frequência é representado como um número pequeno de bits.

[058] O desquantizador 220 desquantiza o coeficiente quantizado, e o transformador invertido 230 transforma de forma invertida o coeficiente desquantizado para gerar um bloco residual.

[059] No caso do modo de intrapredição, o intrapreditor 240 realiza a predição espacial utilizando valores de pixel de blocos vizinhos que já são codificados para gerar os blocos preditos.

[060] No caso do modo de interpredição, o compensador de movimento 250 realiza a compensação de movimento utilizando o vetor de movimento e a imagem de referência armazenada no armazenador de imagem de referência 270 para gerar o bloco predito.

[061] O somador 255 adiciona o bloco predito para o bloco residual, e a unidade de filtro 260 aplica pelo menos um dentre a filtragem de desbloqueio, compensação SAO, ALF para o bloco passando através do somador para enviar uma imagem reconstruída.

[062] A imagem reconstruída pode ser armazenada no armazenador de imagem de referência 270 para, dessa forma, ser utilizado para a compensação de movimento.

[063] Doravante, um bloco significa uma unidade de codificação e decodificação. Em um processo de codificação e decodificação, uma imagem é dividida em um tamanho predeterminado e então codificada e decodificada. Portanto, um bloco tam-

bém pode ser chamado de unidade de codificação (CU), uma unidade de predição (PU), uma unidade de transformação (TU), ou similares, e um bloco único também pode ser dividido em sub-blocos possuindo um tamanho menor.

[064] Aqui, uma unidade de predição significa uma unidade básica na qual a predição e/ou compensação de movimento é realizada. Uma unidade de predição pode ser dividida em uma pluralidade de partições, e cada uma das partições também pode ser chamada de partição de unidade de predição. Quando uma unidade de predição é dividida em uma pluralidade de partições, cada uma das partições de unidade de predição pode se tornar uma unidade básica na qual a predição e/ou compensação de movimento são realizadas. Doravante, na modalidade ilustrativa da presente invenção, uma unidade de predição pode significar também as partições de unidade de predição.

[065] Enquanto isso, na codificação de vídeo de alta eficiência (HEVC), um método de predição de vetor de movimento com base na predição de vetor de movimento avançada (AMVP) é utilizada.

[066] No método de predição de vetor de movimento com base na predição de vetor de movimento avançado, um vetor de movimento (MV) de um bloco, existente em uma posição que é igual a ou corresponde ao de um bloco de árvore de codificação, em uma imagem de referência além de vetores de movimento de blocos reconstruídos posicionados em torno do bloco de árvore de codificação pode ser utilizado. Aqui, o bloco, existente em uma posição que é igual a ou corresponde espacialmente à do bloco de árvore de codificação, na imagem de referência é chamada de bloco colocalizado, e um vetor de movimento do bloco colocalizado é chamado de um vetor de movimento colocalizado ou um vetor de movimento temporal. No entanto, o bloco colocalizado pode ser um bloco, existente em uma posição similar a (isto é, correspondente a) igual à do bloco de árvore de codificação, na imagem de referência além de um bloco existente na mesma posição que o bloco de árvore de codificação.

[067] Em um método de mistura de informação de movimento, a informação de movimento é estimada a partir do bloco colocalizado além de blocos reconstruídos posicionados em torno do bloco de árvore de codificação para, dessa forma, ser utilizado como informação de movimento do bloco de árvore de codificação. Aqui, a

informação de movimento inclui pelo menos uma dentre a informação de modo de interpredição indicando um índice de imagem de referência, um vetor de movimento, uma direção única ou uma direção dupla, ou similares, necessários no momento da interpredição, uma lista de imagem de referência, e informação de modo de predição sobre se a codificação é realizada em um modo de intrapredição ou em um modo de interpredição.

[068] Um vetor de movimento predito no bloco de árvore de codificação pode ser um vetor de movimento do bloco colocalizado, que é um bloco temporalmente adjacente ao bloco de árvore de codificação, além de um dos vetores de movimento dos blocos vizinhos espacialmente adjacentes ao bloco de árvore de codificação.

[069] A figura 3 ilustra exemplos de uma imagem de objeto de codificação e decodificação e uma imagem de referência.

[070] Na figura 3, um bloco X indica um bloco de árvore de codificação em uma imagem de objeto de codificação e decodificação 310, e um bloco A, um bloco B, um bloco C, um bloco D, e um bloco E indicam blocos reconstruídos posicionados em torno do bloco de árvore de codificação. Adicionalmente, um bloco T na imagem de referência 320 indica um bloco colocalizado existente em uma posição correspondente à do bloco de árvore de codificação.

[071] Qual vetor de movimento no bloco de árvore de codificação é utilizado como o vetor de movimento predito pode ser reconhecido através de um índice de preditor de vetor de movimento.

Tabela 1

<code>prediction_unit(x0, y0, log2PUWidth, log2PUHeight) {</code>	Descriptor
<code>if(skip_flag[x0][y0]) {</code>	
<code>if(NumMVPCand(L0) > 1)</code>	
<code> mvp_idx_l0[x0][y0]</code>	<code>ue(v) ae(v)</code>
<code>if(NumMVPCand(L1) > 1)</code>	
<code> mvp_idx_l1[x0][y0]</code>	<code>ue(v) ae(v)</code>
<code>} else if(PredMode == MODE_INTRA) {</code>	
<code> ...</code>	
<code>} else { /* MODE_MERGE, MODE_INTER */</code>	
<code>if(merge_flag[x0][y0] && NumMergeCandidates > 1) {</code>	
<code> ...</code>	
<code>} else {</code>	
<code>if(inter_pred_idc[x0][y0] != Pred_L1) {</code>	
<code>if(NumMVPCand(L0) > 1)</code>	
<code> mvp_idx_l0</code>	<code>ue(v) ae(v)</code>
<code>}</code>	
<code>if(inter_pred_idc[x0][y0] != Pred_L0) {</code>	
<code>if(NumMVPCand(L1) > 1)</code>	
<code> mvp_idx_l1</code>	<code>ue(v) ae(v)</code>
<code>}</code>	
<code>}</code>	
<code>}</code>	
<code>}</code>	

[072] Como ilustrado na tabela 1, os índices de preditor de vetor de movimento `mvp_idx_10` e `mvp_idx_11` para cada lista de imagem de referência são transmitidos para um decodificador, e o decodificador utiliza o mesmo vetor de movimento como um vetor de movimento predito por um codificador como um vetor de movimento predito.

[073] No caso no qual o bloco de árvore de codificação é codificado e decodificado utilizando os vetores de movimento dos blocos vizinhos espacialmente adjacentes ao bloco de árvore de codificação, o vetor de movimento pode ser armazenado apenas com uma memória possuindo um tamanho relativamente pequeno. No entanto, no caso no qual um vetor de movimento temporal é utilizado, visto que todos os vetores de movimento da imagem de referência precisam ser armazenados em uma memória, uma memória possuindo um tamanho relativamente grande é necessária, e um tamanho de uma largura de banda de acesso de memória necessária para se coletar os dados da memória também aumenta. Portanto, existe a necessidade de se armazenar de forma mais eficiente o vetor de movimento temporal em

um ambiente de aplicativo no qual um espaço de memória de um terminal portátil, ou similar, não é suficiente ou o consumo de energia é minimizado.

[074] Enquanto isso, como uma tecnologia de armazenamento de um vetor de movimento em uma memória, existe um método de redução de uma resolução espacial do vetor de movimento. Nesse método, o vetor de movimento é comprimido em qualquer razão e então armazenado na memória. Por exemplo, um vetor de movimento armazenado em uma unidade de bloco 4x4 é armazenado em uma unidade de bloco 4x4 ou maior para reduzir o número de vetores de movimento armazenados. Aqui, a fim de se ajustar um tamanho de bloco do vetor de movimento armazenado, a informação sobre uma razão de compressão é transmitida. A informação é transmitida através de um conjunto de parâmetros de sequência (SPS) como ilustrado na Tabela 2.

Tabela 2

<code>seq_parameter_set_rbsp() {</code>	C	Descriptor
<code>.....</code>		
<code>motion_vector_buffer_comp_flag</code>	0	u(1)
<code>if (motion_vector_buffer_comp_flag)</code>		
<code>motion_vector_buffer_comp_ratio_log2</code>	0	u(8)
<code>rbp_trailing_bits()</code>	0	
<code>}</code>		

[075] Com referência à Tabela 2, no caso no qual `motion_buffer_comp_flag` é igual a 1, um processo de compressão de armazenador de vetor de movimento é realizado.

[076] `motion_vector_buffer_comp_ratio_log2` indica uma razão de compressão do processo de compressão de armazenador de vetor de movimento. No caso no qual o `motion_vector_buffer_comp_ratio_log2` não existe, `motion_vector_buffer_comp_ratio_log2` é estimado para 0, e uma razão de compressão de armazenador de vetor de movimento é representada pela Equação 1.

Equação 1

$$MVBufferCompRatio = 1 \ll \text{motion_vector_buffer_comp_ratio_log2}$$

[077] Por exemplo, no caso no qual todos os blocos 4x4 de 1920x1080 imagens possuem vetores de movimento diferente e utilizam duas listas de imagem de referência, cada uma utilizando duas imagens de referência, um espaço de memória total de 3,21 Mbytes é necessário para armazenar um vetor de movimento temporal

como descrito abaixo.

1. Profundidade de bit de 26 bits por vetor de movimento

(1) Faixa dinâmica de componente X do vetor de movimento: -252 para +7676 (comprimento de bit: 13 bits)

(2) Faixa dinâmica de componente Y de vetor de movimento: -252 para +4316 (comprimento de bit: 13 bits)

(3) (Faixas dinâmicas de cada componente do vetor de movimento foram calculados com base em uma primeira unidade de predição em uma imagem correspondente).

2. No caso no qual todas as unidades de bloco 4x4 possuem vetores de movimento diferentes $480 \times 270 = 129600$ blocos.

3. Uso de dois vetores de movimento por cada bloco

4. O número de listas de imagem de referência: 2

5. Uso de duas imagens de referência por lista de imagem de referência.

=> $26 \text{ bits} \times 129600 \text{ blocos} \times \text{dois vetores de movimento} \times \text{duas listas de imagem de referência} \times \text{duas imagens de referência} = 26956800 \text{ bits} = 3,21 \text{ Mbytes}$.

[078] De acordo com o método de redução de uma resolução espacial de um vetor de movimento como descrito acima, é possível se reduzir o tamanho do espaço de memória necessário e a largura de banda de acesso demonstrando utilizando a correlação espacial do vetor de movimento. No entanto, o método de redução de uma resolução espacial de um vetor de movimento não limita a faixa dinâmica do vetor de movimento.

[079] Quando o tamanho do espaço de memória é reduzido para 1/4, um tamanho de espaço de memória necessário no exemplo mencionado acima é reduzido para cerca de 0,8 Mbytes. Aqui, quando apenas seis bits da profundidade de bit necessária para o armazenamento do vetor de movimento são utilizados para cada componente do vetor de movimento pela limitação adicional da faixa dinâmica do vetor de movimento, o tamanho do espaço de memória necessário pode ser reduzido adicionalmente para 0,37 Mbytes.

[080] Portanto, na modalidade ilustrativa da presente invenção, a faixa dinâmica do vetor de movimento é limitada a fim de reduzir um tamanho de um espaço de memória necessário para o armazenamento do vetor de movimento e uma largura

de banda de acesso de memória necessária para a coleta de dados a partir da memória. O vetor de movimento da imagem de referência do qual a faixa dinâmica é limitada pode ser utilizado como um vetor de movimento temporal no bloco de árvore de codificação.

[081] Doravante, uma faixa dinâmica significa uma faixa entre um valor mínimo e um valor máximo de um componente negativo ou um componente positivo de um vetor de movimento com base em 0, a profundidade de bit, que indica um tamanho de um espaço necessário para o armazenamento do vetor de movimento, significa uma largura de bit. Adicionalmente, a menos que particularmente descrito, o vetor de movimento significa um vetor de movimento de uma imagem de referência, isso é, um vetor de movimento temporal.

[082] No caso no qual cada componente do vetor de movimento está fora da faixa dinâmica, o mesmo é representado pelo valor mínimo ou valor máximo da faixa dinâmica correspondente. Por exemplo, no caso no qual um componente X do vetor de movimento 312 e um valor máximo de uma faixa dinâmica de cada componente do vetor de movimento 256, o componente X do vetor de movimento é limitado a 256.

[083] Da mesma forma, no caso no qual uma profundidade de bit de cada componente do vetor de movimento é igual a 16 bits e o vetor de movimento é (-36, 24), quando a profundidade de bit de cada componente do vetor de movimento é limitado a 6 bits, cada componente do vetor de movimento possui uma faixa dinâmica de -32 a +31, de modo que o vetor de movimento seja representado por (-32, 24) que está em sua faixa dinâmica.

[084] Adicionalmente, no caso no qual uma profundidade de bit de cada componente do vetor de movimento é igual a 16 bits e o vetor de movimento é (-49, 142), quando a profundidade de bit de cada componente do vetor de movimento é limitada a 9 bits, cada componente do vetor de movimento possui uma faixa dinâmica de -256 a +255, de modo que o vetor de movimento seja representado por (-49, 142) sem uma alteração.

[085] A figura 4 ilustra um exemplo de limitação de uma faixa dinâmica de um vetor de movimento.

[086] Com referência à figura 4, quando uma faixa dinâmica de um vetor de mo-

vimento possuindo uma faixa dinâmica de -4096 a +4095 é limitada a -128 a +127, uma profundidade de bit pode ser reduzida de 13 bits para 8 bits.

[087] Cada componente de um vetor de movimento temporal é cortado como representado pelas Equações 2 e 3 a fim de ser armazenado em uma profundidade de bit de N bits. Onde N indica um inteiro positivo.

Equação 2

$$\text{clippedMV_X} = \min(1 \ll (N-1) - 1, \max(-1 \ll (N-1), \text{MV_X}))$$

Equação 3

$$\text{clippedMV_Y} = \min(1 \ll (N-1) - 1, \max(-1 \ll (N-1), \text{MV_Y}))$$

[088] onde MV_X indica um componente X do vetor de movimento, MV_Y indica um componente Y do vetor de movimento, min(a,b) significa uma operação de envio de um valor menor em a e b, e max(a,b) significa uma operação de envio de um valor maior em a e b. Cada MV_X cortado e MV_Y cortado indica os componentes X e Y do vetor de movimento temporal cortado e é armazenado na memória para, dessa forma, ser utilizado como um vetor de movimento temporal do bloco de árvore de codificação.

[089] Por exemplo, como ilustrado na Tabela 3, no caso no qual um tamanho de um espaço de memória é de 48 bytes e cada componente do vetor de movimento utiliza uma profundidade de bit de 16 bits, um total de doze vetores de movimento pode ser armazenado.

Tabela 3

MV1-X	MV1-Y	MV2-X	MV2-Y	MV3-X	MV3-Y	MV4-X	MV4-Y
MV5-X	MV5-Y	MV6-X	MV6-Y	MV7-X	MV7-Y	MV8-X	MV8-Y
MV9-X	MV9-Y	MV10-X	MV10-Y	MV11-X	MV11-Y	MV12-X	MV12-Y

[090] No entanto, quando cada componente do vetor de movimento utiliza apenas uma profundidade de bit de 8 bits, um total de vinte e quatro vetores de movimento pode ser armazenado como ilustrado na Tabela 4.

Tabela 4

MV1-X	MV1-Y	MV2-X	MV2-Y	MV3-X	MV3-Y	MV4-X	MV4-Y
MV5-X	MV5-Y	MV6-X	MV6-Y	MV7-X	MV7-Y	MV8-X	MV8-Y

MV9-X	MV9-Y	MV10-X	MV10-Y	MV11-X	MV11-Y	MV12-X	MV12-Y
-------	-------	--------	--------	--------	--------	--------	--------

[091] Portanto, de acordo com a modalidade ilustrativa da presente invenção, quando uma imagem reconstruída em um codificador e/ou um decodificador é submetida a um processo de filtragem em circuito tal como um filtro de desbloqueio, um filtro de circuito adaptativo, ou similar, e então armazenada em um armazenador de imagem decodificada (DPB), a faixa dinâmica do vetor de movimento é limitada, de modo que um vetor de movimento de uma imagem de referência seja armazenado. O armazenador de imagem decodificada significa o armazenador de imagem de referência da figura 1 ou figura 2.

I. Processo de Corte do Vetor de Movimento

[092] Um processo de corte de cada componente de um vetor de movimento é invocado no caso no qual um slice_type não é igual a I. O processo de corte de um vetor de movimento é realizado em um bloco de árvore ou unidade de codificação maior (LCU) depois de um processo de filtragem ter sido encerrado.

[093] Os registros no processo de corte de um vetor de movimento são uma localização (xP, yP) especificando a amostra superior esquerda da unidade de predição com relação à amostra superior esquerda da imagem atual, e as matrizes de vetor de movimento MvL0 e MvL1. As saídas no processo são as matrizes de vetor de movimento cortado CMvL0 e CMvL1.

[094] Com relação às matrizes MvL0, MvL1, CMvL0 e CMvL1, as operações das Equações 4 a 7 são realizadas.

Equação 4

$$mvLX = MvLX[xP, yP]$$

Equação 5

$$cmvLX[0] = Clip3(-1 << (TMVBitWidth - 1), 1 << (TMVBitWidth - 1) - 1, mvLX[0])$$

Equação 6

$$cmvLX[1] = Clip3(-1 << (TMVBitWidth - 1), 1 << (TMVBitWidth - 1) - 1, mvLX[1])$$

Equação 7

$$CMvLX [xP, yP] = cmvLX$$

onde TMVBitWidth indica uma profundidade de bit de um vetor de movimento, Clip3 (a, b, c) significa uma função de corte c de modo a existir em uma faixa en-

tre a e b.

II. Processo de Armazenamento de Vetor de Movimento

[095] As figuras de 5 a 8 são fluxogramas ilustrando um método de armazenamento de um vetor de movimento de uma imagem de referência.

[096] Com referência à figura 5, o vetor de movimento da imagem de referência pode ser armazenado utilizando ambos um armazenador de imagem armazenando uma imagem reconstruída e um armazenador de vetor de movimento armazenando um vetor de movimento. Aqui, a imagem reconstruída é submetida a um processo de filtragem em circuito (S510) e o vetor de movimento é submetido a um processo de faixa dinâmica limitadora (S520) e então armazenada (S540).

[097] Adicionalmente, com referência à figura 6, ambos um armazenador de imagem e um armazenador de vetor de movimento são utilizados, e o vetor de movimento é submetido a um processo de faixa dinâmica limitadora (S620) e a um processo de resolução espacial reduzida (S630) e então armazenado (S640).

[098] Adicionalmente, com referência à figura 7, a imagem reconstruída é submetida a um processo de filtragem em circuito (S710) e então armazenado em um armazenador de imagem (S740), e o vetor de movimento é submetido a um processo de faixa dinâmica limitadora (S720) e então armazenado em um armazenador de vetor de movimento (S750).

[099] Adicionalmente, com referência à figura 8, a imagem reconstruída é submetida a um processo de filtragem em circuito (S810) e então armazenado em um armazenador de imagem (S840), e o vetor de movimento é submetido a um processo de faixa dinâmica limitadora (S820) e um processo de resolução espacial reduzida (S830) e então armazenado (S850).

[0100] Enquanto isso, nas modalidades ilustrativas das figuras 6 e 8, uma sequência e processo de faixa dinâmica limitadora S620 ou S820 e o processo de resolução espacial reduzida S630 e S830 não é limitada, mas pode ser alterada.

[0101] Adicionalmente, a fim de se reduzir adicionalmente uma largura de banda de acesso à memória, as faixas dinâmicas de cada componente do vetor de movimento podem ser limitadas de forma diferente. Por exemplo, apenas uma dentre uma faixa dinâmica de um componente X e uma faixa dinâmica de um componente Y pode ser limitada ou a faixa dinâmica do componente Y pode ser adicionalmente

limitada em comparação com a faixa dinâmica do componente X.

[0102] A faixa dinâmica limitada do vetor de movimento é transmitida através de um conjunto de parâmetros de sequência, um conjunto de parâmetros de imagem (PPS), um cabeçalho de fatia, ou similares, e o decodificador realiza de forma similar a limitação de uma faixa dinâmica de um vetor de movimento temporal na sequência, na imagem ou na fatia. Nesse caso, uma profundidade de bit, que é um tamanho de um espaço de memória necessário para o armazenamento de vetor de movimento representado na faixa dinâmica também pode ser transmitida. Adicionalmente, é possível se armazenar de forma eficiente o vetor de movimento temporal de modo a ser combinado com as características de movimento da imagem utilizando a faixa dinâmica transmitida através do conjunto de parâmetros de sequência, conjunto de parâmetros de imagem, cabeçalho de fatia, ou similares, ao invés do armazenamento do vetor de movimento utilizando uma profundidade de bit possuindo um tamanho fixo.

[0103] Enquanto isso, o vetor de movimento pode ser quantizado e armazenado. No caso no qual o vetor de movimento é quantizado e armazenado, a precisão do vetor de movimento é reduzida. Como um método de quantização, existe a quantização uniforme na qual os tamanhos de etapa são uniformes, a quantização não uniforme na qual os tamanhos de etapa não são uniformes, e similares. O tamanho de etapa na quantização é determinado para um valor fixo predefinido entre o codificador e o decodificador ou é transmitido a partir do codificador para o decodificador através do conjunto de parâmetros de sequência, o conjunto de parâmetros de imagem, o cabeçalho de fatia ou similares. O decodificador utiliza o vetor de movimento quantizado como está ou desquantiza e utiliza o vetor de movimento quantizado. A figura 9 ilustra um exemplo da quantização de um vetor de movimento. Com referência à figura 9, no caso no qual o vetor de movimento possui um valor de componente de 32 a 48, o vetor de movimento é quantizado para 40.

[0104] Adicionalmente, o vetor de movimento pode ser limitado em uma resolução de representação e armazenado. A resolução de representação significa uma unidade de pixel inteiro (1 unidade de pixel), uma unidade de pixel de fração (uma unidade de pixel 1/2, uma unidade de pixel 1/4, ou similares). Por exemplo, uma resolução do vetor de movimento processado em uma unidade de pixel de 1/4 pode ser arma-

zenada como um pixel inteiro. A resolução de representação do vetor de movimento é determinada para um valor fixo predefinido entre o codificador e o decodificador ou é transmitida a partir do codificador para o decodificador através do conjunto de parâmetros de sequência, conjunto de parâmetros de imagem, cabeçalho de fatia, ou similares.

[0105] Adicionalmente, apenas com relação a alguns vetores de movimento entre os vetores de movimento temporal armazenados em uma memória, um processo de faixa dinâmica limitadora, um processo de resolução de espaço reduzido, e um processo de quantização do vetor de movimento podem ser realizados.

[0106] No caso no qual a faixa dinâmica do vetor de movimento é limitada e armazenada, a informação sobre a faixa dinâmica do vetor de movimento pode ser adicionada ou armazenada à memória. Por exemplo, no caso no qual a faixa dinâmica do vetor de movimento é igual a -128 a +127, um indicador de 1 pode ser adicionalmente armazenado, e no caso no qual a faixa dinâmica do vetor de movimento é -32 a +31, um indicador de 0 pode ser adicionalmente armazenado. Nesse caso, a informação de indicador pode ser armazenada juntamente com o vetor de movimento ou pode ser armazenada em uma memória diferente da memória na qual o vetor de movimento está armazenado. No caso no qual a informação de indicador e o vetor de movimento são armazenados em memórias diferentes, quando em que faixa dinâmica um vetor de movimento específico está armazenado for reconhecido, o acesso arbitrário à informação de indicador pode ser permitida. Adicionalmente, a informação sobre em que faixa dinâmica alguns dos vetores de movimento são armazenados é transmitida através do conjunto de parâmetro de sequência, conjunto de parâmetro de imagem, cabeçalho de fatia, ou similares, possibilitando, assim, poder ser decodificar par realizar uma operação similar à de um codificador.

[0107] No caso no qual a resolução espacial do vetor de movimento é reduzida e armazenada, a informação em um tamanho de bloco do vetor de movimento pode ser adicionada e armazenada na memória. Por exemplo, no caso no qual o tamanho de bloco do vetor de movimento é 4x4, um indicador de 1 pode ser adicionalmente armazenado, e no caso no qual o tamanho de bloco do vetor de movimento é 16x16, um indicador de 0 pode ser adicionalmente armazenado. Nesse caso, a informação de indicador pode ser armazenada juntamente com o vetor d movimento ou pode ser

armazenada em uma memória diferente da memória na qual o vetor de movimento é armazenado. No caso no qual a informação de indicador e o vetor de movimento são armazenados em memórias diferentes, quando em que tamanho de bloco um vetor de movimento específico é armazenado é reconhecido, o acesso arbitrário à informação de indicador pode ser permitido. Adicionalmente, a informação sobre qual tamanho de bloco alguns vetores de movimento são armazenados é transmitida através do conjunto de parâmetros de sequência, conjunto de parâmetros de imagem, cabeçalho de fatia, ou similares, possibilitando, assim, que um decodificador realize uma operação similar à de um codificador.

[0108] No caso no qual o vetor de movimento é quantizado e armazenado, a informação sobre a precisão do vetor de movimento pode ser adicionada e armazenada na memória. Por exemplo, no caso no qual um tamanho de etapa de quantização é igual a 4, um indicador igual a 1 pode ser adicionalmente armazenado, e no caso no qual o tamanho de etapa de quantização é igual a 1, um indicador igual a 0 pode ser adicionalmente armazenado. Nesse caso, a informação de indicador pode ser armazenada juntamente com o vetor de movimento ou pode ser armazenada em uma memória diferente da memória na qual o vetor de movimento é armazenado. No caso no qual a informação de indicador e o vetor de movimento são armazenados em memórias diferentes, quando em que tamanho de etapa um vetor de movimento específico é quantizado e armazenado é reconhecido, o acesso arbitrário à informação de indicador pode ser permitido. Adicionalmente, a informação sobre em que tamanho de etapa alguns vetores de movimento são quantizados e armazenados é transmitida através do conjunto de parâmetro de sequência, o conjunto de parâmetros de imagem, o cabeçalho de fatia, ou similares, possibilitando, assim, que um decodificador realize uma operação similar à de um codificador.

[0109] Adicionalmente, no caso no qual a informação de movimento é armazenada na memória, a resolução espacial do vetor de movimento pode ser reduzida e armazenada. Aqui, a informação de movimento inclui pelo menos uma dentre a informação de modo de interpretação indicando um índice de imagem de referência, um vetor de movimento, uma direção única, uma direção dupla, ou similares, necessários no momento de interpretação, uma lista de imagens de referência, uma informação de modo de predição sobre se um modo de intrapredição é realizado ou um modo

de interpredição é realizado.

[0110] Por exemplo, a informação de movimento de uma unidade de predição possuindo o maior tamanho de divisão dentre uma pluralidade de informações de movimento de uma região específica pode ser armazenada como informação de movimento representativa na memória. Aqui, a região específica pode incluir uma região no bloco de árvore de codificação e regiões dos blocos vizinhos do bloco de árvore de codificação. Adicionalmente, a região específica pode ser uma região incluindo um bloco no qual a informação de movimento é armazenada no caso no qual toda a imagem ou fatia é dividida em um tamanho predeterminado.

[0111] Por exemplo, depois da informação de movimento, que é codificada em um método de mistura de informação de movimento, um método de pular informação de codificação, ou similar, é excluído a partir da pluralidade de informações de movimento incluídas na região específica, a informação de movimento representativa podendo ser armazenada na memória. Por exemplo, a informação de movimento gerada com maior frequência entre a pluralidade de informações de movimento incluída na região específica pode ser armazenada como informação de movimento representativa na memória. Nesse caso, o número de geração de informação de movimento para cada tamanho de bloco, ou similar, pode ser calculado.

[0112] Por exemplo, a informação de movimento em uma posição específica dentre a pluralidade de informações de movimento incluídas na região específica pode ser armazenada. Aqui, a posição específica que é uma posição incluída na região específica, pode ser uma posição fixa da região específica. Adicionalmente, a posição específica pode ser selecionada como uma dentre uma pluralidade de posições. Quando a pluralidade de posições é utilizada, uma prioridade para cada posição pode ser determinada, e a informação de movimento pode ser armazenada na memória de acordo com a prioridade.

[0113] Por exemplo, quando a pluralidade de informações de movimento incluída na região específica é armazenada na memória, visto que a informação de movimento não existe fora de um limite de um bloco codificado em um modo de intrapredição, um bloco codificado em um modo de modulação codificada por pulso (PCM), uma fatia ou uma imagem, a informação de movimento da posição correspondente pode não ser armazenada na memória.

[0114] Nos exemplos mencionados acima, quando a informação de movimento da posição específica é armazenada, no caso no qual a informação de movimento da posição correspondente não existe, a informação de movimento de um bloco colocalizado, informação de movimento de um bloco já codificado, ou informação de movimento de um bloco vizinho podem ser utilizadas como a informação de movimento da posição correspondente. Aqui, a posição específica pode ser uma posição de amostra em um bloco vizinho ou uma posição do bloco. Por exemplo, no caso no qual a informação de movimento da posição específica não existe, um valor médio ou valor médio entre a informação de movimento dos blocos vizinhos que são codificados em interpretação podem ser armazenados na memória. Por exemplo, no caso no qual a informação de movimento da posição específica não existe, um valor médio da informação de posição dos blocos vizinhos pode ser armazenado na memória. Quando o valor médio e o valor de média são calculados, no caso no qual a informação de movimento dos blocos vizinhos é diferente de pelo menos um dos índices de imagem de referência, lista de imagem de referência e informação de modo de interpretação, um tamanho do vetor de movimento pode ser ajustado de acordo com o índice de imagem de referência, lista de imagem de referência, informação de modo de interpretação, contador de ordem de imagem, e similares.

III. Processo de Derivação de Vetor de Movimento

[0115] No caso no qual a informação de movimento é armazenada na memória utilizando os métodos de informação de movimento mencionados acima e a informação de movimento da imagem de referência é utilizada no método de predição de vetor de movimento, o método de predição de vetor de movimento, ou o método de mistura de informação de movimento, a informação de movimento armazenada pode ser coletada.

[0116] Por exemplo, a informação de movimento de uma posição correspondente à do bloco de árvore de codificação na imagem de referência pode ser coletada. Nesse caso, a posição correspondente à do bloco de árvore de codificação na imagem de referência pode ser uma posição fixa em uma região específica ou uma posição relativa a partir da posição do bloco de árvore de codificação.

[0117] As figuras 10 a 13 ilustram exemplos de coleta de informação de movimento a partir de uma imagem de referência.

[0118] Nas figuras de 10 a 13, um bloco X indica um bloco de árvore de codificação em uma imagem de codificação e decodificação de objeto 1010, 1110, 1210 ou 1310 e um bloco A, um bloco B, um bloco C, um bloco D, e um bloco E indica os blocos vizinhos reconstruídos. Adicionalmente, um bloco T na imagem de referência 1020, 1120, 1220, e 1320 indica um bloco colocalizado correspondente ao bloco de árvore de codificação. Um bloco Y na imagem de referência 1320 da figura 13 indica um bloco correspondente a uma posição além do bloco de objeto de codificação e decodificação.

[0119] Com referência à figura 10, a informação de movimento correspondente a uma posição correspondente a uma posição de pixel superior esquerda entre as posições de um bloco de árvore de codificação X em uma imagem de referência pode ser coletada.

[0120] Com referência à figura 11, a informação de movimento correspondente a uma posição correspondente a uma posição de pixel central entre as posições de um bloco de árvore de codificação X em uma imagem de referência pode ser coletada.

[0121] Com referência à figura 12, a informação de movimento correspondente a uma posição correspondente a uma posição de pixel inferior direita entre as posições de um bloco de árvore de codificação X em uma imagem de referência pode ser coletada.

[0122] Com referência à figura 13, a informação de movimento correspondentes a uma posição correspondente a uma posição de pixel além de um bloco de árvore de codificação X em uma imagem de referência pode ser coletada.

[0123] Um método de codificação e decodificação tal como uma predição de vetor de movimento, predição de vetor de movimento avançada, mistura de informação de movimento, pulo de mistura de informação de movimento, ou similares, pode ser realizado utilizando-se a informação de movimento armazenada na memória, isso é, a informação de movimento da imagem de referência.

[0124] O vetor de movimento pode ser armazenado na memória utilizando pelo menos um dentre um método de limitação de uma faixa dinâmica de um vetor de movimento, um método de redução de uma resolução espacial de um vetor de movimento, um método de quantização de um vetor de movimento, e um método de

redução de uma resolução de representação de um vetor de movimento, e o vetor de movimento armazenado pode ser utilizado para prever um vetor de movimento do bloco de árvore de codificação e misturando a informação de movimento.

[0125] Um processo de coleta do vetor de movimento da imagem de referência a partir da memória é chamado de processo de derivação de um vetor de movimento temporal. Em um processo de derivação de um vetor de movimento temporal, TMVBitWidth indica uma largura de bit de um vetor de movimento temporal armazenado na memória.

[0126] Os registros no processo de derivação de um vetor de movimento temporal são uma localização (xP, yP) especificando a amostra de luminescência superior esquerda da unidade de predição atual com relação à amostra superior esquerda da imagem atual, variáveis especificando a largura e a altura da unidade de predição para luminescência, nPSW e nPSH, o índice de referência da partição de unidade de predição atual refIdxLX (com X sendo igual a 0 ou 1). Saídas no processo são a predição de vetor de movimento mvLXCol e o indicador de disponibilidade availableFlagLXCol.

[0127] A função RefPicOrderCnt(pic, refIdx, LX) é especificada pelo valor de PicOrderCnt da imagem que é a imagem de referência RefPicListX[refIdx] de pic com X sendo igual a 0 ou 1. PicOrderCnt da imagem de referência deve ser mantido até que a imagem seja marcada como "não existente". Clip3(a,b) significa uma função corte c de modo a existir em uma faixa entre a e b.

[0128] Se slice_type for igual a B e collocated_from_10_flag for igual a 0, a variável colPic especifica a imagem que contém a partição colocalizada como especificado por RefPicList1[0]. Do contrário, (slice_type é igual a B e collocated_from_10_flag é igual a 1 ou slice_type é igual a P), a variável colPic especifica a imagem que contém a partição colocalizada como especificado por RefPicList0[0].

[0129] A variável colPu e sua posição (xPCol, yPCol) são derivadas nas seguintes etapas ordenadas:

1. posição de luminescência inferior direita (xPRb, yPRb) da unidade de predição atual é definida como representado pelas Equações 8 e 9.

Equação 8

$$xPRb = xP + nPSW$$

Equação 9

$$yPRb = yP + nPSH$$

2. Se colPu for codificado em um modo de intrapredição ou colPu não estiver disponível, (1) a posição de luminescência Central da unidade de predição atual é definida como representada pelas Equações 10 e 11.

Equação 10

$$xPCtr = (xP + (nPSW \gg 1)) - 1$$

Equação 11

$$yPCtr = (yP + (nPSH \gg 1)) - 1$$

(2) A variável colPu é configurada como a unidade de predição que cobre a posição modificada fornecida por $((xPCtr \gg 4) \ll 4, (yPCtr \gg 4) \ll 4)$ dentro de colPic.

3. (xPCol, yPCol) é configurado igual à amostra de luminescência superior esquerda de colPu com relação à amostra de luminescência superior esquerda de colPic.

[0130] As variáveis mvLXCol e availableFlagLXCol são derivadas como se segue:

1. Se colPu for codificada em um modo de intrapredição ou colPu não estiver disponível, ambos os componentes de mvLXCol são configurados iguais a 0 e availableFlagLXCol é configurado igual a 0.

2. Do contrário, (colPu não é codificado em um modo de intrapredição e colPu está disponível), as variáveis mvCol e refldxCol são derivadas como se segue,

(1) Se PredFlagL0[xPCol][yPCol] for igual a 0, o vetor de movimento mvCol e o índice de referência refldxCol são configurados iguais a MvL1[xPCol][yPCol] e RefldxL1[xPCol][yPCol], respectivamente

(2) Do contrário, (PredFlagL0[xPCol][yPCol] é igual a 1), o seguinte se aplica:

1) Se PredFlagL1[xPCol][yPCol] for igual a 0, o vetor de movimento mvCol e o índice de referência refldxCol são configurados igual a MvL0[xPCol][yPCol] e RefldxL0[xPCol][yPCol], respectivamente,

2) Do contrário (PredFlagL1[xPCol][yPCol] é igual a 1), o seguinte se aplica:

a. As designações a seguir são feitas com X sendo igual a 0 ou 1.

i. RefldxColLX = RefldxLX[xPCol][yPCol]

ii. Se PicOrderCnt(colPic) for inferior a PicOrderCnt(currPic) e RefPicOr-

derCnt(colPic, RefldxCOL, LX) for maior que PicOrderCnt(currPic) ou PicOrderCnt(colPic) for maior do que PicOrderCnt(currPic), a variável MvXCross é igual a 1.

iii. Do contrário, (PicOrderCnt(colPic) é inferior a PicOrderCnt(currPic) e RefPicOrderCnt(colPic, RefldxCOL, LX) é inferior a ou igual a PicOrderCnt(currPic) ou PicOrderCnt(colPic) é maior do que PicOrderCnt(currPic) e RefPicOrderCnt(colPic, RefldxCOL, LX) é maior do que ou igual a PicOrderCnt(currPic)), a variável MvXCross é igual a 0.

b. Se uma das seguintes condições for verdadeira, o vetor de movimento mvCol, o índice de referência RefldxCOL e ListCol são configurados iguais a MvL1[xPCOL][yPCOL], RefldxCOL1 e L1, respectivamente.

i. Mv0Cross é igual a 0 e Mv1Cross é igual a 1.

ii. Mv0Cross é igual a Mv1Cross e a lista de índice de referência é igual a L1

c. Do contrário, o vetor de movimento mvCol, o índice de referência refldxCOL e ListCol são configurados para MvL0[xPCOL][yPCOL]. RefldxCOL0 e L0, respectivamente.

3) a variável availableFlagLXCol é configurada igual a 1 e operações da Equação 12 e Equações de 13 a 18 são aplicadas.

a. Se PicOrderCnt(colPic)-RefPicOrderCnt(colPic, refldxCOL, ListCol) for igual a PicOrderCnt(currPic) – RefPicOrderCnt (currPic, refldxLX, LX)

Equação 12

$$mvLXCol = Clip3(-1 << (TMVBitWidth - 1), 1 << (TMVBitWidth - 1) - 1, mvCol)$$

b. Do contrário,

Equação 13

$$tx = (16384 + Abs(td / 2)) / td$$

Equação 14

$$DistScaleFactor = Clip3(-1024, 1023, (tb * tx + 32) >> 6)$$

Equação 15

$$mvLXCol^{\dagger} = Clip3(-1 << (TMVBitWidth - 1), 1 << (TMVBitWidth - 1) - 1, mvCol)$$

Equação 16

$$mvLXCol = ClipMv((DistScaleFactor * mvLXCol^{\dagger} + 128) >> 8)$$

onde td e Tb são derivados das Equações 17 e 18.

Equação 17

$$id = Clip3(-128, 127, PicOrderCnt(colPic) - RefPicOrderCnt(colPic, refIdxCol, ListCol))$$

Equação 18

$$tb = Clip3(-128, 127, PicOrderCnt(currPic) - RefPicOrderCnt(currPic, refIdxLX, LX))$$

[0131] Isso é, com referência às Equações de 13 a 16, mvLXCol é derivado como uma versão escalonada do vetor de movimento mvCol.

[0132] Enquanto isso, apesar de o vetor de movimento ser cortado em uma faixa dinâmica, no caos no qual o vetor de movimento cortado é escalonado, o vetor de movimento cortado pode novamente estar fora da faixa dinâmica. Portanto, depois de o vetor de movimento escalonado ser derivado, a faixa dinâmica do vetor de movimento pode ser limitada. Nesse caso, cada uma das Equações 15 e 16 pode ser substituída pelas Equações 19 e 20.

Equação 19

$$mvLXCol' = ClipMv((DistScaleFactor * mvLXCol + 128) >> 8)$$

Equação 20

$$mvLXCol = Clip3(-1 << (TMVBitWidth - 1), 1 << (TMVBitWidth - 1) - 1, mvCol')$$

IV. Método de Transmissão de Informação para Corte de Vetor de Movimento Temporal no Decodificador

[0133] Doravante, um método de transmissão de informação necessária para o corte cortado de um vetor de movimento temporal em um decodificador utilizando o mesmo método que o de um codificador será descrito.

[0134] TMVBitWidth no processo de derivação de um vetor de movimento temporal pode ser transmitido a partir do codificador para o decodificador através de um conjunto de parâmetros de sequência, um conjunto de parâmetros de imagem, um cabeçalho de fatia, ou similares.

Tabela 5

seq_parameter_set_rbsp() {	C	Descriptor
.....		
bit_width_temporal_motion_vector_minus8	0	se(v)
.....		
rbp_trailing_bits()	0	
}		

[0135] bit_width_temporal_motion_vector_minus8 da Tabela 5 especifica a largura

de bit do componente de vetor de movimento temporal. Quando `bit_width_temporal_motion_vector_minus8` não está presente, deve ser inferido como igual a 0. A largura de bit do componente de vetor de movimento temporal é especificada como segue:

Equação 21

$$TMVBitWidth = bit_width_temporal_motion_vector_minus8 + 8$$

[0136] Método de transmissão de informação 1 – no caso no qual o vetor de movimento é comprimido e a profundidade de bit do vetor de movimento é limitada.

Tabela 6

seq_parameter_set_rbsp() {	C	Descriptor
.....		
motion_vector_buffer_comp_flag	0	u(1)
if(motion_vector_buffer_comp_flag)		
motion_vector_buffer_comp_ratio_log2	0	u(8)
bit_depth_temporal_motion_vector_constraint_flag	0	u(1)
if(bit_depth_temporal_motion_vector_constraint_flag)		
bit_depth_temporal_motion_vector_minus8	0	se(v)
.....		
rsbp_trailing_bits()	0	
}		

[0137] Com referência à Tabela 6, no caso no qual `motion_vector_buffer_comp_flag` é igual a 1 o processo de compressão de armazenador de vetor de movimento é aplicado.

[0138] `motion_vector_buffer_comp_ratio_log2` especifica a razão de compressão no processo de compressão de armazenador de vetor de movimento. Quando `motion_vector_buffer_comp_ratio_log2` não está presente, o mesmo deve ser inferido como sendo igual a 0. A razão de compressão de armazenador de vetor de movimento é especificada como se segue.

Equação 22

$$MVBufferCompRatio = 1 \ll motion_vector_buffer_comp_ratio_log2$$

[0139] Novamente com referência à Tabela 6, no caso no qual `bit_depth_temporal_motion_vector_constraint_flag` é igual a 1 o processo de restrição de limitação de profundidade de bit de vetor de movimento temporal é aplicado.

[0140] `bit_depth_temporal_motion_vector_minus8` especifica a profundidade de bit do vetor de movimento temporal. Quando

`bit_depth_temporal_motion_vector_minus8` não está presente, deve ser inferido como sendo igual a 0. A profundidade de bit do vetor de movimento temporal é especificada como segue:

Equação 23

$$TMVBitDepth = bit_depth_temporal_motion_vector_minus8 + 8$$

[0141] 2. Método de transmissão de informação 2 – no caso no qual a profundidade de bit do vetor de movimento é limitado

Tabela 7

<code>seq_parameter_set_rbsp() {</code>	C	Descriptor
.....		
<code>bit_depth_temporal_motion_vector_constraint_flag</code>	0	<code>u(1)</code>
<code>if (bit_depth_temporal_motion_vector_constraint_flag)</code>		
<code>bit_depth_temporal_motion_vector_minus8</code>	0	<code>se(v)</code>
.....		
<code>rbp_trailing_bits()</code>	0	
<code>}</code>		

[0142] Com referência à Tabela 7, no caso no qual `bit_depth_temporal_motion_vector_constraint_flag` é igual a 1 o processo de restrição de profundidade de bit de vetor de movimento temporal é aplicado.

[0143] `bit_depth_temporal_motion_vector_minus8` especifica a profundidade de bit do vetor de movimento temporal. Quando `bit_depth_temporal_motion_vector_minus8` não está presente, deve ser inferior como sendo igual a 0. A profundidade de bit do vetor de movimento temporal é especificada como se segue:

Equação 24

$$TMVBitDepth = bit_depth_temporal_motion_vector_minus8 + 8$$

[0144] 3. Método de transmissão de informação 3 – no caso no qual a profundidade de bit do vetor de movimento é limitada.

Tabela 8

<code>seq_parameter_set_rbsp() {</code>	C	Descriptor
.....		
<code>bit_depth_temporal_motion_vector_minus8</code>	0	<code>se(v)</code>
.....		
<code>rbp_trailing_bits()</code>	0	
<code>}</code>		

[0145] `bit_depth_temporal_motion_vector_minus8` especifica a profundidade de bit do vetor de movimento temporal. Quando `bit_depth_temporal_motion_vector_minus8` não está presente, deve ser inferido que é igual a 0. A profundidade de bit do vetor de movimento temporal é especificado como segue.

Equação 25

$$TMVBitDepth = bit_depth_temporal_motion_vector_minus8 + 8$$

[0146] 4. Método de transmissão de informação 4 – no caso no qual a profundidade de bit é limitada com relação a cada um dos componentes X e Y do vetor de movimento

Tabela 9

<code>seq_parameter_set_rbsp() {</code>	C	Descriptor
.....		
<code>bit_depth_temporal_motion_vector_constraint_flag</code>	0	u(1)
<code>if(bit_depth_temporal_motion_vector_constraint_flag) {</code>		
<code>bit_depth_temporal_motion_vector_x_minus8</code>	0	se(v)
<code>bit_depth_temporal_motion_vector_y_minus8</code>	0	se(v)
<code>}</code>		
.....		
<code>rbsp_trailing_bits()</code>	0	
<code>}</code>		

[0147] Com referência à Tabela 9, no caso no qual `bit_depth_temporal_motion_vector_constraint_flag` é igual a 1 o processo de restrição de profundidade de bit de vetor de movimento temporal é aplicado.

[0148] `bit_depth_temporal_motion_vector_x_minus8` especifica a profundidade de bit do componente de vetor de movimento temporal x. Quando `bit_depth_temporal_motion_vector_x_minus8` não está presente, deve ser inferido como sendo igual a 0. A profundidade de bit do componente de vetor de movimento temporal x é especificado como segue:

Equação 26

$$TMVXBitDepth = bit_depth_temporal_motion_vector_x_minus8 + 8$$

[0149] `bit_depth_temporal_motion_vector_y_minus8` especifica a profundidade de bit do componente de vetor de movimento temporal y. Quando `bit_depth_temporal_motion_vector_y_minus8` não está presente, deve ser inferido que seja igual a 0. A profundidade de bit do componente de vetor de movimento

temporal y é especificada com se segue:

Equação 27

$$TMVXBitDepth = bit_depth_temporal_motion_vector_y_minus8 + 8$$

[0150] 5. Método de transmissão de informação 5 – no caso no qual o vetor de movimento é comprimido e a profundidade de bit do vetor de movimento é limitada.

Tabela 10

seq_parameter_set_rbsp() {	C	Descriptor
.....		
motion_vector_buffer_comp_flag	0	u(1)
if (motion_vector_buffer_comp_flag) {		
motion_vector_buffer_comp_ratio_log2	0	u(8)
bit_depth_temporal_motion_vector_minus8	0	se(v)
}		
.....		
rbp_trailing_bits()	0	
}		

[0151] Com referência à Tabela 10, no caso no qual motion_vector_buffer_comp_flag é igual a 1 o processo de compressão de armazenador de vetor de movimento é aplicado.

[0152] motion_vector_buffer_comp_ratio_log2 especifica a razão de compressão no processo de compressão de armazenador de vetor de movimento. Quando motion_vector_buffer_comp_ratio_log2 não está presente, deve ser inferido como sendo igual a 0. A razão de compressão de armazenador de vetor de movimento é especificada como segue:

Equação 28

$$MVBufferCompRatio = 1 \ll motion_vector_buffer_comp_ratio_log2$$

V. Definição de uma Faixa Dinâmica através dos Níveis de CODEC de Vídeo

[0153] A faixa dinâmica do vetor de movimento temporal pode ser definida através de um nível de um codec de vídeo ao invés de ser transmitida através do conjunto de parâmetro de sequência, conjunto de parâmetro de imagem, ou cabeçalho de fatia. O codificador e o decodificador podem determinar uma faixa dinâmica limitada de vetor de movimento utilizando informação de nível.

[0154] Adicionalmente, mesmo nos níveis, as faixas dinâmicas e/ou profundidades de bit de cada um dos componentes X e Y do vetor de movimento podem ser definidas diferentemente, e valores mínimos e valores máximos de cada um dos compo-

nentes podem ser definidos.

[0155] As Tabelas 11 e 12 ilustram um exemplo de um caso no qual TMVBitWidth no processo de derivação de um vetor de movimento temporal descrito acima é definido nos níveis.

Tabela 11

Número de nível	MaxTMVBitWidth (largura de bit de componente MV temporal máxima)
1	8
b	8
1,1	8
1,2	8
1,3	8
2	8
2,1	8
2,2	8
3	8
3,1	10
3,2	10
4	10
4,1	10
4,2	10
5	10
5,1	10

[0156] Com referência à Tabela 11, TMVBitWidth é configurado como MaxTMVBitWidth definido nos níveis. Aqui, MaxTMVBitWidth dica uma largura máxima de bit de um vetor de movimento temporal quando o vetor de movimento temporal é armazenado na memória.

[0157] Enquanto isso, TMVBitWidth também pode ser definido nos níveis, e uma diferença do valor definido (um valor delta) pode ser transmitida através do conjunto de parâmetros de sequência, conjunto de parâmetros de imagem, ou cabeçalho de fatia. Isso é, TMVBitWidth pode ser configurado como um valor obtido pela adição da diferença transmitida através do conjunto de parâmetro de sequência, conjunto

de parâmetro de imagem, ou cabeçalho de fatia para MaxTMVBitWidth definido nos níveis. Aqui, TMVBitWidth indica uma largura de bit do vetor de movimento temporal quando o vetor de movimento temporal é armazenado na memória.

Tabela 12

Número de nível	MaxTMVBitWidth (largura de bit de componente MV temporal máxima)
1	8
b	8
1,1	8
1,2	8
1,3	8
2	8
2,1	8
2,2	8
3	8
3,1	10
3,2	10
4	10
4,1	10
4,2	10
5	10
5,1	10

Tabela 13

seq_parameter_set_rbsp() {	C	Descriptor
.....		
delta_bit_width_temporal_motion_vector_minus8	0	se(v)
.....		
rbp_trailing_bits()	0	
}		

[0158] delta_bit_width_temporal_motion_vector_minus8 especifica a largura de bit delta do componente de vetor de movimento temporal. Quando delta_bit_width_temporal_motion_vector_minus8 não está presente, deve ser inferido que é igual a 9. A largura de bit do componente de vetor de movimento temporal é

especificada como se segue:

Equação 29

$$\text{TMVBitWidth} = \text{delta_bit_width_temporal_motion_vector_minus8} + \text{Max-TMVBitWidth}.$$

[0159] Adicionalmente, como ilustrado na Tabela 14, as faixas dinâmicas de cada componente do vetor de movimento temporal também podem ser definidas em níveis.

Tabela 14

Número de nível	Max-MBPS (taxa de processamento de macrobloco máxima) (MB/s)	MaxF S (tamanho máximo de imagem) (MBs)	MaxD-pbMbs (tamanho de armazénador de imagem decodificada máximo) (MBs)	MaxBR (taxa de bit de vídeo máxima) (1000 bits/s, 1200 bits/s, cpbBrVclFactor bits/s, ou cpbBrNalFactor bits/s)	MaxCPB (tamanho de CPB máximo) (1000 bits, 1200 bits, pbBrVclFactor bits, ou cpbBrNalFactor bits)	MaxTmvr (faixa de componente MV temporal máximo) (amostras de imagem de luminescência)	MinCR (razão de compressão mínima)	MaxMvs-Per2Mb (número máximo de vetores de movimento por dois MBs consecutivos)
1	1485	99	396	64	175	[-64, +63,75]	2	-
b	1485	99	396	128	350	[-64, +63,75]	2	--
1,1	3000	396	900	192	500	[-128, +127,75]	2	-
1,2	6000	396	2376	384	1000	[-128, +127,75]	2	-
1,3	11880	396	2376	768	2000	[-128, +127,75]	2	-
2	11880	396	2376	2000	2000	[-128, +127,75]	2	-
2,1	19800	792	4752	4000	4000	[-256, +255,75]	2	-
2,2	20250	1620	8100	4000	4000	[-256,	2	-

						+255,75]		
3	40500	1620	8100	10000	10000	[-256, +255,75]	2	32
3,1	108000	3600	18000	14000	14000	[-512, +511,75]	4	16
3,2	216000	5120	20480	20000	20000	[-512, +511,75]	4	16
4	245760	8192	32768	20000	25000	[-512, +511,75]	4	16
4,1	245760	8192	32768	50000	62500	[-512, +511,75]	2	16
4,2	522240	8704	34816	50000	62500	[-512, +511,75]	2	16
5	589824	22080	110400	135000	135000	[-512, +511,75]	2	16
5,1	983040	36864	184320	240000	240000	[-512, +511,75]	2	16

[0160] Adicionalmente, como ilustrado nas Tabelas de 15 a 17, as larguras de bit de cada componente do vetor de movimento temporal também podem ser definidos em níveis.

[0161] Tabela 15

Número de nível	Max-MBPS (taxa de processamento de macrobloco máxima) (MB/s)	MaxFS (tamanho máximo de imagem) (MBs)	MaxDPbMbs (tamanho de armazenador de imagem decodificada máximo) (MBs)	MaxBR (taxa de bit de vídeo máxima) (1000 bits/s, 1200 bits/s, cpbBrVclFactor bits/s, ou cpbBr-NalFactor bits/s)	MaxCPB (tamanho CPB máximo) (1000 bits, 1200 bits, pbBrVclFactor bits, ou cpbBr-NalFactor bits)	Max-TMVBitWidth (largura máxima de bit de componente MV temporal)	MinCR (razão de compressão mínima)	MaxMvs-Per2Mb (número máximo de vetores de movimento por dois MBs consecutivos)
-----------------	--	--	--	--	---	---	------------------------------------	---

1	1485	99	396	64	175	8	2	-
b	1485	99	396	128	350	8	2	--
1,1	3000	396	900	192	500	8	2	-
1,2	6000	396	2376	384	1000	8	2	-
1,3	11880	396	2376	768	2000	8	2	-
2	11880	396	2376	2000	2000	8	2	-
2,1	19800	792	4752	4000	4000	8	2	-
2,2	20250	1620	8100	4000	4000	8	2	-
3	40500	1620	8100	10000	10000	8	2	32
3,1	108000	3600	18000	14000	14000	10	4	16
3,2	216000	5120	20480	20000	20000	10	4	16
4	245760	8192	32768	20000	25000	10	4	16
4,1	245760	8192	32768	50000	62500	10	2	16
4,2	522240	8704	34816	50000	62500	10	2	16
5	589824	22080	110400	135000	135000	10	2	16
5,1	983040	36864	184320	240000	240000	10	2	16

Tabela 16

Número de nível	Max-MBPS (taxa de processamento de macrobloco máxima) (MB/s)	MaxFS (tamanho máximo de imagem) (MBs)	MaxDpbMbs (tamanho de armazénador de imagem decodificada máximo) (MBs)	MaxBR (taxa de bit de vídeo máxima) (1000 bits/s, 1200 bits/s, cpbBrVclFactor bits/s, ou cpbBr-NalFactor bits/s)	MaxCPB (tamanho CPB máximo) (1000 bits, 1200 bits, pbBrVclFactor bits, ou cpbBr-NalFactor bits)	Max-TMVBitWidth (largura máxima de bit de componente MV temporal)	MinCR (razão de compressão mínima)	MaxMvs-Per2Mb (número máximo de vetores de movimento por dois MBs consecutivos)
1	1485	99	396	64	175	6	2	-
b	1485	99	396	128	350	6	2	--

1,1	3000	396	900	192	500	7	2	-
1,2	6000	396	2376	384	1000	7	2	-
1,3	11880	396	2376	768	2000	7	2	-
2	11880	396	2376	2000	2000	7	2	-
2,1	19800	792	4752	4000	4000	8	2	-
2,2	20250	1620	8100	4000	4000	8	2	-
3	40500	1620	8100	10000	10000	8	2	32
3,1	108000	3600	18000	14000	14000	10	4	16
3,2	216000	5120	20480	20000	20000	10	4	16
4	245760	8192	32768	20000	25000	10	4	16
4,1	245760	8192	32768	50000	62500	10	2	16
4,2	522240	8704	34816	50000	62500	10	2	16
5	589824	22080	110400	135000	135000	10	2	16
5,1	983040	36864	184320	240000	240000	10	2	16

Tabela 17

Número de nível	MaxTMVBitWidth (largura de bit de componente MV temporal máxima)
1	8
b	8
1,1	8
1,2	8
1,3	8
2	8
2,1	8
2,2	8
3	8
3,1	10
3,2	10
4	10
4,1	10
4,2	10
5	10

5,1	10
-----	----

[0162] Adicionalmente, como ilustrado na Tabela 18, uma largura de bit de um componente Y do vetor de movimento temporal também pode ser definida em níveis.

Tabela 18

Número de nível	Max-MBPS (taxa de processamento de macrobloco máxima) (MB/s)	MaxFS (tamanho máximo de imagem) (MBs)	MaxDPbMs (tamanho de armazenador de imagem decodificada máximo) (MBs)	MaxBR (taxa de bit de vídeo máxima) (1000 bits/s, 1200 bits/s, cpbBrVcl Fctor bits/s, ou cpbBr-NalFactor bits/s)	MaxCPB (tamanho CPB máximo) (1000 bits, 1200 bits, pbBrVcl-Factor bits, ou cpbBr-NalFactor bits)	Max-TMVYBitWidth (profundidade de bit de componente MV temporal vertical máxima)	MinCR (razão de compressão mínima)	MaxMvs-Per2Mb (número máximo de vetores de movimento por dois MBs consecutivos)
1	1485	99	396	64	175	8	2	-
b	1485	99	396	128	350	8	2	--
1,1	3000	396	900	192	500	8	2	-
1,2	6000	396	2376	384	1000	8	2	-
1,3	11880	396	2376	768	2000	8	2	-
2	11880	396	2376	2000	2000	8	2	-
2,1	19800	792	4752	4000	4000	8	2	-
2,2	20250	1620	8100	4000	4000	8	2	-
3	40500	1620	8100	10000	10000	8	2	32
3,1	108000	3600	18000	14000	14000	10	4	16
3,2	216000	5120	20480	20000	20000	10	4	16
4	245760	8192	32768	20000	25000	10	4	16
4,1	245760	8192	32768	50000	62500	10	2	16
4,2	522240	8704	34816	50000	62500	10	2	16
5	589824	22080	110400	135000	135000	10	2	16
5,1	983040	36864	184320	240000	240000	10	2	16

[0163] Adicionalmente, a faixa dinâmica do vetor de movimento temporal pode ser definida como um valor fixo predefinido entre o codificador e o decodificador sem

transmissão da informação em uma limitação do vetor de movimento ou ser armazenado em uma forma de uma profundidade de bit fixo.

[0164] No caso no qual TMVBitWidth é fixo no mesmo valor e utilizado no codificador e no decodificador, TMVBitWidth pode ser um inteiro positivo tal como 4, 6, 8, 10, 12, 14, 16 ou similar. Aqui, TMVBitWidth indica a largura do bit do vetor de movimento temporal quando o vetor de movimento temporal é armazenado na memória.

[0165] A figura 14 é um fluxograma ilustrando um método de codificação de uma imagem de acordo com a modalidade ilustrativa da presente invenção. Com referência à figura 14, o método de codificação de uma imagem inclui uma etapa de corte (S1410), uma etapa de armazenamento (S1420) e uma etapa de codificação (S1430).

[0166] Um aparelho de codificação de uma imagem e/ou um aparelho de decodificação de uma de imagem cortar um vetor de movimento de uma imagem de referência em uma faixa dinâmica predeterminada (S1410). Como descrito acima através de "I. Processo de vetor de movimento de corte", o vetor de movimento que está fora da faixa dinâmica é representado por um valor mínimo ou um valor máximo da faixa dinâmica correspondente. Portanto, como descrito acima através de "IV. Método de Transmissão de informação para Corte de Vetor de Movimento Temporal no Decodificador", e "V. Definição da Faixa Dinâmica Através de Níveis de CODEC de Vídeo", a profundidade de bit é limitada através do nível do codec de vídeo, conjunto de parâmetro de sequência, e similares, ou a faixa dinâmica é limitada através do nível de codec de vídeo, possibilitando assim o corte do vetor de movimento da imagem de referência na faixa dinâmica predeterminada.

[0167] O aparelho de codificação de uma imagem e/ou o aparelho de decodificação de uma imagem armazenam o vetor de movimento cortado da imagem de referência em um armazenador como descrito acima através de "II. Processo de Armazenamento de Vetor de Movimento" (S1420). O vetor de movimento pode ser armazenado no armazenador juntamente com ou separadamente da imagem reconstruída.

[0168] O aparelho de codificação de uma imagem codifica um vetor de movimento de um bloco de árvore de codificação utilizando o vetor de movimento armazenado

da imagem de referência (S1430). Como descrito acima através de "III. Processo de Derivação de Vetor de Movimento", no método de predição de vetor de movimento avançado utilizado em HEVC, um vetor de movimento de um bloco existente em uma posição que é igual a ou corresponde à do bloco de árvore de codificação na imagem de referência além dos vetores de movimento dos blocos reconstruídos posicionados em torno do bloco de árvore de codificação pode ser utilizado. Portanto, o vetor de movimento do bloco de árvore de codificação também pode ser um vetor de movimento da imagem de referência, isso é, um vetor de movimento temporal, além de vetores de movimento dos blocos vizinhos adjacentes ao bloco de árvore de codificação.

[0169] Enquanto isso, visto que a faixa dinâmica do componente X e componente Y do vetor de movimento da imagem de referência pode ser definida diferentemente, cada componente do vetor de movimento da imagem de referência pode ser cortado em cada faixa dinâmica.

[0170] Adicionalmente, um método de compressão de vetor de movimento de uma imagem de referência além de um método de limitação de uma faixa dinâmica de um vetor de movimento de uma imagem de referência pode ser utilizado. No caso de limitação de faixa dinâmica do vetor de movimento da imagem de referência ou compressão de vetor de movimento da imagem de referência, um indicador indicando a faixa dinâmica e o vetor de movimento e um parâmetro relacionado com isso podem ser definidos em um nível de um codec de vídeo, um conjunto de parâmetros de sequência, ou similares.

[0171] Adicionalmente, um método de codificação tal como uma predição de vetor de movimento, predição de vetor de movimento avançado, mistura de informação de movimento, pulo de mistura de informação de movimento, ou similar, pode ser realizado utilizando a informação de movimento armazenada na memória, isso é, a informação de movimento da imagem de referência.

[0172] A figura 15 é um fluxograma ilustrando um método de decodificação de uma imagem de acordo com a modalidade ilustrativa da presente invenção. Com referência à figura 15, o método de decodificação de uma imagem inclui uma etapa de corte (S1510), uma etapa de armazenamento (S1520), uma etapa de derivação (S1530) e uma etapa de decodificação (S1540).

[0173] A etapa de corte (S1510) e a etapa de armazenamento (S1520) da figura 15 são similares à etapa de corte (S1410) e a etapa de armazenamento (S1420) da figura 14 utilizando "I. Processo de Corte de Vetor de Movimento" e "II. Processo de Armazenamento de Vetor de Movimento" descritos acima. Adicionalmente, a etapa de derivação (S1530) da figura 15 utiliza "III. Processo de Derivação de Vetor de Movimento" descrito acima e é simétrica à etapa de codificação (S1430) da figura 14. Portanto, uma descrição detalhada será omitida.

[0174] Um aparelho de decodificação de uma imagem realiza a decodificação por interpredição utilizando um vetor de movimento de um bloco de árvore de codificação (S1540). O aparelho de decodificação de uma imagem pode armazenar um vetor de movimento em uma memória utilizando pelo menos um dentre um método de limitação de uma faixa dinâmica de um vetor de movimento, um método de redução de uma resolução espacial de um vetor de movimento, um método de quantização de um vetor de movimento, e um método de redução de uma resolução de representação de um vetor de movimento, e utilizar o vetor de movimento armazenado para prever um vetor de movimento do bloco de árvore de codificação e misturar a informação de movimento do mesmo.

[0175] Adicionalmente, o aparelho de decodificação de uma imagem pode realizar um método de decodificação tal como a predição de vetor de movimento, predição de vetor de movimento avançada, mistura de informação de movimento, pulo de mistura de informação de movimento, ou similar, utilizando a informação de movimento armazenada na memória, isso é, a informação de movimento da imagem de referência.

[0176] Apesar de as modalidades ilustrativas mencionadas acima terem sido descritas através de fluxogramas representados por uma série de etapas ou blocos, a presente invenção não está limitada a uma sequência de etapas descrita acima. Isso é, algumas etapas podem ser geradas em uma sequência diferente ou simultaneamente a partir de ou com outras etapas. Adicionalmente, pode ser compreendido pelos versados na técnica à qual a presente invenção pertence que as etapas ilustradas nos fluxogramas são não exclusivas, de modo que outras etapas possam ser incluídas ou algumas etapas possam ser eliminadas.

[0177] Adicionalmente, as modalidades ilustrativas mencionadas acima incluem

exemplos de vários aspectos. Apesar de todas as possíveis combinações para ilustrar vários aspectos não serem descritas, pode ser apreciado pelos versados na técnica que outras combinações podem ser feitas. Portanto, a presente invenção deve ser considerada como incluindo todas as outras substituições, alterações e modificações que pertencem às reivindicações a seguir.

REIVINDICAÇÕES

1. Método de decodificação de imagem, **CARACTERIZADO** pelo fato de compreender:

calcular um fator de escalonamento com base em uma contagem de ordem de imagem de uma imagem de referência;

cortar (*clipping*) o fator de escalonamento em uma primeira faixa predeterminada;

escalonar um vetor de movimento da imagem de referência com base no fator de escalonamento cortado;

cortar o vetor de movimento escalonado da imagem de referência em uma segunda faixa predeterminada; e

derivar um vetor de movimento de um bloco a ser decodificado utilizando o vetor de movimento escalonado cortado; e

realizar decodificação de interpriedição utilizando o vetor de movimento do bloco a ser decodificado para gerar um bloco de predição com base na imagem de referência e no vetor de movimento escalonado cortado da imagem de referência,

em que o vetor de movimento da imagem de referência é determinado como um vetor de movimento de um bloco colocalizado na imagem de referência,

em que o vetor de movimento do bloco colocalizado é determinado ao executar as etapas ordenadas de:

calcular uma primeira posição correspondendo a uma posição direita inferior representando uma posição deslocada a partir de uma posição esquerda superior de um bloco atual em uma imagem atual pela altura e largura do bloco atual,

determinar um primeiro bloco cobrindo a primeira posição na imagem de referência,

determinar se o primeiro bloco é codificado em um modo de intrapriedição,

calcular, quando o primeiro bloco é codificado em um modo de intrapriedição, uma segunda posição correspondendo a uma posição central do bloco atual na imagem atual, e

determinar um segundo bloco cobrindo a segunda posição na imagem de referência.

2. Método de decodificação de imagem, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de adicionalmente compreender o escalonamento por magnitude do vetor de movimento da imagem de referência;

em que o vetor de movimento da imagem de referência escalonada por magnitude é cortado na faixa.

3. Método de decodificação de imagem, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato do vetor de movimento ser cortado em uma faixa de valor fixo predeterminado.

4. Método de decodificação de imagem, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato do vetor de movimento da imagem de referência ser armazenado em uma unidade de bloco predeterminada; e

em que, na derivação do vetor de movimento, o vetor de movimento da imagem de referência armazenada na unidade de bloco predeterminada é derivado.

5. Método de decodificação de imagem, de acordo com a reivindicação 3, **CARACTERIZADO** pelo fato de que os componentes X e Y do vetor de movimento são cortados na mesma faixa de valor fixo.

6. Método de decodificação de imagem, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato do vetor de movimento ser um vetor de movimento de um bloco decodificado em um modo de interpredição.

FIG. 1

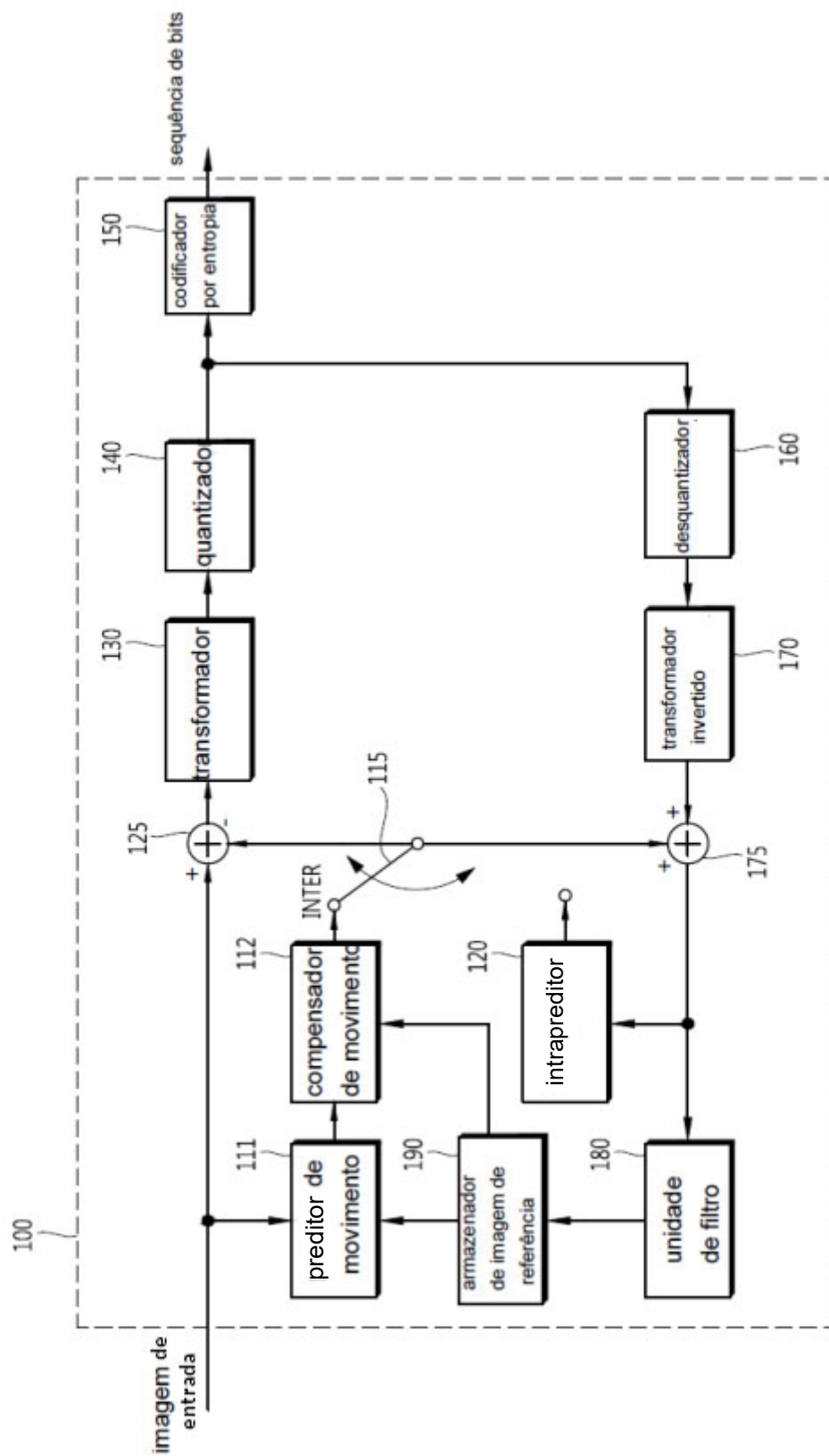


FIG. 2

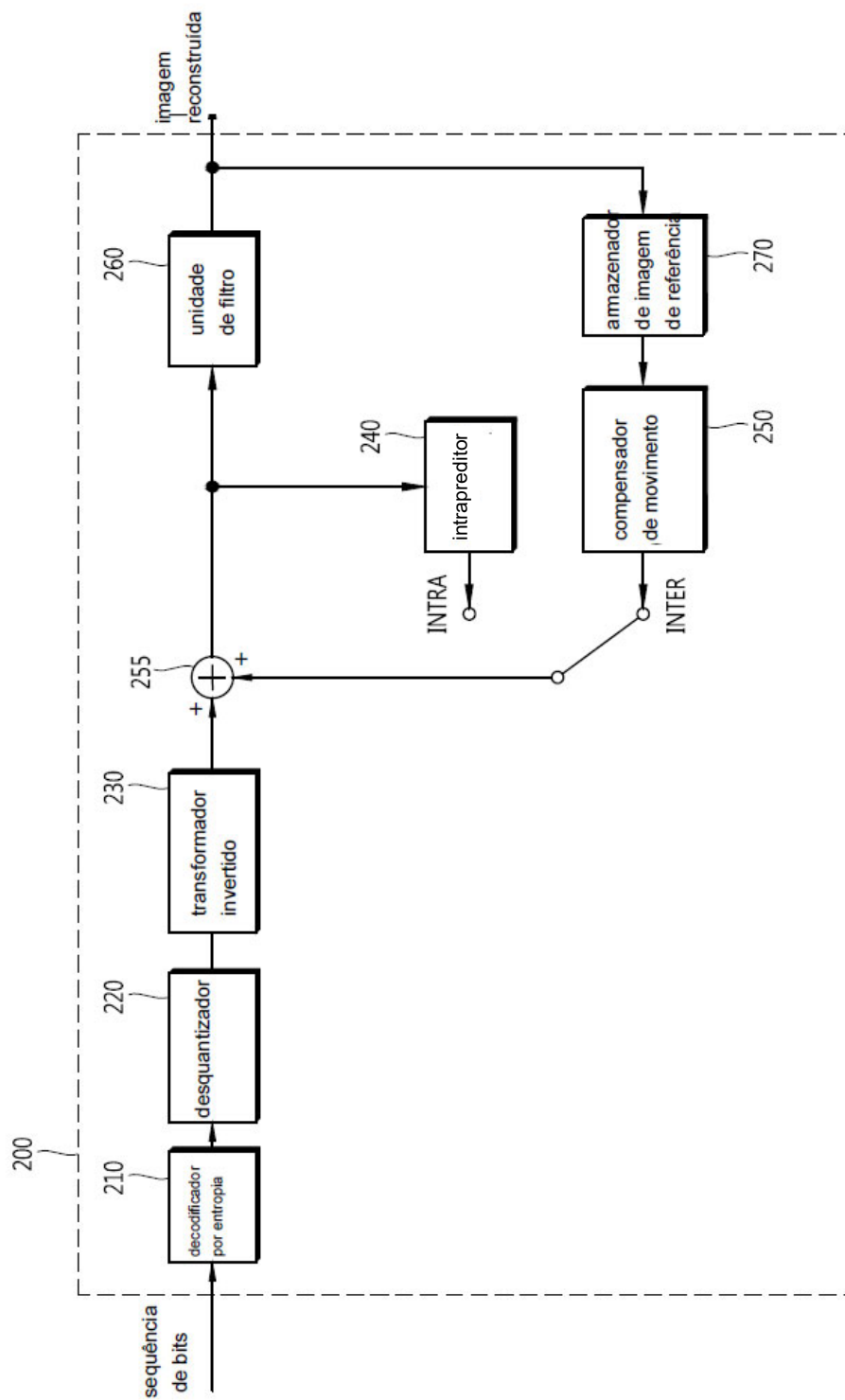


FIG. 3

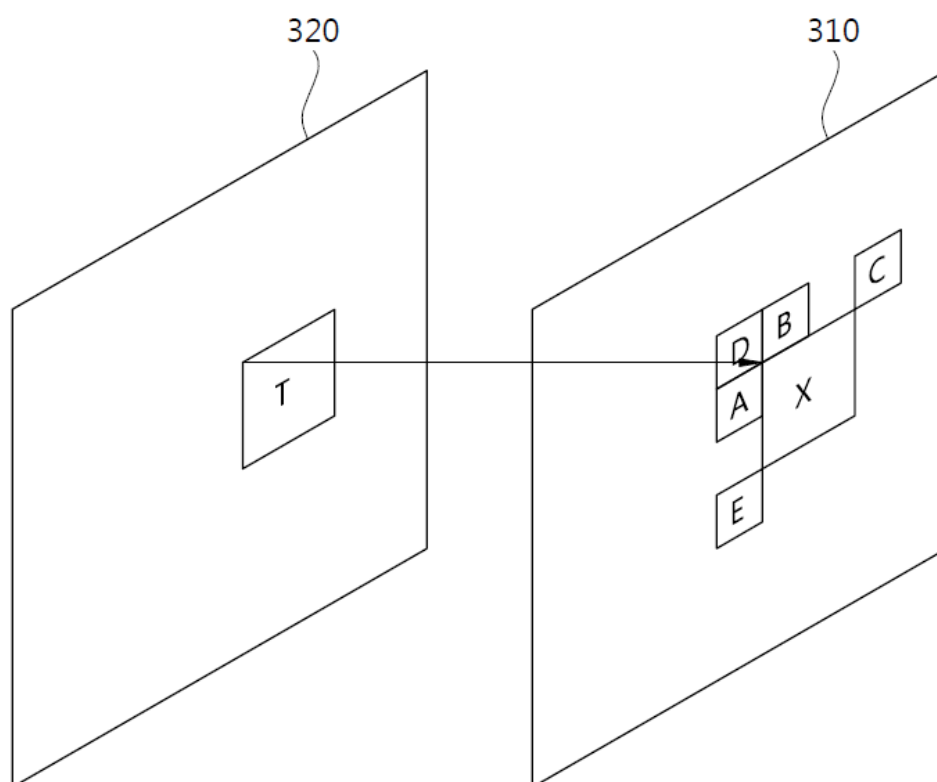


FIG. 4

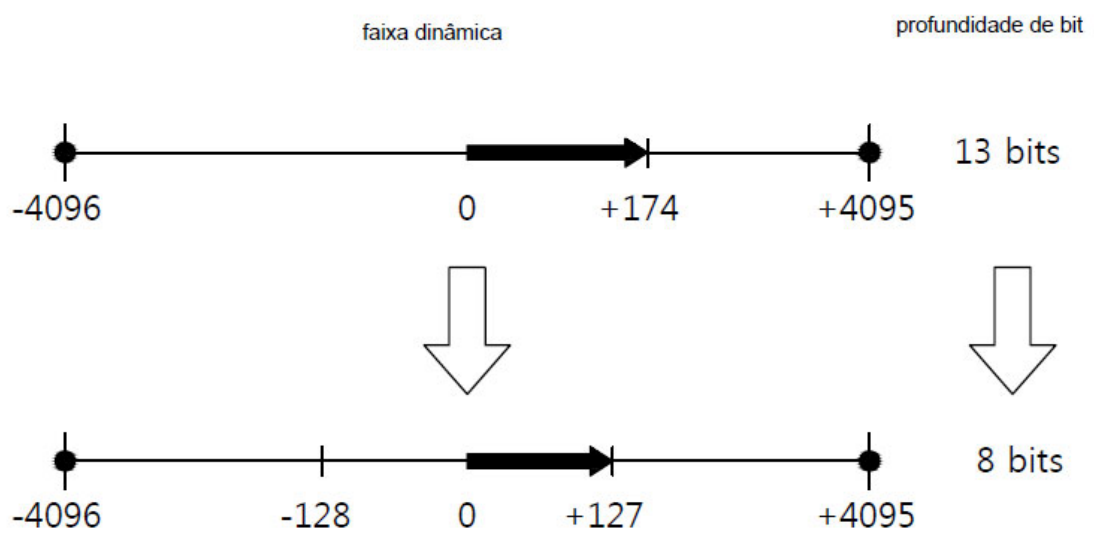


FIG. 5

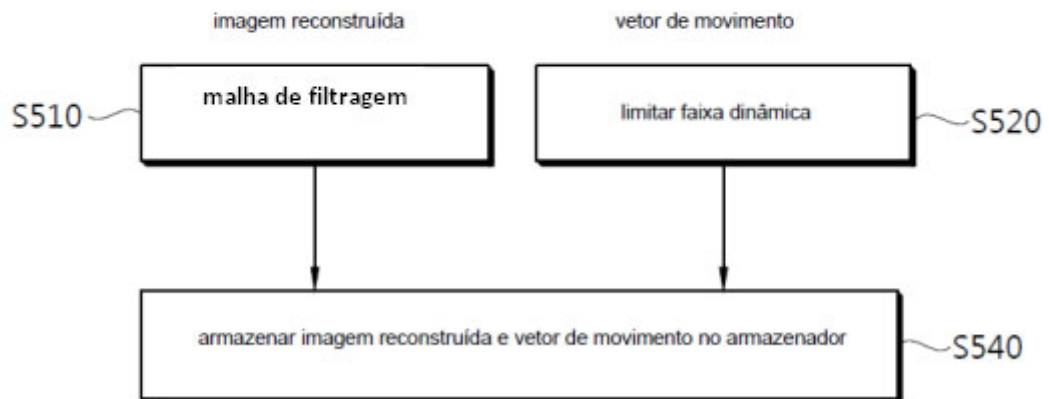


FIG. 6

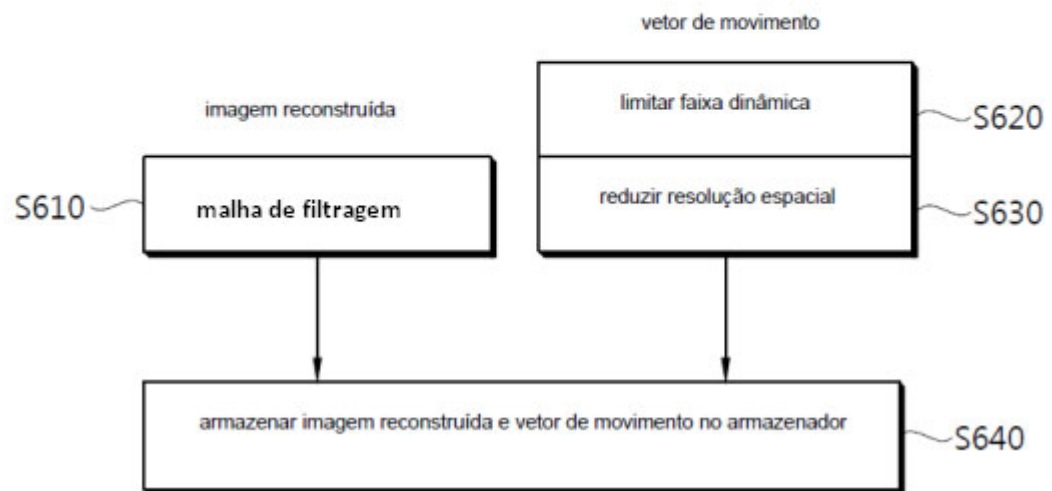


FIG. 7

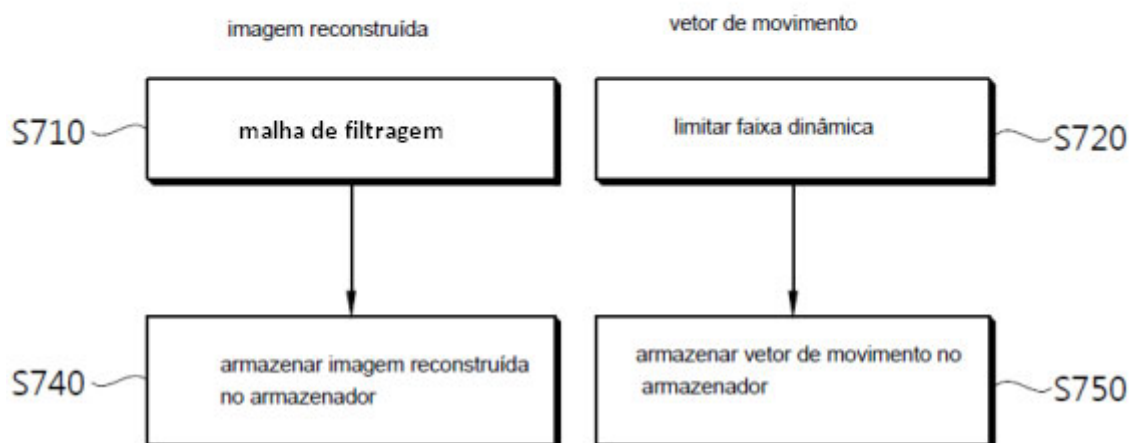


FIG. 8

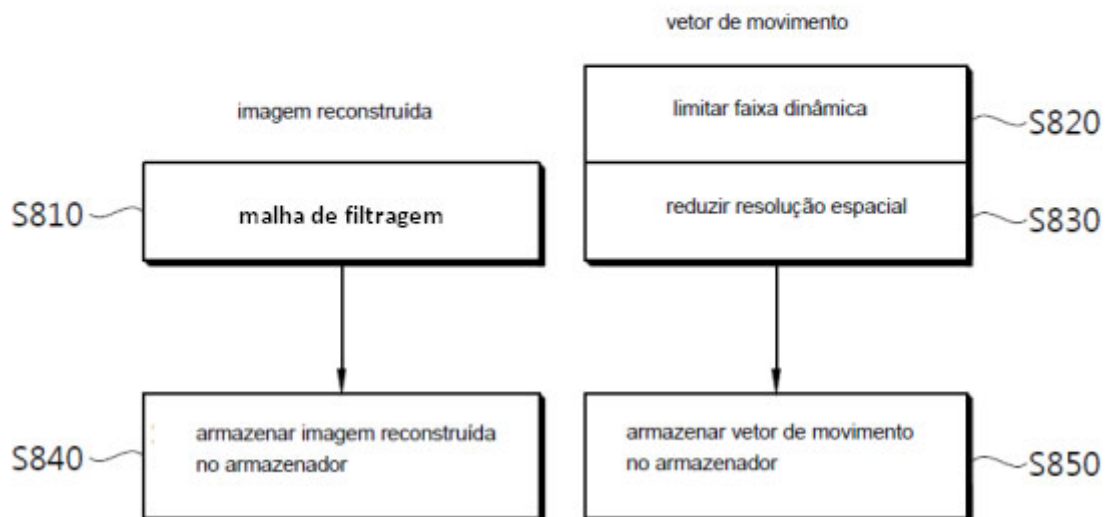


FIG. 9

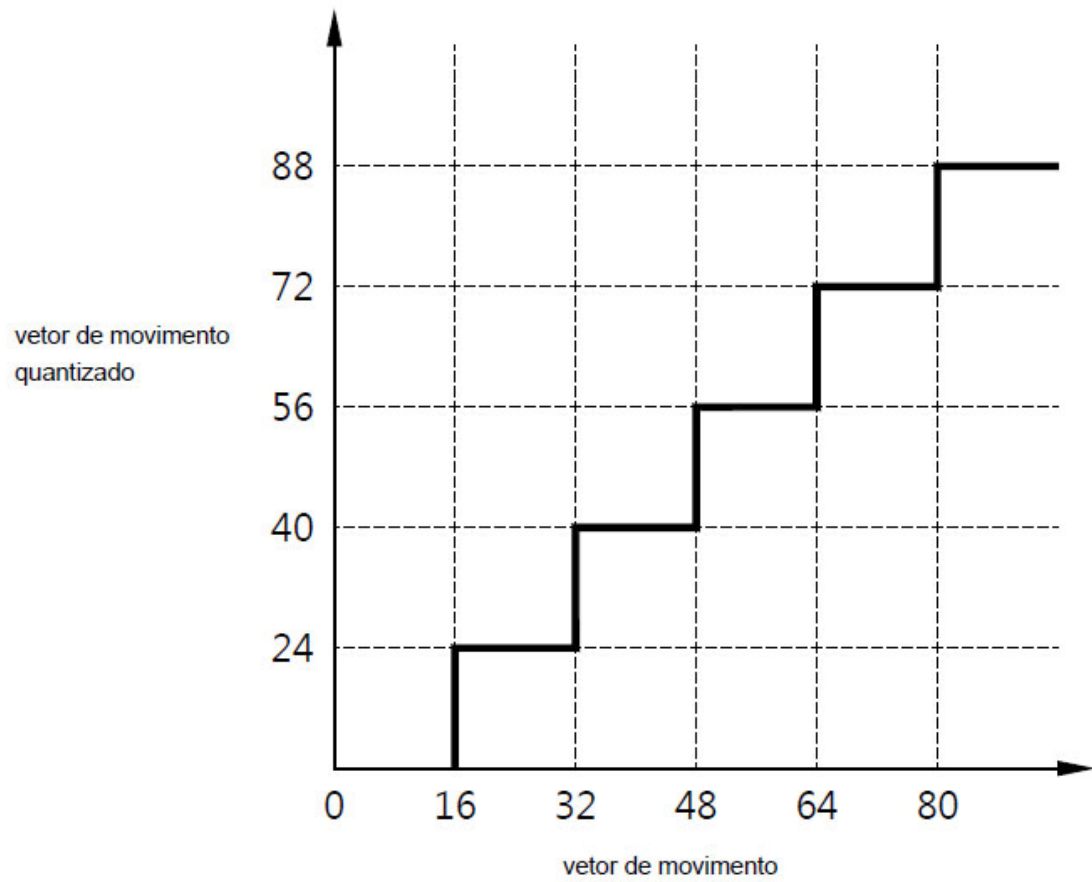


FIG. 10

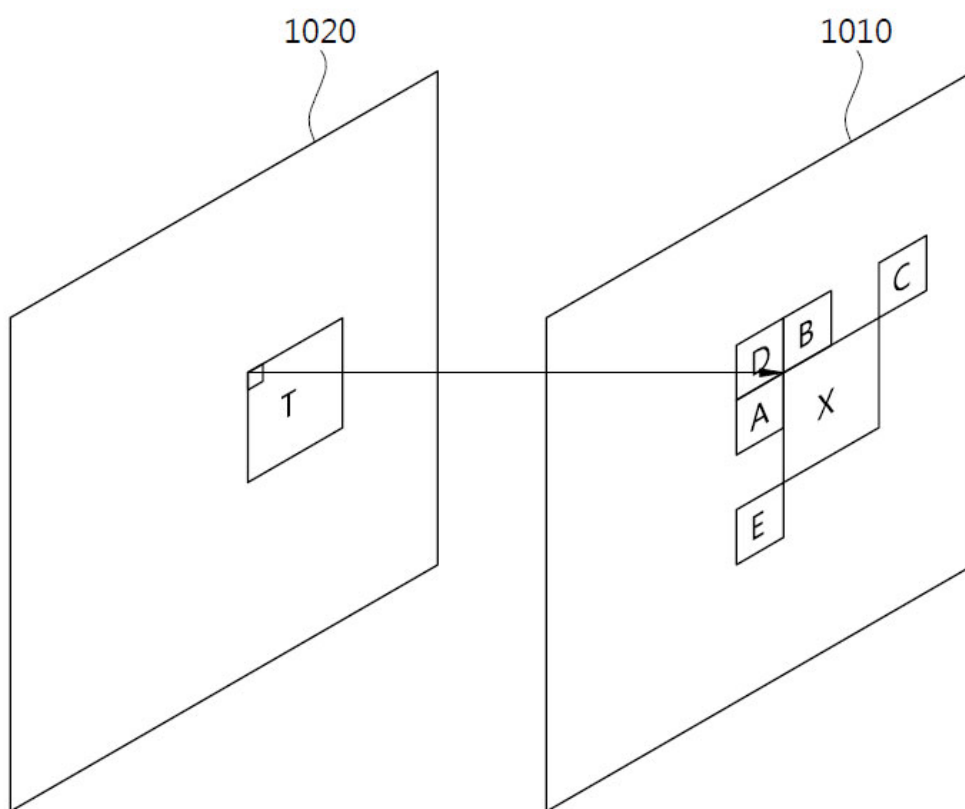


FIG. 11

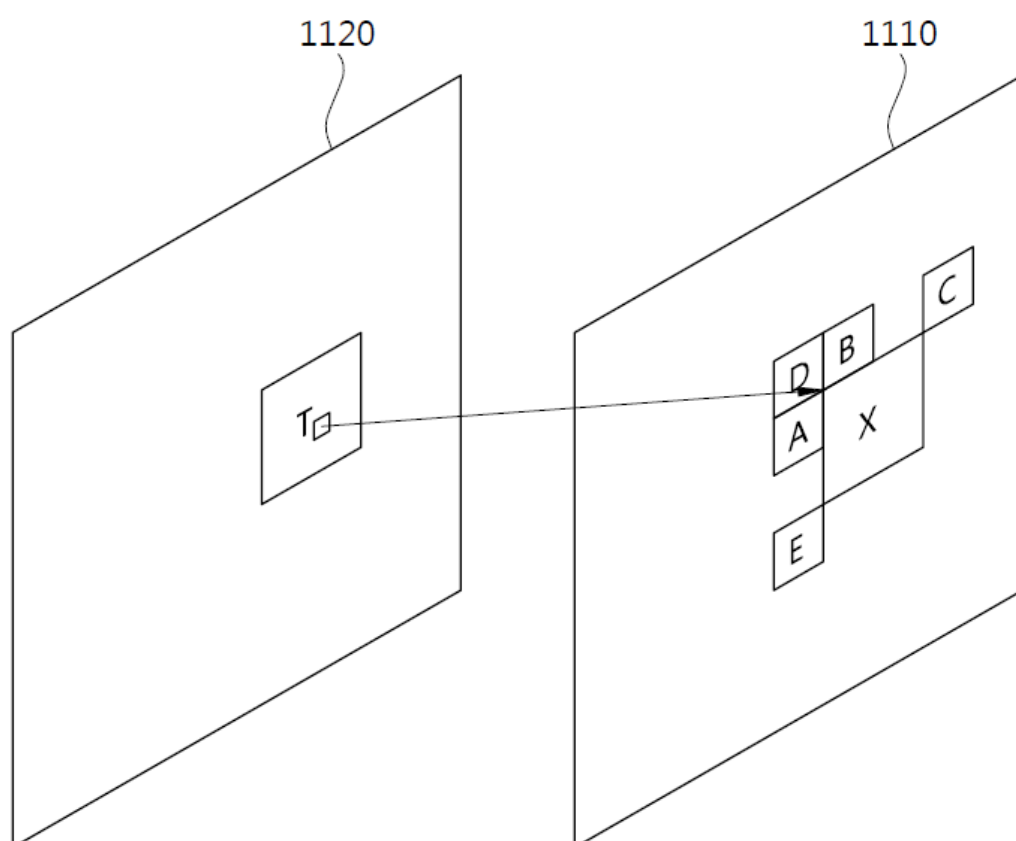


FIG. 12

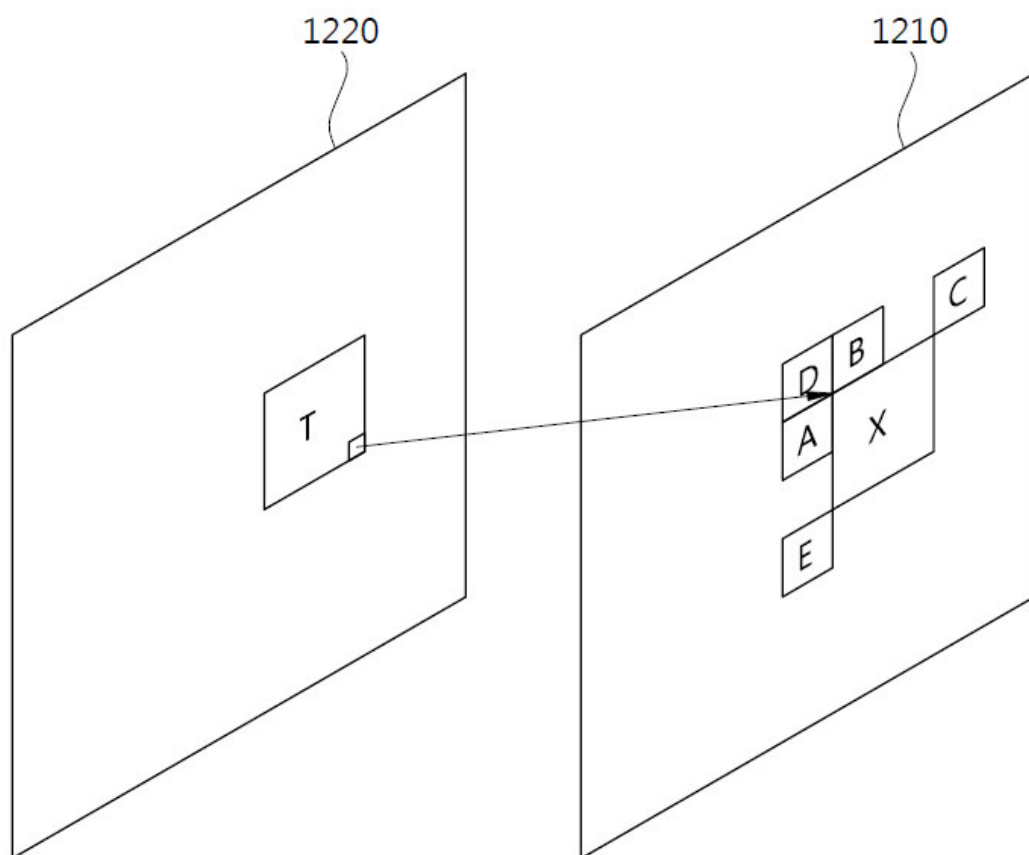


FIG. 13

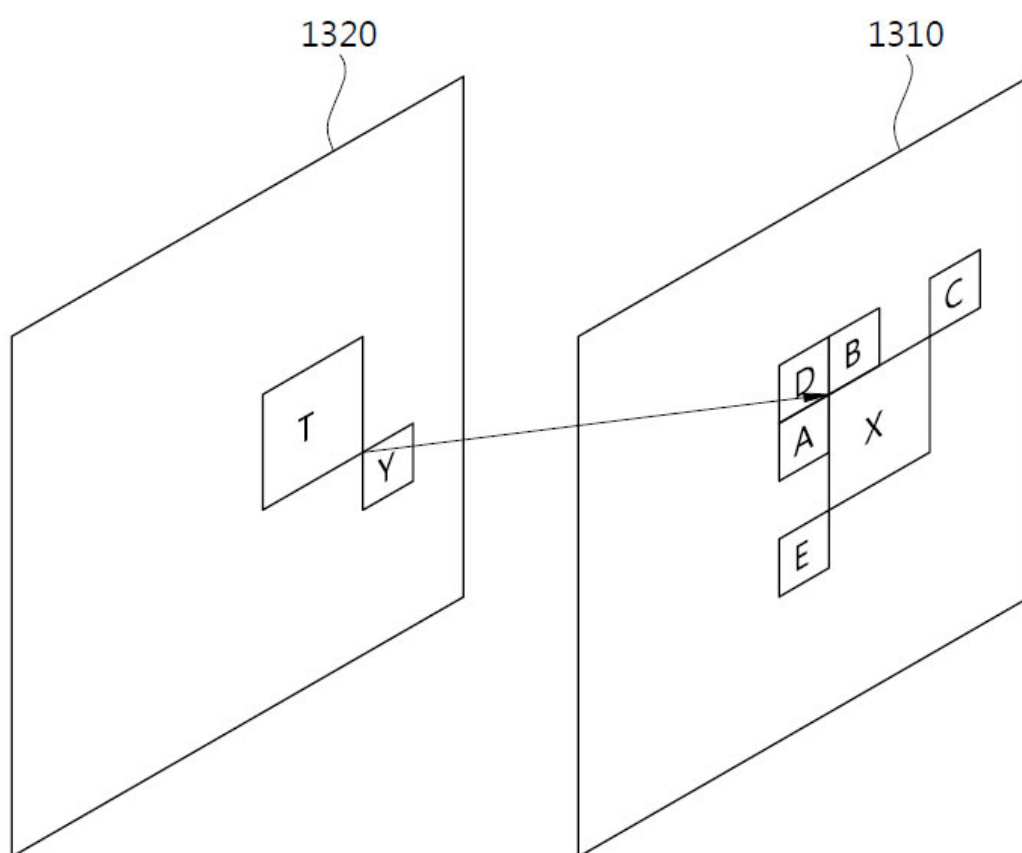


FIG. 14

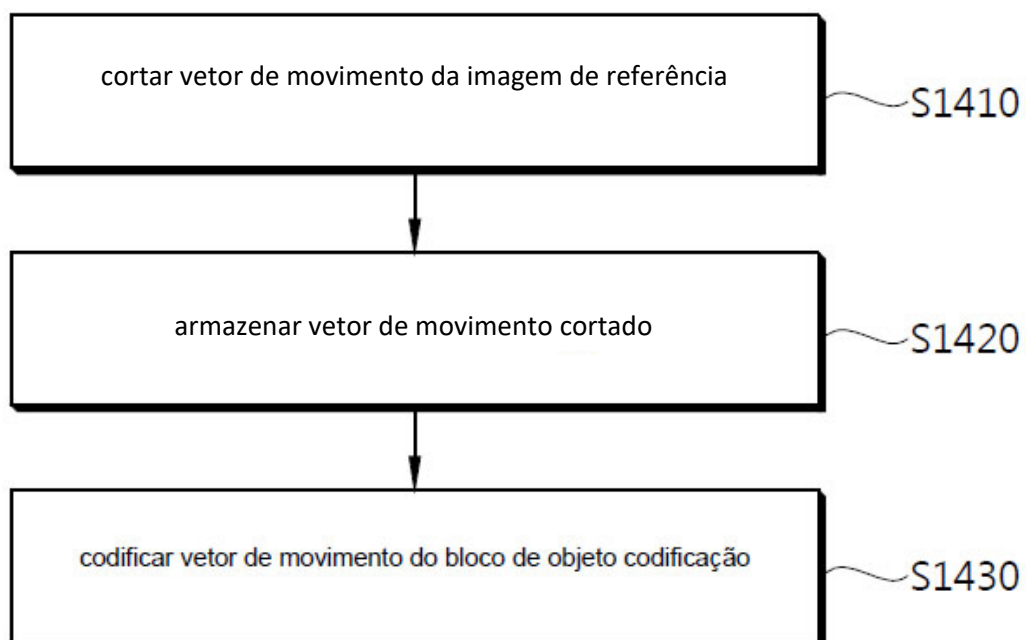


FIG. 15

