



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS
ESPAÑA



⑪ Número de publicación: **2 899 990**

⑮ Int. Cl.:

H04N 19/61 (2014.01)
H04N 19/44 (2014.01)
H04N 19/426 (2014.01)
H04N 19/513 (2014.01)
H04N 19/503 (2014.01)

⑫

TRADUCCIÓN DE PATENTE EUROPEA

T3

⑥ Fecha de presentación y número de la solicitud internacional: **31.01.2012** PCT/KR2012/000770
⑦ Fecha y número de publicación internacional: **09.08.2012** WO12105807
⑨ Fecha de presentación y número de la solicitud europea: **31.01.2012** E 12742304 (4)
⑩ Fecha y número de publicación de la concesión europea: **17.11.2021** EP 2672708

⑮ Título: **Método y aparato para codificar/decodificar imágenes usando un vector de movimiento**

⑩ Prioridad:

31.01.2011 KR 20110009636
03.03.2011 KR 20110019166
27.05.2011 KR 20110050853
01.07.2011 KR 20110065707

⑮ Fecha de publicación y mención en BOPI de la traducción de la patente:
15.03.2022

⑬ Titular/es:

**ELECTRONICS AND TELECOMMUNICATIONS
RESEARCH INSTITUTE (100.0%)**
**161 Gajeong-dong Yuseong-gu
Daejeon-si 305-700, KR**

⑭ Inventor/es:

LIM, SUNG CHANG;
KIM, HUI YONG;
JEONG, SE YOON;
CHO, SUK HEE;
JUN, DONG SAN;
KIM, JONG HO;
LEE, HA HYUN;
LEE, JIN HO;
CHOI, JIN SOO y
KIM, JIN WOONG

⑮ Agente/Representante:

ELZABURU, S.L.P

ES 2 899 990 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Método y aparato para codificar/decodificar imágenes usando un vector de movimiento

[Campo técnico]

5 La presente invención se refiere al procesamiento de imágenes y, más particularmente, a un método de codificación/decodificación de video que usa un vector de movimiento recortado y un aparato del mismo.

[Antecedentes de la técnica]

10 Recientemente, según la expansión de un sistema de difusión que soporta resolución de alta definición (HD) en el país y en todo el mundo, muchos usuarios se han acostumbrado a una imagen de alta resolución y definición, de modo que muchas organizaciones han realizado muchos intentos para desarrollar los dispositivos de video de próxima generación. Además, en la medida que ha aumentado el interés por que la HDTV y la ultra alta definición (UHD) que tienen una resolución cuatro veces superior a la de la HDTV, se ha demandado una tecnología de compresión para un video de mayor resolución y mayor definición.

15 Para la compresión de imágenes, se pueden usar una tecnología de interpredicción para predecir valores de píxeles incluidos en una imagen actual a partir de una imagen antes y/o después de la imagen actual, una tecnología de intrapredicción para predecir valores de píxeles utilizando información de píxeles en la imagen, una tecnología de codificación de entropía de asignación de un código corto para un símbolo que tiene una frecuencia de aparición alta y un código largo para un símbolo que tiene una frecuencia de aparición baja, o similar.

20 La codificación de imágenes en movimiento y audio se describe en "Text of ISO/IEC 14496-2 (MPEG-4 Visual) 2001 Edition", 57. REUNIONES DE MPEG 6-07-2001 - 20-07-2001; SYDNEY; (GRUPO DE EXPERTOS DE IMÁGENES EN MOVIMIENTO O ISO/IEC JTC1/SC29/WG11), n.º N4350, 30 de julio de 2001 (30-07-2001), ISSN: 0000-0369.

25 Más información con respecto al almacenamiento de datos de vectores de movimiento ha sido descrita por YEPING SU ET AL: "CE9: Reduced resolution storage of motion vector data", 4. REUNIÓN DE JCT-VC; 95. REUNIÓN DE MPEG; 20-1-2011 - 28-1-2011; DAEGU; (EQUIPO CONJUNTO COLABORATIVO SOBRE CODIFICACIÓN DE VÍDEO DE ISO/IEC JTC1/SC29/WG11 Y SG.16 DE LA UIT-T); URL: <HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/>, n.º JCTVC-D072, 16 de enero de 2011 (16-01-2011), ISSN: 0000-0015.

30 La decodificación de imágenes de vídeo también ha sido descrita por SULLIVAN: "Report of Awaji JVT Meeting (#6)", 6. REUNIÓN DE JVT; 63. REUNIÓN DE MPEG; 09-12-2002 - 13-12-2002; AWAJI, JP; (EQUIPO CONJUNTO DE VIDEO DE ISO/IEC JTC1/SC29/WG11 Y SG.16 DE LA ITU-T), n.º JVT-F001d1, 13 de diciembre de 2002 (13-12-2002), ISSN: 0000-0428.

30 [Divulgación]

[Problema técnico]

La presente invención proporciona un método de codificación/decodificación de vídeo que utiliza un vector de movimiento recortado y un aparato del mismo.

35 La presente invención también proporciona un método para recortar un vector de movimiento de una imagen de referencia.

[Solución técnica]

La presente invención está definida por las reivindicaciones independientes. Las posibles realizaciones se definen mediante las reivindicaciones dependientes.

[Efectos ventajosos]

40 Según la presente invención, el vídeo se codifica utilizando el vector de movimiento recortado.

Según algunos aspectos de la presente invención, se puede reducir el tamaño de la memoria requerida para almacenar vectores de movimiento.

Según algunos aspectos de la presente invención, se puede reducir el ancho de banda de acceso a la memoria requerido para traer datos de la memoria.

45 [Descripción de dibujos]

La FIG. 1 es un diagrama de bloques que muestra un ejemplo de una estructura de un codificador según la presente invención.

La FIG. 2 es un diagrama de bloques que muestra un ejemplo de una estructura de un decodificador según la presente invención.

La FIG. 3 muestra ejemplos de una imagen de objeto de codificación/decodificación y una imagen de referencia.

La FIG. 4 muestra un ejemplo de limitación de un rango dinámico de un vector de movimiento.

5 Las FIGS. 5 a 8 son diagramas de flujo que muestran un método de almacenamiento de un vector de movimiento de una imagen de referencia.

La FIG. 9 muestra un ejemplo de cuantificación de un vector de movimiento.

Las FIGS. 10 a 13 muestran ejemplos de obtención de información de movimiento de una imagen de referencia.

10 La FIG. 14 es un diagrama de flujo que muestra un método de codificación de una imagen según la presente invención.

La FIG. 15 es un diagrama de flujo que muestra un método de decodificación de una imagen según la presente invención.

[Modo para la invención]

15 A continuación, la presente invención se describirá en detalle con referencia a los dibujos adjuntos. Sin embargo, al describir ejemplos de la presente invención, no se describirán en detalle funciones o construcciones bien conocidas, dado que pueden oscurecer innecesariamente la comprensión de la presente invención.

20 Se ha de entender que cuando se hace referencia a cualquier elemento como "conectado a" o "acoplado a" otro elemento, puede estar conectado o acoplado directamente a otro elemento o estar conectado o acoplado a otro elemento, teniendo el otro elemento que interviene entre los mismos. Además, en la presente memoria descriptiva, en el caso de describir "incluir" un componente específico, se ha de entender que no se excluyen componentes adicionales distintos del componente correspondiente.

25 Los términos usados en la memoria descriptiva, "primer", "segundo", etc., se pueden usar para describir varios componentes, pero los componentes no se han de interpretar como que están limitados a los términos. Es decir, los términos se utilizan para distinguir un componente de otro componente. Por ejemplo, el "primer" componente se puede denominar el "segundo" componente, y viceversa, sin apartarse del alcance de la presente invención.

30 Además, los componentes descritos se muestran de forma independiente solo para indicar que realizan diferentes funciones características. Por lo tanto, los componentes que se muestran de forma independiente no significan que cada uno de los componentes no se pueda implementar como un solo hardware o software. Es decir, cada uno de los componentes se divide por conveniencia de la explicación, una pluralidad de componentes se pueden combinar entre sí para operar de ese modo como un componente o un componente se puede dividir en una pluralidad de componentes para operar de este modo como la pluralidad de componentes, que se incluyen en el alcance de la presente invención siempre que se aparte de las características esenciales de la presente invención.

35 Además, algunos de los componentes pueden no ser componentes indispensables que realicen funciones esenciales de la presente invención, sino que son componentes selectivos que mejoran únicamente su rendimiento. La presente invención también se puede implementar solo mediante una estructura que incluye los componentes indispensables excepto los componentes selectivos, y la estructura que incluye solo los componentes indispensables también se incluye en el alcance de la presente invención.

La FIG. 1 es un diagrama de bloques que muestra un ejemplo de una estructura de un codificador según la presente invención.

40 Haciendo referencia a la FIG. 1, el codificador 100 incluye un predictor de movimiento 111, un compensador de movimiento 112, un intrapredictor 120, un conmutador 115, un sustractor 125, un transformador 130, un cuantificador 140, un codificador de entropía 150, un descuantificador 160, un transformador inverso 170, un sumador 175, una unidad de filtro 180 y una memoria intermedia de imágenes de referencia 190.

45 El codificador 100 codifica imágenes de entrada en un modo de intrapredicción o en un modo de interpredicción para que el codificador emita un flujo de bits. La intrapredicción significa predicción intraimagen y la interpredicción significa predicción entre imágenes. El codificador 100 se conmuta entre el modo de intrapredicción y el modo de interpredicción a través de la comutación del conmutador 115. El codificador 100 genera un bloque predicho para un bloque de entrada de la imagen de entrada y luego codifica un residuo entre el bloque de entrada y el bloque predicho.

50 En el caso del modo de intrapredicción, el intrapredictor 120 realiza la predicción espacial utilizando valores de píxeles de bloques vecinos que ya están codificados para generar bloques predichos.

En el caso del modo de interpredicción, el predictor de movimiento 111 busca un bloque de referencia emparejado de manera óptima con el bloque de entrada en una imagen de referencia almacenada en la memoria intermedia de imágenes de referencia 190 durante un proceso de predicción de movimiento para obtener un vector de movimiento. El compensador de movimiento 112 realiza una compensación de movimiento usando el vector de movimiento para 5 generar el bloque predicho. Aquí, el vector de movimiento puede ser un vector bidimensional utilizado para la interpredicción y representar un desplazamiento entre un bloque de árbol de codificación actual y el bloque de referencia.

El sustractor 125 genera un bloque residual en base al residual entre el bloque de entrada y el bloque predicho, y el transformador 130 transforma el bloque residual para emitir un coeficiente de transformación. El cuantificador 140 10 cuantifica el coeficiente de transformación para emitir el coeficiente cuantificado.

El codificador de entropía 150 realiza la codificación de entropía en base a la información obtenida durante un proceso de codificación/cuantificación para emitir el flujo de bits. La codificación de entropía representa símbolos generados con frecuencia como un número pequeño de bits, reduciendo de este modo el tamaño de un flujo de bits para un símbolo de codificación. Por lo tanto, se puede esperar que el rendimiento de compresión de un video se 15 mejore a través de la codificación de entropía. El codificador de entropía 150 puede usar un método de codificación tal como golomb exponencial, codificación de longitud variable adaptativa al contexto (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC), o similares, para la codificación de entropía.

Una imagen codificada necesita ser decodificada y almacenada de nuevo para ser usada como imagen de referencia para realizar la codificación interpredicción. Por lo tanto, el descuantificador 160 descuantifica el 20 coeficiente cuantificado y el transformador inverso 170 transforma inversamente el coeficiente descuantificado para emitir un bloque residual reconstruido. El sumador 175 añade el bloque residual reconstruido al bloque predicho para generar un bloque reconstruido.

La unidad de filtro 180 también se denomina filtro en bucle adaptativo y aplica al menos uno de filtrado de desbloqueo, compensación de desplazamiento adaptativo de muestra (SAO), filtrado de bucle adaptativo (ALF) al 25 bloque reconstruido. El filtrado de desbloqueo significa que se elimina la distorsión de bloque ocurrida en un límite entre bloques, y la compensación de SAO significa que se agrega un desplazamiento apropiado a un valor de píxel para compensar un error de codificación. Además, el ALF significa que el filtrado se realiza en base a un valor de comparación entre una imagen reconstruida y una imagen original.

Mientras tanto, la memoria intermedia de imágenes de referencia 190 almacena el bloque reconstruido que pasa a 30 través de la unidad de filtro 180 en su interior.

La FIG. 2 es un diagrama de bloques que muestra un ejemplo de una estructura de un decodificador según la presente invención.

Haciendo referencia a la FIG. 2, un decodificador incluye un decodificador de entropía 210, un descuantificador 220, 35 un transformador inverso 230, un intrapredictor 240, un compensador de movimiento 250, un sumador 255, una unidad de filtro 260 y una memoria intermedia de imágenes de referencia 270.

El decodificador 200 decodifica el flujo de bits en el modo de intrapredicción o en el modo de interpredicción para emitir una imagen reconstruida. El decodificador 200 se comunica entre el modo de intrapredicción y el modo de interpredicción a través de la comutación del conmutador. El decodificador 200 obtiene un bloque residual del flujo de bits para generar un bloque predicho y luego añade el bloque residual y el bloque predicho entre sí para generar 40 un bloque reconstruido.

El decodificador de entropía 210 realiza la decodificación de entropía en base a la distribución de probabilidad. El proceso de decodificación de entropía es un proceso opuesto al proceso de codificación de entropía mencionado anteriormente. Es decir, el decodificador de entropía 210 genera un símbolo que incluye un coeficiente cuantificado del flujo de bits en donde un símbolo generado con frecuencia se representa como un pequeño número de bits.

45 El descuantificador 220 descuantifica el coeficiente cuantificado y el transformador inverso 230 transforma inversamente el coeficiente descuantificado para generar un bloque residual.

En el caso del modo de intrapredicción, el intrapredictor 240 realiza una predicción espacial utilizando valores de píxeles de bloques vecinos que ya están codificados para generar bloques predichos.

50 En el caso del modo de interpredicción, el compensador de movimiento 250 realiza la compensación de movimiento utilizando el vector de movimiento y la imagen de referencia almacenada en la memoria intermedia de imágenes de referencia 270 para generar el bloque predicho.

El sumador 255 agrega el bloque predicho al bloque residual, y la unidad de filtro 260 aplica al menos uno de filtrado de desbloqueo, compensación de SAO, ALF al bloque que pasa a través del sumador para emitir una imagen reconstruida.

La imagen reconstruida se puede almacenar en la memoria intermedia de imágenes de referencia 270 para ser usada de ese modo para la compensación de movimiento.

En lo sucesivo, un bloque significa una unidad de codificación/decodificación. En un proceso de codificación/decodificación, una imagen se divide en un tamaño predeterminado y luego se codifica/decodifica. Por lo tanto, un bloque también se puede denominar unidad de codificación (CU), unidad de predicción (PU), unidad de transformación (TU) o similar, y un solo bloque también se puede dividir en subbloques que tienen un tamaño más pequeño.

Aquí, una unidad de predicción significa una unidad básica en la que se realiza la predicción y/o compensación de movimiento. Una unidad de predicción se puede dividir en una pluralidad de particiones, y cada una de las particiones también se puede denominar partición de unidad de predicción. Cuando una unidad de predicción se divide en la pluralidad de particiones, cada una de las particiones de la unidad de predicción puede convertirse en una unidad básica en la que se realizan la predicción y/o compensación de movimiento. En lo sucesivo, en los ejemplos de la presente invención, una unidad de predicción también puede significar particiones de la unidad de predicción.

Mientras tanto, en la codificación de video de alta eficiencia (HEVC), se utiliza un método de predicción de vector de movimiento en base a la predicción de vector de movimiento avanzada (AMVP).

En el método de predicción de vectores de movimiento en base a la predicción de vectores de movimiento avanzada, un vector de movimiento (MV) de un bloque, que existe en una posición que es igual o que corresponde a la de un bloque de árbol de codificación, en una imagen de referencia, así como en vectores de movimiento de bloques reconstruidos colocados alrededor del bloque de árbol de codificación. Aquí, el bloque, que existe en una posición que es la misma o que corresponde espacialmente a la del bloque de árbol de codificación, en la imagen de referencia se llama bloque cúbicado, y un vector de movimiento del bloque cúbicado se llama vector de movimiento cúbicado o vector de movimiento temporal. Sin embargo, el bloque cúbicado puede ser un bloque, que existe en una posición similar a (es decir, correspondiente a) la del bloque de árbol de codificación, en la imagen de referencia, así como un bloque que existe en la misma posición que la del bloque de árbol de codificación.

En un método de combinación de información de movimiento, la información de movimiento se estima a partir del bloque cúbicado junto con los bloques reconstruidos colocados alrededor del bloque de árbol de codificación para ser usada de ese modo como información de movimiento del bloque de árbol de codificación. Aquí, la información de movimiento incluye al menos una de la información del modo de interpredicción que indica un índice de imagen de referencia, un vector de movimiento, una dirección en un sentido, una dirección en dos sentidos o similar, requerida en el momento de la interpredicción, una lista de imágenes de referencia e información del modo de predicción sobre si la codificación se realiza en un modo de intrapredicción o en un modo de interpredicción.

Un vector de movimiento predicho en el bloque de árbol de codificación puede ser un vector de movimiento del bloque cúbicado, que es un bloque temporalmente adyacente al bloque de árbol de codificación, así como vectores de movimiento de bloques vecinos adyacentes espacialmente al bloque de árbol de codificación.

La FIG. 3 muestra ejemplos de una imagen de objeto de codificación/decodificación y una imagen de referencia.

En la FIG. 3, un bloque X indica un bloque de árbol de codificación en una imagen de objeto de codificación/decodificación 310, y un bloque A, un bloque B, un bloque C, un bloque D y un bloque E indican bloques reconstruidos colocados alrededor del bloque de árbol de codificación. Además, un bloque T en la imagen de referencia 320 indica un bloque cúbicado que existe en una posición correspondiente a la del bloque de árbol de codificación.

Qué vector de movimiento en el bloque de árbol de codificación se usa como el vector de movimiento predicho se puede reconocer a través de un índice de predictor de vector de movimiento.

[Tabla 1]

<code>prediction_unit (x0, y0, log2PUWidth, log2PUHeight) {</code>	Descriptor
<code> si(skip_flag [x0] [y0]) {</code>	
<code> si(NumMVPCandidate(L0) > 1)</code>	
<code> mvp_idx_l0 [x0][y0]</code>	<code>ue(v) ae(v)</code>
<code> si(NumMVPCandidate(L1) > 1)</code>	

mvp_idx_l1 [x0][y0]	ue(v) ae(v)
} de otro modo si (PredMode == MODE_INTRA) {	
...	
} de otro modo {/* MODE_MERGE, MODE_INTER */	
si(merge_flag [x0] [y0] && NumMergeCandidates > 1) {	
...	
} de otro modo {	
si(inter_pred_idc [x0] [y0] != Pred_L1) {	
si(NumMVPCand(L0) > 1)	
mvp_idx_l0	ue(v) ae(v)
}	
si(inter_pred_idc [x0] [y0] != Pred_L0) {	
si(NumMVPCand(L1) > 1)	
mvp_idx_l1	ue(v) ae(v)
}	
}	
}	
}	
}	

Como se muestra en la Tabla 1, los índices de predictor del vector de movimiento **mvp_idx_l0** y **mvp_idx_l1** para cada lista de imágenes de referencia se transmiten a un decodificador, y el decodificador usa el mismo vector de movimiento que un vector de movimiento predicho por un codificador como vector de movimiento predicho.

En el caso en el que el bloque de árbol de codificación se codifique/decodifique usando los vectores de movimiento de los bloques vecinos espacialmente adyacentes al bloque de árbol de codificación, el vector de movimiento se puede almacenar solo con una memoria que tenga un tamaño relativamente pequeño. Sin embargo, en el caso en el que se use un vector de movimiento temporal, dado que todos los vectores de movimiento de la imagen de referencia necesitan ser almacenados en una memoria, se requiere una memoria que tenga un tamaño relativamente grande y también aumenta el tamaño de ancho de banda de acceso a la memoria requerido para traer datos de la memoria. Por lo tanto, existe la necesidad de almacenar de manera más eficiente el vector de movimiento temporal en un entorno de aplicación en el que el espacio de memoria de un terminal portátil, o similar, no es suficiente o se minimiza el consumo de energía.

Mientras tanto, como tecnología para almacenar un vector de movimiento en una memoria, existe un método de reducción de la resolución espacial del vector de movimiento. En este método, el vector de movimiento se comprime en cualquier proporción y luego se almacena en la memoria. Por ejemplo, un vector de movimiento almacenado en una unidad de bloques de 4x4 se almacena en una unidad de bloques de 4x4 o más para reducir el número de vectores de movimiento almacenados. Aquí, para ajustar un tamaño de bloque del vector de movimiento almacenado, se transmite información sobre una relación de compresión. La información se transmite a través de un conjunto de parámetros de secuencia (SPS) como se muestra en la Tabla 2.

[Tabla 2]

	C	Descriptor
seq_parameter_set_rbsp () {		
.....		
motion_vector_buffer_comp_flag	0	u(1)
si (motion_vector_buffer_comp_flag)		
motion_vector_buffer_comp_ratio_log2	0	u(8)
rbsp_trailing_bits ()	0	
}		

Haciendo referencia a la Tabla 2, en el caso en el que **motion_vector_buffer_comp_flag** es 1, se realiza un proceso de compresión de memoria intermedia de vector de movimiento.

motion_vector_buffer_comp_ratio_log2 indica una relación de compresión del proceso de compresión de memoria intermedia de vector de movimiento. En el caso de que la **motion_vector_buffer_comp_ratio_log2** no exista,

- 5 motion_vector_buffer_comp_ratio_log2 se estima en 0, y la relación de compresión de memoria intermedia de vector de movimiento se representa mediante la Ecuación 1.

[Ecuación 1]

$$MVBufferCompRatio = 1 \ll motion_vector_buffer_comp_ratio_log2$$

10 Por ejemplo, en el caso en donde todos los bloques 4x4 de imágenes de 1920 x 1080 tienen diferentes vectores de movimiento y usan dos listas de imágenes de referencia, cada una usando dos imágenes de referencia, se requiere un espacio de memoria total de 3,21 Mbytes para almacenar un vector de movimiento temporal como se describe a continuación.

1. Profundidad de bits de 26 bits por vector de movimiento

- 15 (1) Rango dinámico del componente X del vector de movimiento: -252 a +7676 (profundidad de bits: 13 bits)
 (2) Rango dinámico del componente Y del vector de movimiento: -252 a +4316 (profundidad de bits: 13 bits)
 (3) (Los rangos dinámicos de cada componente del vector de movimiento se calcularon en base a una primera unidad de predicción en una imagen correspondiente).

- 20 2. En el caso de que todas las unidades de bloques 4x4 tengan diferentes vectores de movimiento: 480 x 270 = 129600 bloques
 3. Uso de dos vectores de movimiento por cada bloque
 4. El número de listas de imágenes de referencia: 2
 5. Uso de dos imágenes de referencia por lista de imágenes de referencia

25 $=> 26 \text{ bits} \times 129600 \text{ bloques} \times 2 \text{ vectores de movimiento} \times 2 \text{ listas de imágenes de referencia} \times 2 \text{ imágenes de referencia} = 26956800 \text{ bits} = 3,21 \text{ Mbytes}$

30 Según el método de reducción de una resolución espacial de un vector de movimiento como se describió anteriormente, es posible reducir el tamaño del espacio de memoria requerido y el ancho de banda de acceso a la memoria usando la correlación espacial del vector de movimiento. Sin embargo, el método de reducción de la resolución espacial de un vector de movimiento no limita el rango dinámico del vector de movimiento.

35 Cuando el tamaño del espacio de memoria se reduce a 1/4, el tamaño del espacio de memoria requerido en el ejemplo mencionado anteriormente se reduce a aproximadamente 0,8 Mbytes. Aquí, cuando solo se usan seis bits de la profundidad de bits requerida para almacenar el vector de movimiento para cada componente del vector de movimiento limitando adicionalmente el rango dinámico del vector de movimiento, el tamaño del espacio de memoria requerido se puede reducir aún más a 0,37 Mbytes.

Por lo tanto, en la presente invención, el rango dinámico del vector de movimiento se limita para reducir el tamaño de un espacio de memoria requerido para almacenar el vector de movimiento y un ancho de banda de acceso a la memoria requerido para traer datos de la memoria. El vector de movimiento de la imagen de referencia cuyo rango dinámico está limitado se puede usar como vector de movimiento temporal en el bloque de árbol de codificación.

- 5 En lo sucesivo, un rango dinámico significa un rango entre un valor mínimo y un valor máximo de un componente negativo o un componente positivo de un vector de movimiento en base a 0, y una profundidad de bits, que indica un tamaño de espacio requerido para almacenar el vector de movimiento, significa una anchura de bits. Además, a menos que se describa en particular, el vector de movimiento significa un vector de movimiento de una imagen de referencia, es decir, un vector de movimiento temporal.
- 10 En el caso de que cada componente del vector de movimiento esté fuera del rango dinámico, se representa por el valor mínimo o el valor máximo del rango dinámico correspondiente. Por ejemplo, en el caso en el que un componente X del vector de movimiento 312 y un valor máximo de un rango dinámico de cada componente del vector de movimiento 256, el componente X del vector de movimiento está limitado a 256.
- 15 Asimismo, en el caso en el que una profundidad de bits de cada componente del vector de movimiento sea de 16 bits y el vector de movimiento sea (-36, 24), cuando la profundidad de bits de cada componente del vector de movimiento se limita a 6 bits, cada componente del vector de movimiento tiene un rango dinámico de -32 a +31, de modo que el vector de movimiento se representa por (-32, 24), que está en su rango dinámico.
- 20 Además, en el caso en el que una profundidad de bits de cada componente del vector de movimiento sea de 16 bits y el vector de movimiento sea (-49, 142), cuando la profundidad de bits de cada componente del vector de movimiento se limita a 9 bits, cada componente del vector de movimiento tiene un rango dinámico de -256 a +255, de modo que el vector de movimiento se representa por (-49, 142) sin cambios.

La FIG. 4 muestra un ejemplo de limitación de un rango dinámico de un vector de movimiento.

Haciendo referencia a la FIG. 4, cuando un rango dinámico de un vector de movimiento que tiene un rango dinámico de -4096 a +4095 se limita a -128 a +127, la profundidad de bits se puede reducir de 13 bits a 8 bits.

- 25 Cada componente de un vector de movimiento temporal se recorta como se representa por las Ecuaciones 2 y 3 para ser almacenado en una profundidad de bits de N bits. Donde N indica un número entero positivo.

[Ecuación 2]

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

[Ecuación 3]

$$30 \quad clippedMV_Y = \min(1 \ll (N-1)-1, \max(-1 \ll (N-1), MV_Y))$$

Donde MV_X indica un componente X del vector de movimiento, MV_Y indica un componente Y del vector de movimiento, $\min(a, b)$ significa una operación de emisión de un valor más pequeño en a y b, y $\max(a, b)$ significa una operación de emisión de un valor mayor en a y b. Cada uno de clippedMV_X y clippedMV_Y indica componentes X e Y del vector de movimiento temporal recortado y se almacena en la memoria para ser utilizado de ese modo como un vector de movimiento temporal del bloque de árbol de codificación.

35 Por ejemplo, como se muestra en la Tabla 3, en el caso en el que el tamaño de un espacio de memoria sea de 48 bytes y cada componente del vector de movimiento use una profundidad de bits de 16 bits, se pueden almacenar un total de doce vectores de movimiento.

[Tabla 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

- 40 Sin embargo, cuando cada componente del vector de movimiento usa solo una profundidad de bits de 8 bits, se pueden almacenar un total de veinticuatro vectores de movimiento como se muestra en la Tabla 4.

[Tabla 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
MV13-X	MV13-Y	MV14-X	MV14-Y	MV15-X	MV15-Y	MV16-X	MV16-Y
MV17-X	MV17-Y	MV18-X	MV18-Y	MV19-X	MV19-Y	MV20-X	MV20-Y
MV21-X	MV21-Y	MV22-X	MV22-Y	MV23-X	MV23-Y	MV24-X	MV24-Y

Por lo tanto, según la presente invención, cuando una imagen reconstruida en un codificador y/o decodificador se somete a un proceso de filtrado en bucle, tal como un filtro de desbloqueo, un filtro de bucle adaptativo o similar, y luego se almacena en una memoria intermedia de imágenes decodificadas (DPB), el rango dinámico del vector de movimiento es limitado, de modo que se almacene un vector de movimiento de una imagen de referencia. La memoria intermedia de imágenes decodificadas significa la memoria intermedia de imágenes de referencia de la FIG. 1 o la FIG. 2.

5 I. Proceso de recorte de vector de movimiento

Se invoca un proceso de recorte de cada componente de un vector de movimiento en el caso en el que un slice_type no es igual a I. El proceso de recorte de un vector de movimiento se realiza en un bloque de árbol o unidad de codificación más grande (LCU) después de que se termina un proceso de filtrado.

10 Las entradas en el proceso de recorte de un vector de movimiento son una ubicación (xP, yP) que especifica la muestra superior izquierda de la unidad de predicción con relación a la muestra superior izquierda de la imagen actual y las matrices de vectores de movimiento MvL0 y MvL1. Las salidas del proceso son las matrices de vectores de movimiento recortadas CMvL0 y CMvL1.

15 Con respecto a las matrices MvL0, MvL1, CMvL0 y CMvL1, se realizan las operaciones de las Ecuaciones 4 a 7.

[Ecuación 4]

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

[Ecuación 5]

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

20 [Ecuación 6]

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

[Ecuación 7]

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

25 Donde TMVBitWidth indica una profundidad de bits de un vector de movimiento, Clip3(a, b, c) significa una función de recorte c para existir en un rango entre a y b.

II. Proceso de almacenamiento de vector de movimiento.

Las FIGS. 5 a 8 son diagramas de flujo que muestran un método de almacenamiento de un vector de movimiento de una imagen de referencia.

30 Haciendo referencia a la FIG. 5, el vector de movimiento de la imagen de referencia se puede almacenar utilizando tanto una memoria intermedia de imágenes que almacena una imagen reconstruida como una memoria intermedia de vectores de movimiento que almacena un vector de movimiento. Aquí, la imagen reconstruida se somete a un proceso de filtrado en bucle (S510) y el vector de movimiento se somete a un proceso de limitación de rango dinámico (S520) y luego se almacena (S540).

Además, con referencia a la FIG. 6, se utilizan tanto una memoria intermedia de imágenes como un memoria intermedia de vectores de movimiento, y el vector de movimiento se somete a un proceso de limitación de rango dinámico (S620) y un proceso de reducción de resolución espacial (S630) y luego se almacena (S640).

5 Además, haciendo referencia a la FIG. 7, la imagen reconstruida se somete a un proceso de filtrado en bucle (S710) y luego se almacena en una memoria intermedia de imágenes (S740), y el vector de movimiento se somete a un proceso de limitación de rango dinámico (S720) y luego se almacena en una memoria intermedia de vectores de movimiento (S750).

10 Además, con referencia a la FIG. 8, la imagen reconstruida se somete a un proceso de filtrado en bucle (S810) y luego se almacena en un memoria intermedia de imágenes (S840), y el vector de movimiento se somete a un proceso de limitación de rango dinámico (S820) y un proceso de reducción de resolución espacial (S830) y luego se almacena (S850).

15 Mientras tanto, en las FIGS. 6 y 8, una secuencia del proceso de limitación de rango dinámico S620 o S820 y el proceso de reducción de resolución espacial S630 y S830 no está limitada, pero se puede cambiar.

20 Además, para reducir aún más el ancho de banda de acceso a la memoria, los rangos dinámicos de cada componente del vector de movimiento se pueden limitar de manera diferente. Por ejemplo, solo uno de un rango dinámico de un componente X y un rango dinámico de un componente Y se puede limitar o el rango dinámico del componente Y se puede limitar aún más en comparación con el rango dinámico del componente X.

25 El rango dinámico limitado del vector de movimiento se transmite a través de un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen (PPS), un encabezado de segmento o similar, y el decodificador realiza de manera similar la limitación de un rango dinámico de un vector de movimiento temporal en la secuencia, la imagen o el segmento. En este caso, también se puede transmitir una profundidad de bits, que es el tamaño de un espacio de memoria requerido para almacenar el vector de movimiento representado en el rango dinámico. Además, es posible almacenar de manera eficiente el vector de movimiento temporal para que coincida con las características de movimiento de la imagen utilizando el rango dinámico transmitido a través del conjunto de parámetros de secuencia, el conjunto de parámetros de imagen, el encabezado de segmento o similares, en lugar de almacenar el vector de movimiento usando una profundidad de bits que tiene un tamaño fijo.

30 Mientras tanto, el vector de movimiento se puede cuantificar y almacenar. En el caso de que el vector de movimiento se cuantifique y almacene, se reduce la precisión del vector de movimiento. Como método de cuantificación, hay una cuantificación uniforme en la que los tamaños de los pasos son uniformes, una cuantificación no uniforme en la que los tamaños de los pasos no son uniformes y similares. El tamaño de paso en la cuantificación se establece en un valor fijo predefinido entre el codificador y el decodificador o se transmite desde el codificador al decodificador a través del conjunto de parámetros de secuencia, el conjunto de parámetros de imagen, el encabezado de segmento o similares. El decodificador usa el vector de movimiento cuantificado tal cual o descuantifica y usa el vector de movimiento cuantificado. La FIG. 9 muestra un ejemplo de cuantificación de un vector de movimiento. Haciendo referencia a la FIG. 9, en el caso de que el vector de movimiento tenga un valor de componente de 32 a 48, el vector de movimiento se cuantifica a 40.

35 Además, el vector de movimiento se puede limitar en una resolución de representación y almacenar. La resolución de representación significa una unidad de píxel entero (unidad de 1 píxel), una unidad de fracción de píxel (una unidad de 1/2 píxel, una unidad de 1/4 de píxel o similar). Por ejemplo, una resolución del vector de movimiento procesado en una unidad de 1/4 de píxel se puede almacenar como un píxel entero. La resolución de representación del vector de movimiento se establece en un valor fijo predefinido entre el codificador y el decodificador o se transmite desde el codificador al decodificador a través del conjunto de parámetros de secuencia, el conjunto de parámetros de imagen, el encabezado de segmento o similares.

40 Además, solo con respecto a algunos vectores de movimiento entre los vectores de movimiento temporal almacenados en una memoria, se puede realizar un proceso de limitación de rango dinámico, un proceso de reducción de resolución de espacio y un proceso de cuantificación del vector de movimiento.

45 En el caso en el que el rango dinámico del vector de movimiento esté limitado y almacenado, la información sobre el rango dinámico del vector de movimiento se puede agregar y almacenar en la memoria. Por ejemplo, en el caso en el que el rango dinámico del vector de movimiento sea de -128 a +127, se puede almacenar adicionalmente una bandera de 1, y en el caso en el que el rango dinámico del vector de movimiento sea de -32 a +31, se puede almacenar adicionalmente una bandera de 0. En este caso, la información de la bandera se puede almacenar junto con el vector de movimiento o almacenar en una memoria diferente de la memoria en la que se almacena el vector de movimiento. En el caso en el que la información de la bandera y el vector de movimiento se almacenen en diferentes memorias, cuando se reconoce en qué rango dinámico se almacena un vector de movimiento específico, se puede permitir un acceso arbitrario a la información de la bandera. Además, la información sobre en qué rango dinámico se almacenan algunos vectores de movimiento se transmite a través del conjunto de parámetros de secuencia, el conjunto de parámetros de imagen, el encabezado de segmento o similares, lo que hace posible de este modo permitir que un decodificador realice una operación similar a la de un codificador.

En el caso en el que la resolución espacial del vector de movimiento se reduzca y se almacene, se puede agregar y almacenar en la memoria información sobre el tamaño de bloque del vector de movimiento. Por ejemplo, en el caso en el que el tamaño de bloque del vector de movimiento sea 4x4, se puede almacenar adicionalmente una bandera de 1, y en el caso en el que el tamaño de bloque del vector de movimiento sea 16x16, se puede almacenar adicionalmente una bandera de 0. En este caso, la información de la bandera se puede almacenar junto con el vector de movimiento o almacenar en una memoria diferente de la memoria en la que se almacena el vector de movimiento. En el caso en el que la información de la bandera y el vector de movimiento se almacenan en diferentes memorias, cuando se reconoce en qué tamaño de bloque se almacena un vector de movimiento específico, se puede permitir un acceso arbitrario a la información de la bandera. Además, la información sobre en qué tamaño de bloque se almacenan algunos vectores de movimiento se transmite a través del conjunto de parámetros de secuencia, el conjunto de parámetros de imagen, el encabezado de segmento o similares, lo que hace posible de este modo permitir que un decodificador realice una operación similar a la de un codificador.

En el caso en el que el vector de movimiento se cuantifique y almacene, la información sobre la precisión del vector de movimiento se puede agregar y almacenar en la memoria. Por ejemplo, en el caso en el que un tamaño de paso de la cuantificación sea 4, se puede almacenar adicionalmente una bandera de 1, y en el caso en el que el tamaño de paso de la cuantificación sea 1, se puede almacenar adicionalmente una bandera de 0. En este caso, la información de la bandera se puede almacenar junto con el vector de movimiento o almacenar en una memoria diferente de la memoria en la que se almacena el vector de movimiento. En el caso en el que la información de la bandera y el vector de movimiento se almacenen en diferentes memorias, cuando se reconoce en qué tamaño de paso se cuantifica y almacena un vector de movimiento específico, se puede permitir un acceso arbitrario a la información de la bandera. Además, la información sobre en qué tamaño de paso se cuantifican y almacenan algunos vectores de movimiento se transmite a través del conjunto de parámetros de secuencia, el conjunto de parámetros de imagen, el encabezado de segmento o similares, lo que hace posible permitir de este modo que un decodificador realice una operación similar a la de un codificador.

Además, en el caso en el que la información de movimiento se almacene en la memoria, la resolución espacial del vector de movimiento se puede reducir y almacenar. Aquí, la información de movimiento incluye al menos una de información del modo de interpredicción que indica un índice de imagen de referencia, un vector de movimiento, una dirección en un sentido, una dirección en dos sentidos o similar, requerida en el momento de la interpredicción, una lista de imágenes de referencia, e información del modo de predicción sobre si se realiza un modo de intrapredicción o se realiza un modo de interpredicción.

Por ejemplo, la información de movimiento de una unidad de predicción que tiene el tamaño de partición más grande entre una pluralidad de información de movimiento de una región específica se puede almacenar como información de movimiento representativa en la memoria. Aquí, la región específica puede incluir una región en el bloque de árbol de codificación y regiones de bloques vecinos del bloque de árbol de codificación. Además, la región específica puede ser una región que incluye un bloque en el que se almacena la información de movimiento en el caso en el que toda la imagen o el segmento se divida en un tamaño predeterminado.

Por ejemplo, después de que la información de movimiento, que se codifica en un método de combinación de información de movimiento, un método de omisión de información de codificación, o similar, se excluye de la pluralidad de información de movimiento incluida en la región específica, la información de movimiento representativa se puede almacenar en la memoria.

Por ejemplo, la información de movimiento generada con más frecuencia entre la pluralidad de información de movimiento incluida en la región específica se puede almacenar como información de movimiento representativa en la memoria. En este caso, se puede calcular el número de generación de la información de movimiento para cada tamaño del bloque o similar.

Por ejemplo, se puede almacenar información de movimiento en una posición específica entre la pluralidad de información de movimiento incluida en la región específica. Aquí, la posición específica, que es una posición incluida en la región específica, puede ser una posición fija de la región específica. Además, la posición específica se puede seleccionar como una de una pluralidad de posiciones. Cuando se usa la pluralidad de posiciones, se puede determinar una prioridad para cada posición, y la información de movimiento se puede almacenar en la memoria según la prioridad.

Por ejemplo, cuando la pluralidad de información de movimiento incluida en la región específica se almacena en la memoria, dado que la información de movimiento no existe fuera de un límite de un bloque codificado en un modo de intrapredicción, un bloque codificado en un modo de modulación codificada por pulsos (PCM), un segmento o una imagen, puede que la información de movimiento de la posición correspondiente no se almacene en la memoria.

En los ejemplos mencionados anteriormente, cuando se almacena la información de movimiento de la posición específica, en el caso en el que la información de movimiento de la posición correspondiente no exista, la información de movimiento de un bloque coubicado, la información de movimiento de un bloque ya codificado o la información de movimiento de un bloque vecino se puede utilizar como la información de movimiento de la posición correspondiente. Aquí, la posición específica puede ser una posición de muestra en un bloque vecino o una posición

5 del bloque. Por ejemplo, en el caso en el que la información de movimiento de la posición específica no exista, se puede almacenar en la memoria un valor medio o un valor promedio entre la información de movimiento de los bloques vecinos que están codificados en interpredicción. Por ejemplo, en el caso en el que no exista la información de movimiento de la posición específica, se puede almacenar en la memoria un valor promedio de la información de movimiento de los bloques vecinos. Cuando se calculan el valor medio y el valor promedio, en el caso de que la información de movimiento de los bloques vecinos sea diferente de al menos uno del índice de imágenes de referencia, la lista de imágenes de referencia y la información del modo de interpredicción, un tamaño del vector de movimiento se puede ajustar según el índice de imágenes de referencia, la lista de imágenes de referencia, la información del modo de interpredicción, un recuento del orden de las imágenes y similares.

10 III. Proceso de derivación de vector de movimiento

En el caso en el que la información de movimiento se almacene en la memoria usando los métodos de información de movimiento mencionados anteriormente y la información de movimiento de la imagen de referencia se use en el método de predicción de vector de movimiento, el método de predicción de vector de movimiento avanzado o el método de combinación de información de movimiento, se puede traer la información de movimiento almacenada.

15 20 Por ejemplo, se puede traer la información de movimiento de una posición correspondiente a la del bloque de árbol de codificación en la imagen de referencia. En este caso, la posición correspondiente a la del bloque de árbol de codificación en la imagen de referencia puede ser una posición fija en una región específica o una posición relativa desde la posición del bloque de árbol de codificación.

Las FIGS. 10 a 13 muestran ejemplos de obtención de información de movimiento a partir de una imagen de referencia.

25 En las FIGS. 10 a 13, un bloque X indica un bloque de árbol de codificación en una imagen de objeto de codificación/decodificación 1010, 1110, 1210 o 1310, y un bloque A, un bloque B, un bloque C, un bloque D y un bloque E indican bloques vecinos reconstruidos. Además, un bloque T en la imagen de referencia 1020, 1120, 1220 y 1320 indica un bloque coubicado correspondiente al bloque de árbol de codificación. Un bloque Y en la imagen de referencia 1320 de la FIG. 13 indica un bloque correspondiente a una posición distinta del bloque de objeto de codificación/decodificación.

Haciendo referencia a la FIG. 10, se puede traer la información de movimiento correspondiente a una posición correspondiente a una posición de píxel superior izquierda entre las posiciones de un bloque de árbol de codificación X en una imagen de referencia.

30 Haciendo referencia a la FIG. 11, se puede traer la información de movimiento correspondiente a una posición correspondiente a una posición de píxel central entre las posiciones de un bloque de árbol de codificación X en una imagen de referencia.

35 Haciendo referencia a la FIG. 12, se puede traer la información de movimiento correspondiente a una posición correspondiente a una posición de píxel inferior derecha entre las posiciones de un bloque de árbol de codificación X en una imagen de referencia.

Haciendo referencia a la FIG. 13, se puede traer la información de movimiento correspondiente a una posición correspondiente a una posición de píxel distinta de un bloque de árbol de codificación X en una imagen de referencia.

40 Se puede realizar un método de codificación/decodificación, como predicción de vector de movimiento, predicción de vector de movimiento avanzada, combinación de información de movimiento, salto de combinación de información de movimiento, o similares, utilizando la información de movimiento almacenada en la memoria, es decir, la información de movimiento de la imagen de referencia.

45 El vector de movimiento se puede almacenar en la memoria usando al menos uno de un método de limitación de un rango dinámico de un vector de movimiento, un método de reducción de una resolución espacial de un vector de movimiento, un método de cuantificación de un vector de movimiento y un método de reducción de una resolución de representación de un vector de movimiento, y el vector de movimiento almacenado se puede usar para predecir un vector de movimiento del bloque de árbol de codificación y combinar la información de movimiento del mismo.

50 Un proceso de traer el vector de movimiento de la imagen de referencia de la memoria se denomina proceso de derivación de un vector de movimiento temporal. En un proceso de derivación de un vector de movimiento temporal, TMVbitWidth indica un ancho de bits de un vector de movimiento temporal almacenado en la memoria.

Las entradas en el proceso de derivación de un vector de movimiento temporal son una ubicación (xP, yP) que especifica la muestra de luma superior izquierda de la unidad de predicción actual con relación a la muestra superior izquierda de la imagen actual, variables que especifican el ancho y la altura de la unidad de predicción para luma, nPSW y nPSH, el índice de referencia de la partición de la unidad de predicción actual refIdxLX (con X que es 0 o 1).

Las salidas en el proceso son la predicción del vector de movimiento mvLXCol y la bandera de disponibilidad availableFlagLXCol.

5 La función RefPicOrderCnt (pic, refidx, LX) se especifica mediante el valor de PicOrderCnt de la imagen que es la imagen de referencia RefPicListX [refidx] de la imagen, con X que es 0 o 1. PicOrderCnt de la imagen de referencia se mantendrá hasta que la imagen sea marcada como "no existente". Clip3(a, b, c) significa una función de recorte c para existir en un rango entre a y b.

10 Si slice_type es igual a B y collocated_from_l0_flag es igual a 0, la variable colPic especifica la imagen que contiene la partición coubicada según lo especificado por RefPicList1[0]. De lo contrario (slice_type es igual a B y collocated_from_l0_flag es igual a 1 o slice_type es igual a P), la variable colPic especifica la imagen que contiene la partición coubicada según lo especificado por RefPicList0[0].

15 La variable colPu y su posición (xPCol, yPCol) se derivan en los siguientes pasos ordenados:

1. La posición de luma inferior derecha (xPRb, yPRb) de la unidad de predicción actual se define como se representa en las Ecuaciones 8 y 9.

[Ecuación 8]

$$xPRb = xP + nPSW$$

[Ecuación 9]

$$yPRb = yP + nPSH$$

2. Si colPu está codificado en un modo de intrapredicción o colPu no está disponible,

20 (1) La posición de luma central de la unidad de predicción actual se define como se representa por las Ecuaciones 10 y 11.

[Ecuación 10]

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

[Ecuación 11]

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

25 (2) La variable colPu se establece como la unidad de predicción que cubre la posición modificada dada por ((xPCtr >> 4) << 4, (yPCtr >> 4) << 4) dentro del colPic.

3. (xPCol, yPCol) se establece igual a la muestra de luma superior izquierda de colPu con relación a la muestra de luma superior izquierda de colPic.

Las variables mvLXCol y availableFlagLXCol se derivan de la siguiente manera.

30 1. Si colPu está codificado en un modo de intrapredicción o colPu no está disponible, ambos componentes de mvLXCol se establecen en 0 y availableFlagLXCol se establece en 0.

2. De lo contrario (colPu no está codificado en un modo de intrapredicción y colPu está disponible), las variables mvCol y refIdxCol se derivan de la siguiente manera,

35 (1) Si PredFlagL0 [xPCol] [yPCol] es igual a 0, el vector de movimiento mvCol y el índice de referencia refIdxCol se establecen en MvL1 [xPCol] [yPCol] y

RefIdxL1 [xPCol] [yPCol], respectivamente.

(2) De lo contrario (PredFlagL0 [xPCol] [yPCol] es igual a 1), se aplica lo siguiente.

40 1) Si PredFlagL1 [xPCol] [yPCol] es igual a 0, el vector de movimiento mvCol y el índice de referencia refIdxCol se establecen en MvL0 [xPCol] [yPCol] y RefIdxL0 [xPCol] [yPCol], respectivamente.

2) De lo contrario (PredFlagL1 [xPCol] [yPCol] es igual a 1), se aplica lo siguiente.

- a. Las siguientes asignaciones se realizan con X que es 0 o 1.

- 5 i. $\text{RefIdxCollX} = \text{RefIdxLX} [\text{xPCol}] [\text{yPCol}]$

10 ii. Si $\text{PicOrderCnt} (\text{colPic})$ es menor que $\text{PicOrderCnt} (\text{currPic})$ y $\text{RefPicOrderCnt} (\text{colPic}, \text{RefIdxCollX}, \text{LX})$ es mayor que $\text{PicOrderCnt} (\text{currPic})$ o $\text{PicOrderCnt} (\text{colPic})$ es mayor que $\text{PicOrderCnt} (\text{currPic})$ y $\text{RefIdxColderCnt} (\text{colPic}, \text{RefIdxCollX}, \text{LX})$ es menor que $\text{PicOrderCnt} (\text{currPic})$, la variable MvXCross es igual a 1.

15 iii. De lo contrario ($\text{PicOrderCnt} (\text{colPic})$ es menor que $\text{PicOrderCnt} (\text{currPic})$ y $\text{RefPicOrderCnt} (\text{colPic}, \text{RefIdxCollX}, \text{LX})$ es menor o igual que $\text{PicOrderCnt} (\text{currPic})$ o $\text{PicOrderCnt} (\text{colPic})$ es mayor que $\text{PicOrderCnt} (\text{currPic})$ y $\text{RefPicOrderCnt} (\text{colPic}, \text{RefIdxCollX}, \text{LX})$ es mayor o igual que $\text{PicOrderCnt} (\text{currPic})$), la variable MvXCross es igual a 0.

20 b. Si es cierta una de las siguientes condiciones, el vector de movimiento mvCol , el índice de referencia refIdxCol y ListCol se establecen en $\text{MvL1} [\text{xPCol}] [\text{yPCol}]$, RefIdxColl1 y L1 , respectivamente.

25 i. Mv0Cross es igual a 0 y Mv1Cross es igual a 1.

ii. Mv0Cross es igual a Mv1Cross y la lista de índices de referencia es igual a L1

c. De lo contrario, el vector de movimiento mvCol , el índice de referencia refIdxCol y ListCol se establecen en $\text{MvL0} [\text{xPCol}] [\text{yPCol}]$, RefIdxColl0 y L0 , respectivamente.

30 3) la variable $\text{availableFlagLXCol}$ se establece igual a 1 y se aplican las operaciones de la Ecuación 12 o las Ecuaciones 13 a 18.

a. Si $\text{PicOrderCnt} (\text{colPic}) - \text{RefPicOrderCnt} (\text{colPic}, \text{refIdxCol}, \text{ListCol})$ es igual a $\text{PicOrderCnt} (\text{currPic}) - \text{RefPicOrderCnt} (\text{currPic}, \text{refIdxLX}, \text{LX})$,

[Ecuación 12]

$$\text{mvLXCol} = \text{Clip3}(-1 \ll (\text{TMVBitWidth} - 1), 1 \ll (\text{TMVBitWidth} - 1) - 1, \text{mvCol})$$

[Ecuación 13]

$$\text{tx} = (16384 + \text{Abs}(\text{td} / 2)) / \text{td}$$

[Ecuación 14]

$$\text{DistScaleFactor} = \text{Clip3}(-1024, 1023, (\text{tb} * \text{tx} + 32) \gg 6)$$

[Ecuación 15]

$$\text{mvLXCol} = \text{Clip3}(-1 \ll (\text{TMVBitWidth} - 1), 1 \ll (\text{TMVBitWidth} - 1) - 1, \text{mvCol})$$

[Ecuación 16]

$$\text{mvLXCol} = \text{ClipMv}(\text{DistScaleFactor} * \text{mvLXCol} + 128) \gg 8$$

35 donde td y tb se derivan de las Ecuaciones 17 y 18.

[Ecuación 17]

$$\text{td} = \text{Clip3}(-128, 127, \text{PicOrderCnt}(\text{colPic}) - \text{RefPicOrderCnt}(\text{colPic}, \text{refIdxCol}, \text{ListCol}))$$

[Ecuación 18]

$$\text{tb} = \text{Clip3}(-128, 127, \text{PicOrderCnt}(\text{currPic}) - \text{RefPicOrderCnt}(\text{currPic}, \text{refIdxLX}, \text{LX}))$$

40 Es decir, haciendo referencia a las Ecuaciones 13 a 16, mvLXCol se deriva como una versión escalada del vector de movimiento mvCol .

5 Mientras tanto, aunque el vector de movimiento se recorta en un rango dinámico, en el caso en el que se escala el vector de movimiento recortado, el vector de movimiento recortado puede estar de nuevo fuera del rango dinámico. Por lo tanto, después de que se deriva el vector de movimiento escalado, el rango dinámico del vector de movimiento puede estar limitado. En este caso, cada una de las Ecuaciones 15 y 16 se puede reemplazar con las Ecuaciones 19 y 20.

[Ecuación 19]

$$mvLXCol = ClipMv((DistScaleFactor * mvLXCol + 128) \gg 8)$$

[Ecuación 20]

$$mvLXCol = Clip3(-1 \ll (TMVBitWidth - 1), 1 \ll (TMVBitWidth - 1) - 1, mvCol)$$

10 IV. Método de transmisión de información para recortar el vector de movimiento temporal en el decodificador.

En lo sucesivo, se describirá un método de transmisión de información requerido para recortar un vector de movimiento temporal en un decodificador utilizando el mismo método que el de un codificador.

15 TMVBitWidth en el proceso de derivación de un vector de movimiento temporal se puede transmitir desde el codificador al decodificador a través de un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, un encabezado de segmento o similar.

[Tabla 5]

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

bit_width_temporal_motion_vector_minus8 de la Tabla 5 especifica el ancho de bits del componente del vector de movimiento temporal. Cuando no está presente bit_width_temporal_motion_vector_minus8, se deducirá que es igual a 0. El ancho de bits del componente del vector de movimiento temporal se especifica de la siguiente manera:

20 [Ecuación 21]

$$TMVBitWidth = bit_width_temporal_motion_vector_minus8 + 8$$

1. Método de transmisión de información 1 - en el caso en el que el vector de movimiento esté comprimido y la profundidad de bits del vector de movimiento esté limitada

[Tabla 6]

seq_parameter_set_rbsp () {	C	Descriptor
.....		
motion_vector_buffer_comp_flag	0	u(1)
si(motion_vector_buffer_comp_flag)		
motion_vector_buffer_comp_ratio_log2	0	u(8)
bit_depth_temporal_motion_vector_constraint_flag	0	u(1)

si(bit_depth_temporal_motion_vector_constraint_flag)		
bit_depth_temporal_motion_vector_minus8	0	se(v)
.....		
rbsp_trailing_bits ()	0	
}		

Haciendo referencia a la Tabla 6, en el caso en el que motion_vector_buffer_comp_flag sea igual a 1, especifica que se aplica el proceso de compresión de la memoria intermedia del vector de movimiento.

- 5 motion_vector_buffer_comp_ratio_log2 especifica la relación de compresión en el proceso de compresión de la memoria intermedia de vector de movimiento. Cuando no está presente motion_vector_buffer_comp_ratio_log2, se deducirá que es igual a 0. La relación de compresión de la memoria intermedia de vector de movimiento se especifica de la siguiente manera:

[Ecuación 22]

$$MVBufferCompRatio = 1 << motion_vector_buffer_comp_ratio_log2$$

- 10 De nuevo haciendo referencia a la Tabla 6, en el caso en el que bit_depth_temporal_motion_vector_constraint_flag sea igual a 1, especifica que se aplica el proceso de restricción de limitación de profundidad de bits del vector de movimiento temporal.

- 15 Bit_depth_temporal_motion_vector_minus8 especifica la profundidad de bits del vector de movimiento temporal. Cuando no está presente bit_depth_temporal_motion_vector_minus8, se deducirá que es igual a 0. La profundidad de bits del vector de movimiento temporal se especifica de la siguiente manera:

[Ecuación 23]

$$TMVBitDepth = bit_depth_temporal_motion_vector_minus8 + 8$$

2. Método de transmisión de información 2 - en el caso en el que la profundidad de bits del vector de movimiento esté limitada

20

[Tabla 7]

seq_parameter_set_rbsp () {	C	Descriptor
.....		
bit_depth_temporal_motion_vector_constraint_flag	0	u(1)
si (bit_depth_temporal_motion_vector_constraint_flag)		
bit_depth_temporal_motion_vector_minus8	0	se(v)
.....		
rbsp_trailing_bits ()	0	
}		

Haciendo referencia a la Tabla 7, en el caso en el que bit_depth_temporal_motion_vector_constraint_flag sea igual a 1, especifica que se aplica el proceso de restricción de profundidad de bits del vector de movimiento temporal.

bit_depth_temporal_motion_vector_minus8 especifica la profundidad de bits del vector de movimiento temporal. Cuando no está presente bit_depth_temporal_motion_vector_minus8, se deducirá que es igual a 0. La profundidad de bits del vector de movimiento temporal se especifica de la siguiente manera:

[Ecuación 24]

$$TMVBitDepth = \text{bit_depth_temporal_motion_vector_minus8} + 8$$

- 5 3. Método de transmisión de información 3 - en el caso en el que la profundidad de bits del vector de movimiento esté limitada

[Tabla 8]

seq_parameter_set_rbsp () {	C	Descriptor
.....		
bit_depth_temporal_motion_vector_minus8	0	se(v)
.....		
rbsp_trailing_bits ()	0	
}		

- 10 bit_depth_temporal_motion_vector_minus8 especifica la profundidad de bits del vector de movimiento temporal. Cuando no está presente bit_depth_temporal_motion_vector_minus8, se deducirá que es igual a 0. La profundidad de bits del vector de movimiento temporal se especifica de la siguiente manera:

[Ecuación 25]

$$TMVBitDepth = \text{bit_depth_temporal_motion_vector_minus8} + 8$$

- 15 4. Método de transmisión de información 4 - en el caso en el que la profundidad de bits esté limitada con respecto a cada uno de los componentes X e Y del vector de movimiento

[Tabla 9]

seq_parameter_set_rbsp () {	C	Descriptor
.....		
bit_depth_temporal_motion_vector_constraint_flag	0	u(1)
si (bit_depth_temporal_motion_vector_constraint_flag) {		
bit_depth_temporal_motion_vector_x_minus8	0	se(v)
bit_depth_temporal_motion_vector_y_minus8	0	se(v)
}		
.....		
rbsp_trailing_bits ()	0	
}		

Haciendo referencia a la Tabla 9, en el caso en el que `bit_depth_temporal_motion_vector_constraint_flag` sea igual a 1, especifica que se aplica el proceso de restricción de profundidad de bits del vector de movimiento temporal.

5 `bit_depth_temporal_motion_vector_x_minus8` especifica la profundidad de bits del componente del vector de movimiento temporal x. Cuando no está presente `bit_depth_temporal_motion_vector_x_minus8`, se deducirá que es igual a 0. La profundidad de bit del componente x del vector de movimiento temporal se especifica de la siguiente manera:

[Ecuación 26]

$$TMVXBitDepth = bit_depth_temporal_motion_vector_x_minus8 + 8$$

10 `bit_depth_temporal_motion_vector_y_minus8` especifica la profundidad de bits del componente y del vector de movimiento temporal. Cuando no está presente `bit_depth_temporal_motion_vector_y_minus8`, se deducirá que es igual a 0. La profundidad de bit del componente y del vector de movimiento temporal se especifica de la siguiente manera:

[Ecuación 27]

$$TMVXBitDepth = bit_depth_temporal_motion_vector_y_minus8 + 8$$

15 5. Método de transmisión de información 5 - en el caso en el que el vector de movimiento esté comprimido y la profundidad de bits del vector de movimiento esté limitada

[Tabla 10]

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

20 Haciendo referencia a la Tabla 10, en el caso en el que `motion_vector_buffer_comp_flag` sea igual a 1 especifica que se aplica el proceso de compresión de la memoria intermedia del vector de movimiento.

`motion_vector_buffer_comp_ratio_log2` especifica la relación de compresión en el proceso de compresión de la memoria intermedia del vector de movimiento. Cuando no está presente `motion_vector_buffer_comp_ratio_log2`, se deducirá que es igual a 0. La relación de compresión de la memoria intermedia del vector de movimiento se especifica de la siguiente manera:

25 [Ecuación 28]

$$MVRBufferCompRatio = 1 << motion_vector_buffer_comp_ratio_log2$$

V. Definición de rango dinámico a través de niveles de códec de video

El rango dinámico del vector de movimiento temporal se puede definir a través de un nivel de un códec de vídeo en lugar de ser transmitido a través del conjunto de parámetros de secuencia, el conjunto de parámetros de imagen o el

encabezado de segmento. El codificador y el decodificador pueden determinar un rango dinámico limitado del vector de movimiento usando información de nivel.

- Además, incluso en los niveles, los rangos dinámicos y/o las profundidades de bits de cada uno de los componentes X e Y del vector de movimiento se pueden definir de manera diferente, y se pueden definir valores mínimos y valores máximos de cada uno de los componentes.

5 Las Tablas 11 y 12 muestran un ejemplo de un caso en el que TMVBitWidth en el proceso de derivación de un vector de movimiento temporal descrito anteriormente se define en los niveles.

[Tabla 11]

Número de nivel	MaxTMVBitWidth (ancho máximo de bits del componente del MV temporal)
1	8
1b	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

- 10 Haciendo referencia a la Tabla 11, TMVBitWidth se establece como MaxTMVBitWidth definido en los niveles. Aquí, MaxTMVBitWidth indica un ancho de bits máximo de un vector de movimiento temporal cuando el vector de movimiento temporal se almacena en la memoria.

- 15 Mientras tanto, TMVBitWidth también se puede definir en los niveles, y se puede transmitir una diferencia del valor definido (un valor delta) a través del conjunto de parámetros de secuencia, el conjunto de parámetros de imagen o el encabezado de segmento. Es decir, TMVBitWidth se puede establecer como un valor obtenido sumando la diferencia transmitida a través del conjunto de parámetros de secuencia, el conjunto de parámetros de imagen o el encabezado de segmento a MaxTMVBitWidth definido en los niveles. Aquí, TMVBitWidth indica un ancho de bits del vector de movimiento temporal cuando el vector de movimiento temporal se almacena en la memoria.

[Tabla 12]

Número de nivel	MaxTMVBitWidth (profundidad máxima de bits del componente del MV temporal)
1	8
1b	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

[Tabla 13]

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

5 delta_bit_width_temporal_motion_vector_minus8 especifica el ancho de bits delta del componente del vector de movimiento temporal. Cuando no está presente delta_bit_width_temporal_motion_vector_minus8, se deducirá que es igual a 0. El ancho de bit del componente del vector de movimiento temporal se especifica de la siguiente manera:

[Ecuación 29]

$$TMVBitWidth = delta_bit_width_temporal_motion_vector_minus8 + MaxTMVBitWidth$$

Además, como se muestra en la Tabla 14, los rangos dinámicos de cada componente del vector de movimiento temporal también se pueden definir en niveles.

[Tabla 14]

Número de nivel	MaxMBPS (velocidad máxima de procesamiento de macrobloques) (MB/s)	MaxFS (tamaño máximo de image n) (MB)	MaxDpbMb s (tamaño máximo de memoria intermedia de imágenes decodificadas) (MB)	MaxBR (velocidad máxima de bits de video) (1000 bits/s, 1200 bits/s, cpbBrVclFactor bits/s o cpbBrNalFactor bits/s)	MaxCPB (tamaño máximo de CPB) (1000 bits, 1200 bits, pbBrVclFactor bits o cpbBrNalFactor bits)	MaxTmvR (rango máximo del componente del MV temporal) (muestras de imagen de luma)	MinCR (relación de compresión mínima)	MaxMvsPer2 Mb (número máximo de vectores de movimiento por dos MB consecutivos)
1	1 485	99	396	64	175	[-64, +63,75]	2	-
1b	1 485	99	396	128	350	[-64, +63,75]	2	-
1.1	3 000	396	900	192	500	[-128, +127,75]	2	-
1.2	6 000	396	2 376	384	1 000	[-128, +127,75]	2	
1.3	11 880	396	2 376	768	2 000	[-128, +127,75]	2	-
2	11 880	396	2 376	2 000	2 000	[-128, +127,75]	2	-
2.1	19 800	792	4 752	4 000	4 000	[-256, +255,75]	2	-
2.2	20 250	1 620	8 100	4 000	4 000	[-256, +255,75]	2	-
3	40 500	1 620	8 100	10 000	10 000	[-256, +255,75]	2	32
3.1	108 000	3 600	18 000	14 000	14 000	[-512, +511,75]	4	16
3.2	216 000	5 120	20 480	20 000	20 000	[-512, +511,75]	4	16
4	245 760	8 192	32 768	20 000	25 000	[-512, +511,75]	4	16
4.1	245 760	8 192	32 768	50 000	62 500	[-512, +511,75]	2	16
4.2	522 240	8 704	34 816	50 000	62 500	[-512, +511,75]	2	16

5	589 824	22 080	110 400	135 000	135 000	[-512, +511,75]	2	16
5.1	983 040	36 864	184 320	240 000	240 000	[-512, +511,75]	2	16

Además, como se muestra en las Tablas 15 a 17, los anchos de bits de cada componente del vector de movimiento temporal también se pueden definir en niveles.

[Tabla 15]

Número de nivel	MaxMBPS (velocidad máxima de procesamiento de macrobloques) (MB/s)	MaxFS (tamaño máximo de imagen) (MB)	MaxDpbMbs (tamaño máximo de memoria intermedia de imágenes decodificadas) (MB)	MaxBR (velocidad máxima de bits de video) (1000 bits/s, 1200 bits/s, cpbBrVclFactor bits/s o cpbBrNalFactor bits/s)	MaxCPB (tamaño máximo de CPB) (1000 bits, 1200 bits, cpbBrVclFactor bits o cpbBrNalFactor bits)	MaxTMVBitWidth (ancho máximo de bits del componente del temporal MV)	MinCR (relación de compresión mínima)	MaxMvsPer2Mb (número máximo de vectores de movimiento por dos MB consecutivos)
1	1 485	99	396	64	175	8	2	-
1b	1 485	99	396	128	350	8	2	-
1.1	3 000	396	900	192	500	8	2	-
1.2	6 000	395	2 376	384	1 000	8	2	-
1.3	11 880	396	2 376	768	2 000	8	2	-
2	11 880	396	2 376	2 000	2 000	8	2	-
2.1	19 800	792	4 752	4 000	4 000	8	2	-
2.2	20 250	1 620	8 100	4 000	4 000	8	2	-
3	40 500	1 620	8 110	10 000	10 000	8	2	32
3.1	108 000	3 600	18 000	14 000	14 000	10	4	16
3.2	216 000	5 120	20 480	20 000	20 000	10	4	16
4	245 760	8 192	32 768	20 000	25 000	10	4	16
4.1	245 760	8 192	32 768	50 000	62 500	10	2	16
4.2	522 240	8 704	34 816	50 000	62 500	10	2	16
5	589 824	22 080	110 400	135 000	135 000	10	2	16
5.1	983 040	36 864	184 320	240 000	240 000	10	2	16

[Tabla 16]

Número de nivel	MaxMBPS (velocidad máxima de procesamiento de macrobloques) (MB/s)	MaxFS (tamaño máximo de imagen) (MB)	MaxDpbMbs (tamaño máximo de memoria intermedia de imágenes decodificadas) (MB)	MaxBR (velocidad máxima de bits de video) (1000 bits/s, 1200 bits/s, cpbBrVclFactor bits/s o cpbBrNalFactor bits/s)	MaxCPB (tamaño máximo de CPB) (1000 bits, 1200 bits, cpbBrVclFactor bits o cpbBrNalFactor bits)	MaxTMVBitWidth (ancho máximo de bits del componente del MV temporal)	MinCR (relación de compresión mínima)	MaxMvsPer2Mb (número máximo de vectores de movimiento por dos MB consecutivos)
1	1 485	99	396	64	175	6	2	-
1b	1 485	99	396	128	350	6	2	-
1.1	3	396	900	192	500	7	2	-
1.2	6 000	396	2 376	384	1 000	7	2	-
1.3	11 880	396	2 376	768	2 000	7	2	-
2	11 880	396	2 376	2 000	2 000	7	2	-
2.1	19 800	792	4 752	4 000	4 000	8	2	-
2.2	20 250	1 620	8 100	4 000	4 000	8	2	-
3	40 500	1 620	8 100	10 000	10 000	8	2	32
3.1	108 000	3 600	18 000	14 000	14 000	10	4	16
3.2	216 000	5 120	20 480	20 000	20 000	10	4	16
4	245 760	8 192	32 768	20 000	25 000	10	4	16
4.1	245 760	8 192	32 768	50 000	62 500	10	2	16
4.2	522 240	8 704	34 816	50 000	62 500	10	2	16
5	589 824	22 080	110 400	135 000	135 000	10	2	16
5.1	983 040	36 864	184 320	240 000	240 000	10	2	16

[Tabla 17]

Número de nivel	MaxTMVBitWidth (ancho máximo de bits del componente del MV temporal)
1	8
1b	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

Además, como se muestra en la Tabla 18, un ancho de bits de un componente Y del vector de movimiento temporal también se puede definir en niveles.

[Tabla 18]

Número de nivel	MaxMBPS (velocidad máxima de procesamiento de macrobloques) (MB/s)	MaxFS (tamaño máximo de imagen) (MB)	MaxDpbMbs (tamaño máximo de memoria intermedia de imágenes decodificadas) (MB)	MaxBR (velocidad máxima de bits de video) (1000 bits/s, 1200 bits/s, cpbBrVclFactor bits/s o cpbBrNalFactor bits/s)	MaxCPB (tamaño máximo de CPB) (1000 bits, 1200 bits, cpbBrVclFactor bits o cpbBrNalFactor bits)	MaxTMVYBitWidth (profundidad máxima de bits del componente del MV temporal vertical)	MinCR (relación de compresión mínima)	MaxMvsPer2Mb (número máximo de vectores de movimiento por dos MB consecutivos)
1	1 485	99	396	64	175	8	2	
1b	1 485	99	396	128	350	8	2	-
1.1	3 000	396	900	192	500	8	2	
1.2	6 000	396	2 376	384	1 000	8	2	-
1.3	11 880	396	2 376	768	2 000	8	2	-
2	11 880	396	2 376	2 000	2 000	8	2	
2.1	19 800	792	4 752	4 000	4 000	8	2	-
2.2	20 250	1 620	8 100	4 000	4 000	8	2	-

Número de nivel	MaxMBPS (velocidad máxima de procesamiento de macrobloques (MB/s))	MaxFS (tamaño máximo de image (MB))	MaxDpbMbs (tamaño máximo de memoria intermedia de imágenes decodificadas (MB))	MaxBR (velocidad máxima de bits de video (1000 bits/s, 1200 bits/s, cpbBrVclFactor bits/s o cpbBrNalFactor bits/s))	MaxCPB (tamaño máximo de CPB) (1000 bits, 1200 bits, cpbBrVclFactor bits o cpbBrNalFactor bits)	MaxTMVYBitWidth (profundidad máxima de bits del componente del MV temporal vertical)	MinCR (relación de compresión mínima)	MaxMvsPer2Mb (número máximo de vectores de movimiento por dos MB consecutivos)
3	40 500	1 620	8 100	10 000	10 000	8	2	32
3.1	108 000	3 600	18 000	14 000	14 000	10	4	16
3.2	216 000	5 120	20 480	20 000	20 000	10	4	16
4	245 760	8 192	32 768	20 000	25 000	10	4	16
4.1	245 760	8 192	32 768	50 000	62 500	10	2	16
4.2	522 240	8 704	34 816	50 000	62 500	10	2	16
5	589 824	22 080	110 400	135 000	135 000	10	2	16
5.1	983 040	36 864	184 320	240 000	240 000	10	2	16

Además, el rango dinámico del vector de movimiento temporal se puede definir como un valor fijo predefinido entre el codificador y el decodificador sin transmisión de información sobre una limitación del vector de movimiento o almacenar en forma de una profundidad de bits fija.

En el caso en el que TMVBitWidth se fija al mismo valor y se usa en el codificador y el decodificador, TMVBitWidth puede ser un número entero positivo tal como 4, 6, 8, 10, 12, 14, 16 o similar. Aquí, TMVBitWidth indica el ancho de bits del vector de movimiento temporal cuando el vector de movimiento temporal se almacena en la memoria.

La FIG. 14 es un diagrama de flujo que muestra un método de codificación de una imagen según la presente invención. Haciendo referencia a la FIG. 14, el método de codificación de una imagen incluye un paso de recorte (S1410), un paso de almacenamiento (S1420) y un paso de codificación (S1430).

- 10 Un aparato de codificación de una imagen y/o un aparato de decodificación de una imagen recortan un vector de movimiento de una imagen de referencia en un rango dinámico predeterminado (S1410). Como se describe arriba hasta "I. Proceso de recorte del vector de movimiento", el vector de movimiento que está fuera del rango dinámico está representado por un valor mínimo o un valor máximo del rango dinámico correspondiente. Por lo tanto, como se describe anteriormente a través de "IV. Método de transmisión de información para recortar el vector de movimiento temporal en el decodificador" y "V. Definición de rango dinámico a través de niveles de códec de video, la profundidad de bits está limitada a través del nivel del códec de video, el conjunto de parámetros de secuencia y similares, o el rango dinámico está limitado a través del nivel del códec de video, lo que lo hace posible recortar el vector de movimiento de la imagen de referencia en el rango dinámico predeterminado.

- 20 El aparato de codificación de una imagen y/o el aparato de decodificación de una imagen almacenan el vector de movimiento recortado de la imagen de referencia en una memoria intermedia como se describe anteriormente a través de "II. Proceso de almacenamiento de vector de movimiento" (S1420). El vector de movimiento se puede almacenar en la memoria intermedia junto con la imagen reconstruida o por separado.

- 25 El aparato de codificación de una imagen codifica un vector de movimiento de un bloque de árbol de codificación usando el vector de movimiento almacenado de la imagen de referencia (S1430). Como se describe anteriormente a través de "III. Proceso de derivación del vector de movimiento", en el método avanzado de predicción del vector de movimiento utilizado en el HEVC, se puede usar un vector de movimiento de un bloque que existe en una posición que es la misma o corresponde a la del bloque de árbol de codificación en la imagen de referencia así como vectores de movimiento de bloques reconstruidos colocados alrededor del bloque de árbol de codificación. Por lo tanto, el vector de movimiento del bloque de árbol de codificación también puede ser un vector de movimiento de la imagen de referencia, es decir, un vector de movimiento temporal, así como vectores de movimiento de bloques vecinos adyacentes al bloque de árbol de codificación.

Mientras tanto, dado que el rango dinámico del componente X y el componente Y del vector de movimiento de la imagen de referencia se puede definir de manera diferente, cada componente del vector de movimiento de la imagen de referencia se puede recortar en cada rango dinámico.

- 5 Además, se puede utilizar un método de compresión de un vector de movimiento de una imagen de referencia así como un método de limitación de un rango dinámico de un vector de movimiento de una imagen de referencia. En el caso de limitación del rango dinámico del vector de movimiento de la imagen de referencia o compresión del vector de movimiento de la imagen de referencia, se puede definir una bandera que indica el rango dinámico y el vector de movimiento y un parámetro relacionado con el mismo en un nivel de un códec de video, un conjunto de parámetros de secuencia o similares.
- 10 Además, se puede realizar un método de codificación tal como predicción de vector de movimiento, predicción de vector de movimiento avanzada, combinación de información de movimiento, salto de combinación de información de movimiento o similar, utilizando la información de movimiento almacenada en la memoria, es decir, la información de movimiento de la imagen de referencia.
- 15 La FIG. 15 es un diagrama de flujo que muestra un método de decodificación de una imagen según la presente invención. Haciendo referencia a la FIG. 15, el método de decodificación de una imagen incluye un paso de recorte (S1510), un paso de almacenamiento (S1520), un paso de derivación (S1530) y un paso de decodificación (S1540).
- 20 El paso de recorte (S1510) y el paso de almacenamiento (S1520) de la FIG. 15 son similares al paso de recorte (S1410) y al paso de almacenamiento (S1420) de la FIG. 14 utilizando "I. Proceso de recorte de vector de movimiento" y "II. Proceso de almacenamiento de vector de movimiento" descritos anteriormente. Además, el paso de derivación (S1530) de la FIG. 15 usa "III. Proceso de derivación del vector de movimiento" descrito anteriormente y es simétrico al paso de codificación (S1430) de la FIG. 14. Por lo tanto, se omitirá una descripción detallada de los mismos.
- 25 Un aparato de decodificación de una imagen realiza la decodificación de interpredicción utilizando un vector de movimiento de un bloque de árbol de codificación (S1540). El aparato de decodificación de una imagen puede almacenar un vector de movimiento en una memoria usando al menos uno de un método de limitación de un rango dinámico de un vector de movimiento, un método de reducción de una resolución espacial de un vector de movimiento, un método de cuantificación de un vector de movimiento y un método de reducción de una resolución de representación de un vector de movimiento, y utilizar el vector de movimiento almacenado para predecir un vector de movimiento del bloque de árbol de codificación y combinar la información de movimiento del mismo.
- 30 Además, el aparato de decodificación de una imagen puede realizar un método de decodificación tal como predicción de vector de movimiento, predicción de vector de movimiento avanzada, combinación de información de movimiento, salto de combinación de información de movimiento o similar, utilizando la información de movimiento almacenada en la memoria, es decir, la información de movimiento de la imagen de referencia.

REIVINDICACIONES

1. Un aparato de decodificación de imágenes, que comprende:

una memoria intermedia de imágenes de referencia para almacenar una pluralidad de imágenes de referencia; y

una unidad de compensación de movimiento para generar un bloque de predicción para reconstruir un bloque a ser decodificado en una imagen actual

5 escalando un vector de movimiento de un bloque cúbicado en una imagen cúbicada, colPic, entre la pluralidad de imágenes de referencia, generándose el vector de movimiento escalado multiplicando un factor de escala y el vector de movimiento, y modificando el resultado de la multiplicación sumando primero un primer desplazamiento y luego aplicando una primera operación de desplazamiento aritmético,

10 recortando el vector de movimiento escalado en un primer rango predeterminado para generar un vector de movimiento recortado,

derivando un vector de movimiento del bloque a ser decodificado agregando el vector de movimiento recortado como un vector de movimiento predicho y una diferencia de vector de movimiento decodificada a partir de un flujo de bits, y

15 realizando interpredicción utilizando el vector de movimiento del bloque a ser decodificado, en caso de que una primera diferencia de recuento de orden de imágenes, POC, entre la imagen actual y una primera imagen de referencia a la que se hace referencia por el bloque a ser decodificado sea diferente de una segunda diferencia de POC entre la imagen cúbicada y una segunda imagen de referencia a la que se hace referencia por el bloque cúbicado,

20 en donde el factor de escala se calcula multiplicando la primera diferencia de POC recortada en un segundo rango predeterminado y un valor proporcional inverso de la segunda diferencia de POC recortada en el segundo rango predeterminado, modificando un resultado de la multiplicación agregando un segundo desplazamiento y luego aplicando una segunda operación de desplazamiento aritmético, y recortando el resultado modificado de la multiplicación en un tercer rango predeterminado, el valor proporcional inverso se calcula dividiendo un valor intermedio por la segunda diferencia de POC, y

25 el valor intermedio se calcula usando un valor absoluto de la segunda diferencia de POC dividido por 2, y se modifica en base a una tercera operación de desplazamiento,

en donde el bloque cúbicado se determina mediante la realización de

30 calcular una primera posición correspondiente a una posición inferior derecha del bloque a ser decodificado en la imagen actual tomando una posición superior izquierda del bloque a ser decodificado en la imagen actual y agregando un ancho y una altura del bloque a ser decodificado en la imagen actual a la coordenada x y la coordenada y de la posición superior izquierda del bloque a ser decodificado en la imagen actual, respectivamente;

35 determinar si un primer bloque que cubre la primera posición calculada en la imagen cúbicada está codificado en un modo de intrapredicción;

en caso de que el primer bloque no esté codificado en un modo de intrapredicción, determinar el primer bloque en la imagen cúbicada como el bloque cúbicado,

40 en caso de que el primer bloque esté codificado en un modo de intrapredicción, calcular una segunda posición correspondiente a una posición central del bloque a ser decodificado en la imagen actual tomando la posición superior izquierda del bloque a ser decodificado en la imagen actual y agregar la mitad del ancho y la mitad de la altura del bloque a ser decodificado en la imagen actual a la coordenada x y la coordenada y de la posición superior izquierda del bloque a ser decodificado en la imagen actual, respectivamente; y

determinar un segundo bloque que cubra la segunda posición calculada en la imagen cúbicada como el bloque cúbicado.

45 2. El aparato de decodificación de imágenes de la reivindicación 1, en donde cada uno del primer rango predeterminado al tercer rango predeterminado es un rango de valor fijo.

3. Un método de decodificación de imágenes, que comprende:

50 (a) escalar un vector de movimiento de un bloque cúbicado contenido en una imagen cúbicada, colPic, entre una pluralidad de imágenes de referencia, el vector de movimiento escalado que se genera multiplicando un factor de escala y el vector de movimiento, y modificar el resultado de la multiplicación agregando primero un primer desplazamiento y luego aplicando una primera operación de desplazamiento aritmético;

(b) recortar el vector de movimiento escalado en un primer rango predeterminado para generar un vector de movimiento recortado;

5 (c) derivar un vector de movimiento de un bloque a ser decodificado en una imagen actual agregando el vector de movimiento recortado como un vector de movimiento predicho y una diferencia de vector de movimiento decodificada a partir de un flujo de bits; y

(d) reconstruir el bloque a ser decodificado realizando la interprediccción utilizando el vector de movimiento del bloque a ser decodificado,

10 en donde los pasos (a) a (d) se realizan en caso de que una primera diferencia de recuento de orden de imagen, POC, entre la imagen actual y una primera imagen de referencia a la que se hace referencia por el bloque a ser decodificado sea diferente de una segunda diferencia de POC entre la imagen cúbicada y una segunda imagen de referencia a la que se hace referencia por el bloque cúbicado,

15 en donde el factor de escala se calcula multiplicando la primera diferencia de POC recortada en un segundo rango predeterminado y un valor proporcional inverso de la segunda diferencia de POC recortada en el segundo rango predeterminado, modificando el resultado de la multiplicación agregando primero un segundo desplazamiento y luego aplicando una segunda operación de desplazamiento aritmético y recortando el resultado modificado de la multiplicación en un tercer rango predeterminado, el valor proporcional inverso se calcula dividiendo un valor intermedio por la segunda diferencia de POC, y

20 el valor intermedio se calcula usando un valor absoluto de la segunda diferencia de POC dividido por 2, y se modifica en base a una tercera operación de desplazamiento,

25 en donde el bloque cúbicado se determina mediante la realización de

calcular una primera posición correspondiente a una posición inferior derecha del bloque a ser decodificado en la imagen actual tomando una posición superior izquierda del bloque a ser decodificado en la imagen actual y agregando un ancho y una altura del bloque a ser decodificado en la imagen actual a la coordenada x y la coordenada y de la posición superior izquierda del bloque a ser decodificado en la imagen actual, respectivamente;

25 determinar si un primer bloque que cubre la primera posición calculada en la imagen cúbicada está codificado en un modo de intrapredicción;

30 en caso de que el primer bloque no esté codificado en un modo de intrapredicción, determinar el primer bloque en la imagen cúbicada como el bloque cúbicado,

35 en caso de que el primer bloque esté codificado en un modo de intrapredicción, calcular una segunda posición correspondiente a una posición central del bloque a ser decodificado en la imagen actual tomando la posición superior izquierda del bloque a ser decodificado en la imagen actual y añadir la mitad del ancho y la mitad de la altura del bloque a ser decodificado en la imagen actual a la coordenada x y la coordenada y de la posición superior izquierda del bloque a ser decodificado en la imagen actual, respectivamente; y

35 determinar un segundo bloque que cubra la segunda posición calculada en la imagen cúbicada como el bloque cúbicado.

4. El método de decodificación de imágenes de la reivindicación 3, en donde cada uno del primer rango predeterminado al tercer rango predeterminado es un rango de valor fijo.

5. Un aparato de codificación de imágenes, que comprende:

40 una memoria intermedia de imágenes de referencia para almacenar una pluralidad de imágenes de referencia;

una unidad de estimación de movimiento para generar un vector de movimiento de un bloque a ser codificado en una imagen actual; y

una unidad de codificación para generar un flujo de bits

45 escalando un vector de movimiento de un bloque cúbicado en una imagen cúbicada, colPic, entre la pluralidad de imágenes de referencia, el vector de movimiento escalado que se genera multiplicando un factor de escala y el vector de movimiento, y modificando un resultado de la multiplicación agregando primero un primer desplazamiento y luego aplicando una primera operación de desplazamiento aritmético,

recortando el vector de movimiento escalado en un primer rango predeterminado para generar un vector de movimiento recortado, y

- codificando una diferencia de vector de movimiento obtenida restando el vector de movimiento recortado como un vector de movimiento predicho del vector de movimiento del bloque a ser codificado, en caso de que una primera diferencia de recuento de orden de imagen, POC, entre la imagen actual y una primera imagen de referencia a la que se hace referencia por el bloque a ser codificado es diferente de una segunda diferencia de POC entre la imagen cúbicada y una segunda imagen de referencia a la que se hace referencia por el bloque cúbicado,
- en donde el factor de escala se calcula multiplicando la primera diferencia de POC recortada en un segundo rango predeterminado y un valor proporcional inverso de la segunda diferencia de POC recortada en el segundo rango predeterminado, modificando el resultado de la multiplicación agregando primero un segundo desplazamiento y luego aplicando una segunda operación de desplazamiento aritmético,
- 10 y recortando el resultado modificado de la multiplicación en un tercer rango predeterminado, el valor proporcional inverso se calcula dividiendo un valor intermedio por la segunda diferencia de POC, y el valor intermedio se calcula usando un valor absoluto de la segunda diferencia de POC dividido por 2, y se modifica en base a una tercera operación de desplazamiento,
- 15 en donde el bloque cúbicado se determina mediante la realización de calcular una primera posición correspondiente a una posición inferior derecha del bloque a ser codificado en la imagen actual tomando una posición superior izquierda del bloque a ser codificado en la imagen actual y agregar un ancho y una altura del bloque a ser codificado en la imagen actual en la coordenada x y la coordenada y de la posición superior izquierda del bloque a ser codificado en la imagen actual, respectivamente;
- 20 determinar si un primer bloque que cubre la primera posición calculada en la imagen cúbicada está codificado en un modo de intrapredicción; en caso de que el primer bloque no esté codificado en un modo de intrapredicción, determinar el primer bloque en la imagen cúbicada como el bloque cúbicado,
- 25 en caso de que el primer bloque esté codificado en un modo de intrapredicción, calcular una segunda posición correspondiente a una posición central del bloque a ser codificado en la imagen actual tomando la posición superior izquierda del bloque a ser codificado en la imagen actual y añadir la mitad del ancho y la mitad de la altura del bloque a ser codificado en la imagen actual a la coordenada x y la coordenada y de la posición superior izquierda del bloque a ser codificado en la imagen actual, respectivamente; y
- 30 determinar un segundo bloque que cubra la segunda posición calculada en la imagen cúbicada como el bloque cúbicado.
6. El aparato de codificación de imágenes de la reivindicación 5, en donde cada uno del primer rango predeterminado al tercer rango predeterminado es un rango de valor fijo.
7. Un método de codificación de imágenes, que comprende:
- 35 generar un vector de movimiento de un bloque a ser codificado en una imagen actual;
- (a) escalar un vector de movimiento de un bloque cúbicado contenido en una imagen cúbicada, entre una pluralidad de imágenes de referencia, el vector de movimiento escalado que se genera multiplicando un factor de escala y el vector de movimiento, y modificando el resultado de la multiplicación agregando primero un primer desplazamiento y luego aplicando una primera operación de desplazamiento aritmético;
- 40 (b) recortar el vector de movimiento escalado en un primer rango predeterminado para generar un vector de movimiento recortado; y
- (c) codificar una diferencia de vector de movimiento obtenida restando el vector de movimiento recortado como un vector de movimiento predicho del vector de movimiento del bloque a ser codificado,
- 45 en donde los pasos (a) a (c) se realizan en caso de que una primera diferencia de recuento de orden de imagen, POC, entre la imagen actual y una primera imagen de referencia a la que se hace referencia por el bloque a ser codificado sea diferente de una segunda diferencia de POC entre la imagen cúbicada y una segunda imagen de referencia a la que se hace referencia por el bloque cúbicado, y en donde el factor de escala se calcula multiplicando la primera diferencia de POC recortada en un segundo rango predeterminado y un valor proporcional inverso de la segunda diferencia de POC recortada en el segundo rango predeterminado, modificando el resultado de la multiplicación agregando primero un segundo desplazamiento y luego aplicando una segunda operación de desplazamiento aritmético,

y recortando el resultado modificado de la multiplicación en un tercer rango predeterminado, el valor proporcional inverso se calcula dividiendo un valor intermedio por la segunda diferencia de POC, y

el valor intermedio se calcula usando un valor absoluto de la segunda diferencia de POC dividido por 2, y se modifica en base a una tercera operación de desplazamiento,

- 5 en donde el bloque cúbico se determina mediante la realización de
calcular una primera posición correspondiente a una posición inferior derecha del bloque a ser codificado en la imagen actual tomando una posición superior izquierda del bloque a ser codificado en la imagen actual y agregar un ancho y una altura del bloque que a ser codificado en la imagen actual en la coordenada x y la coordenada y de la posición superior izquierda del bloque a ser codificado en la imagen actual, respectivamente;
- 10 determinar si un primer bloque que cubre la primera posición calculada en la imagen cúbica está codificado en un modo de intrapredicción;
en caso de que el primer bloque no esté codificado en un modo de intrapredicción, determinar el primer bloque en la imagen cúbica como el bloque cúbico;
- 15 en caso de que el primer bloque esté codificado en un modo de intrapredicción, calcular una segunda posición correspondiente a una posición central del bloque a ser codificado en la imagen actual tomando la posición superior izquierda del bloque a ser codificado en la imagen actual y agregar la mitad del ancho y la mitad de la altura del bloque a ser codificado en la imagen actual a la coordenada x y la coordenada y de la posición superior izquierda del bloque a ser codificado en la imagen actual, respectivamente; y
determinar un segundo bloque que cubra la segunda posición calculada en la imagen cúbica como el bloque cúbico.
- 20

FIG. 1

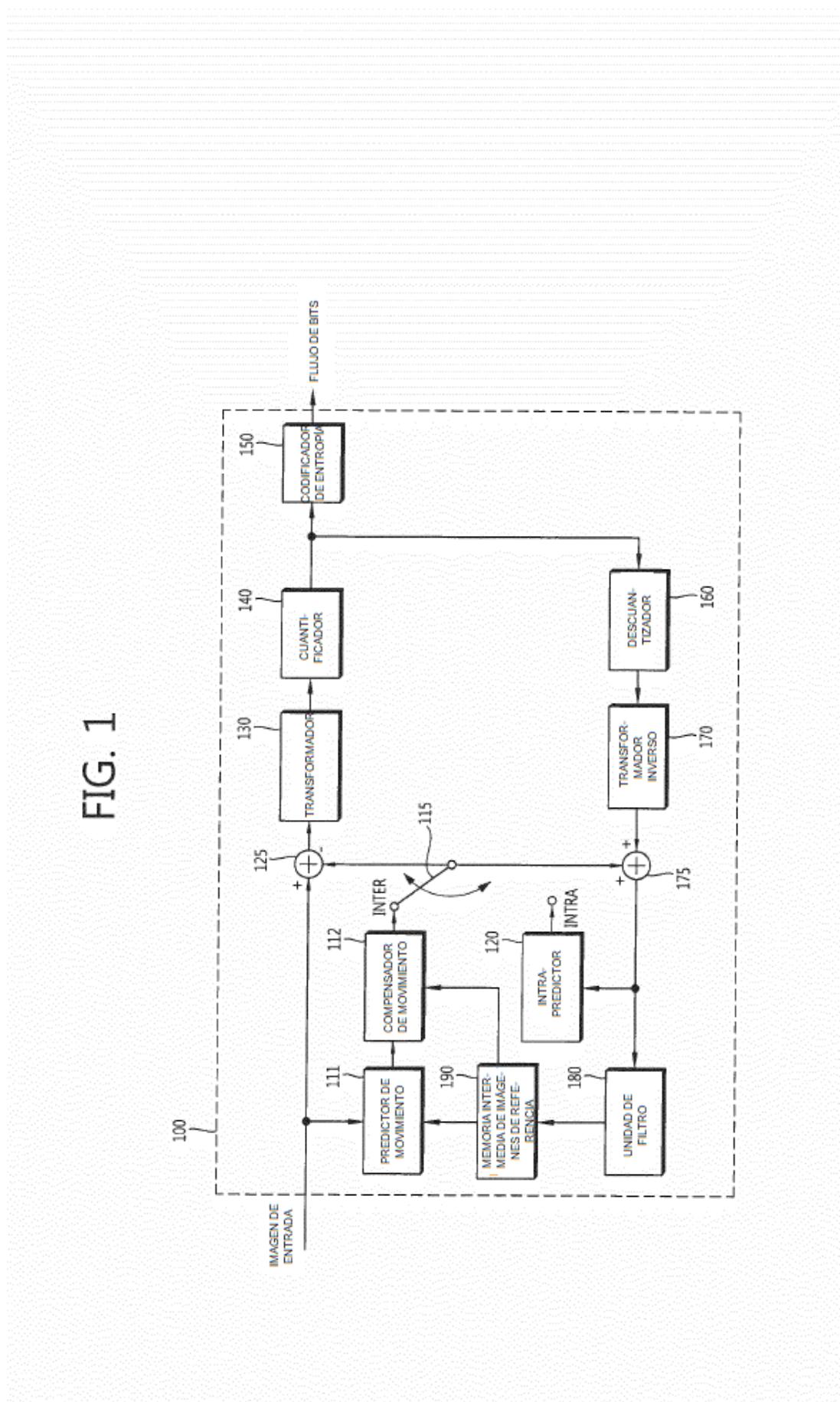


FIG. 2

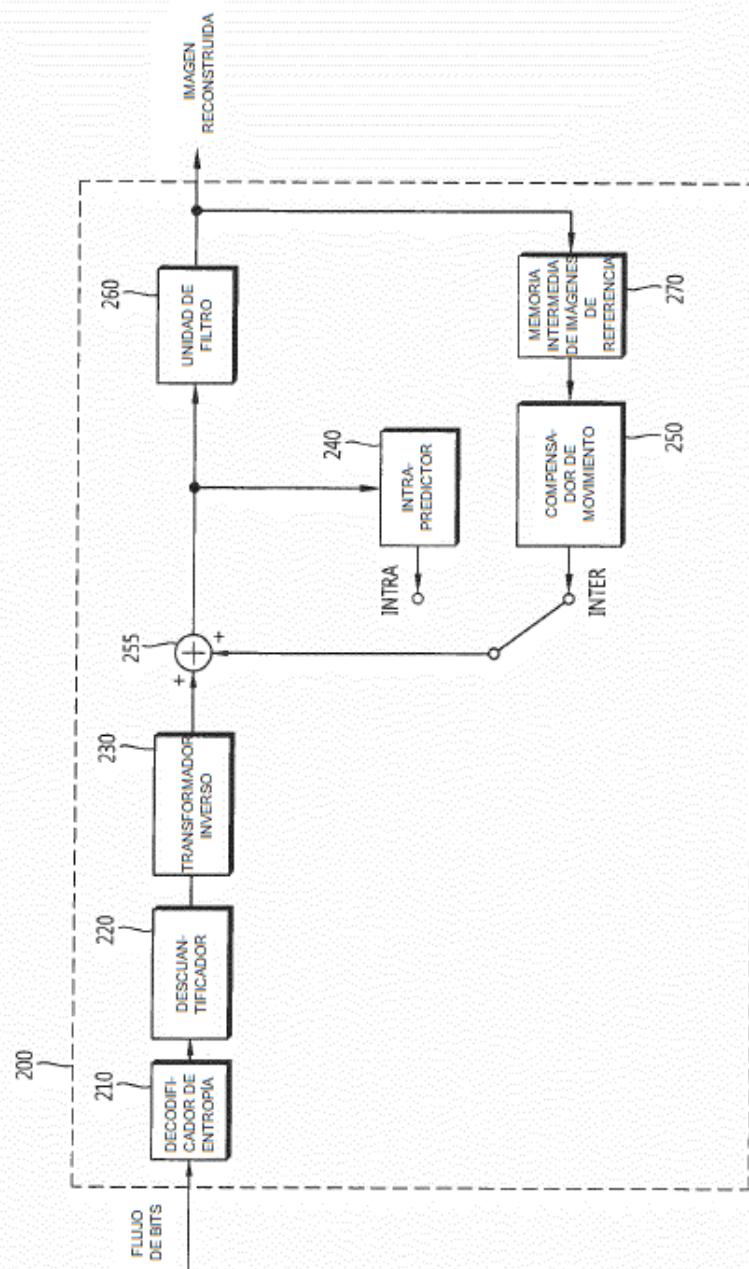


FIG. 3

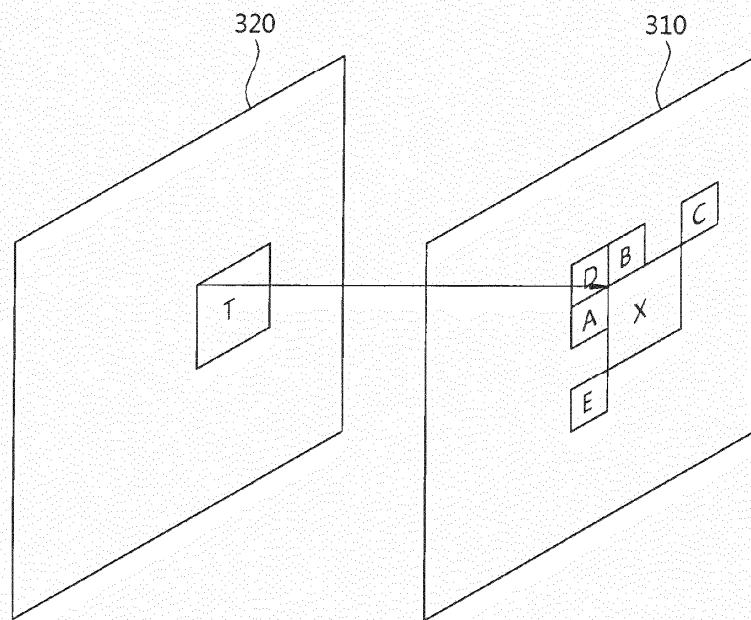


FIG. 4

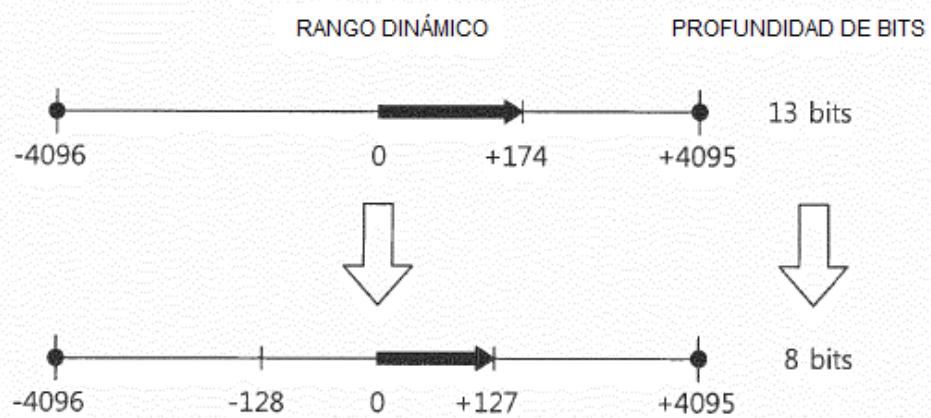


FIG. 5

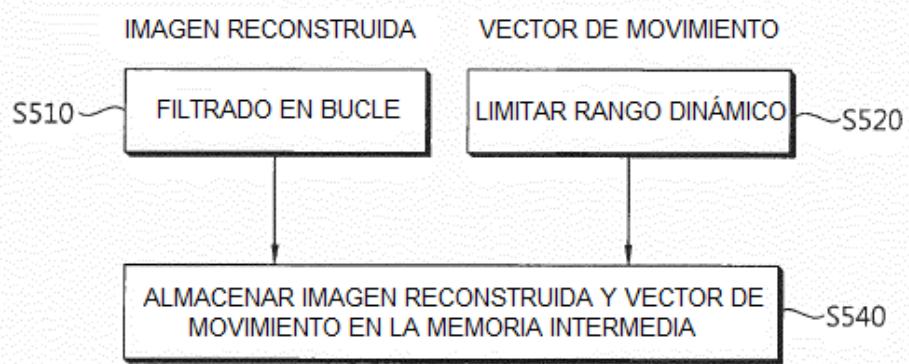


FIG. 6

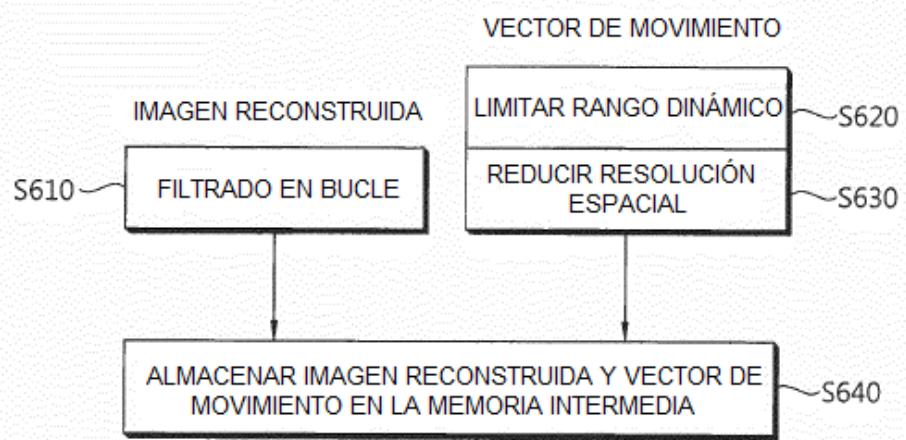


FIG. 7

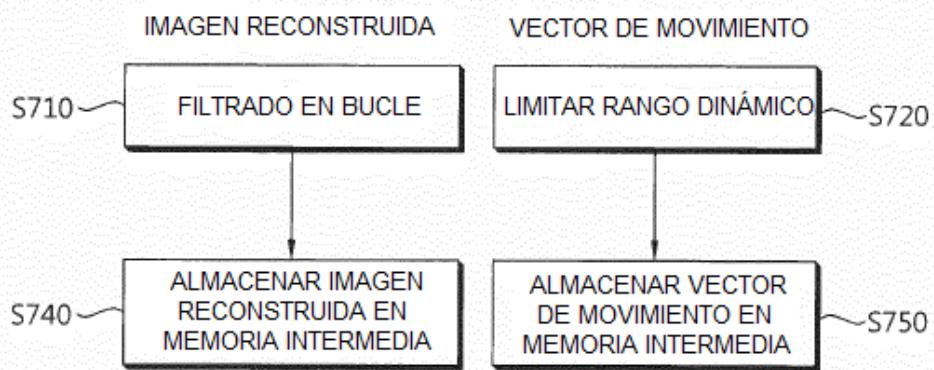


FIG. 8

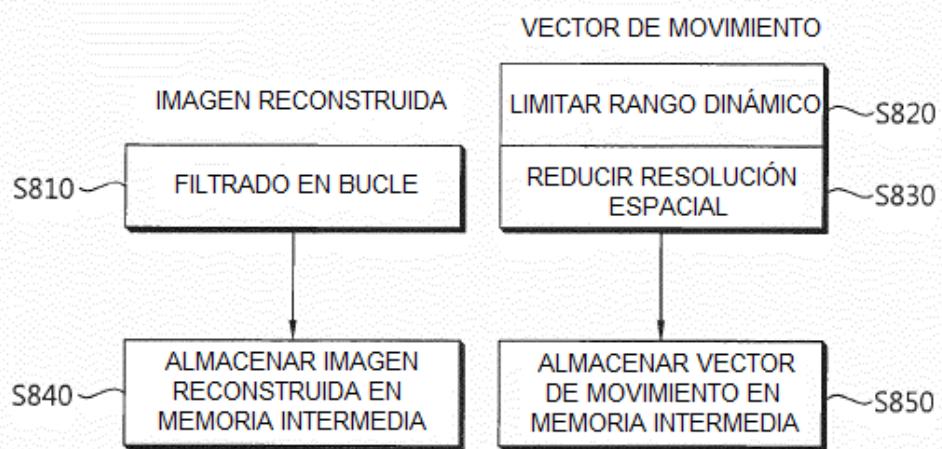


FIG. 9

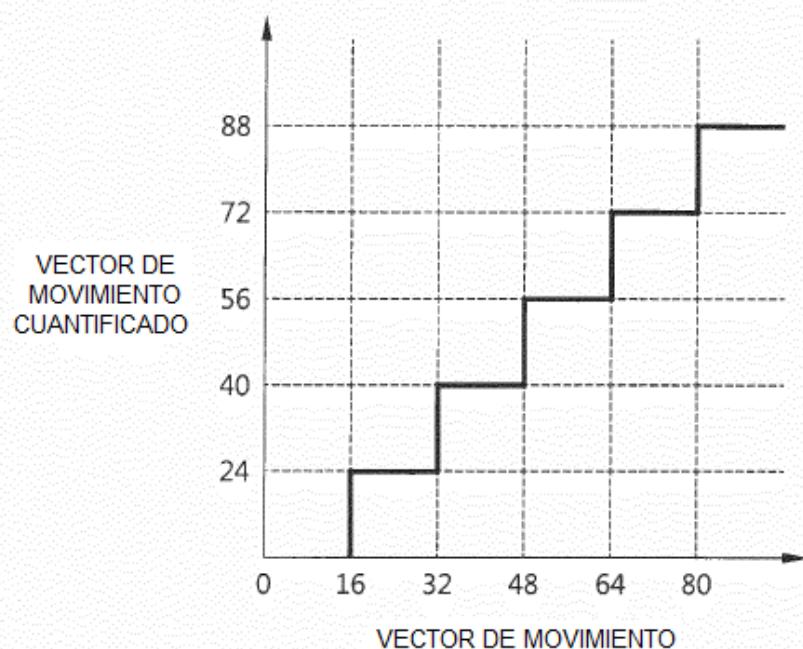


FIG. 10

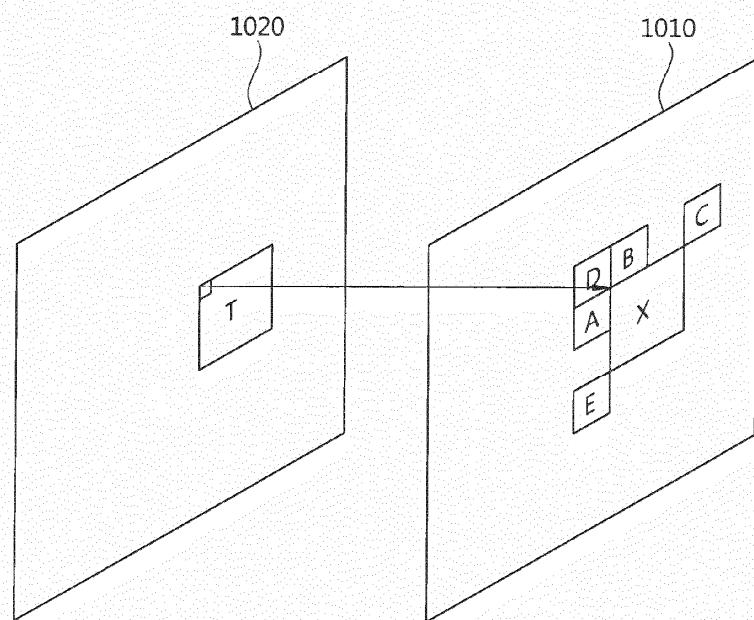


FIG. 11

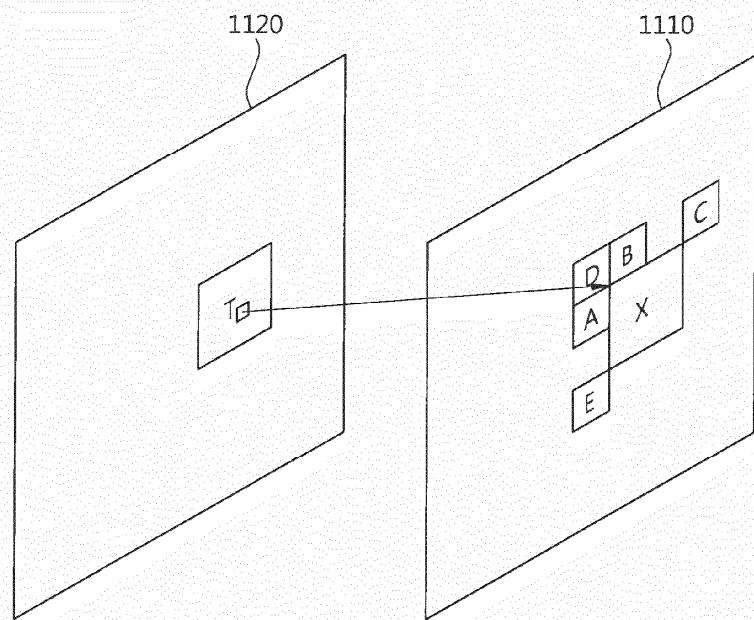


FIG. 12

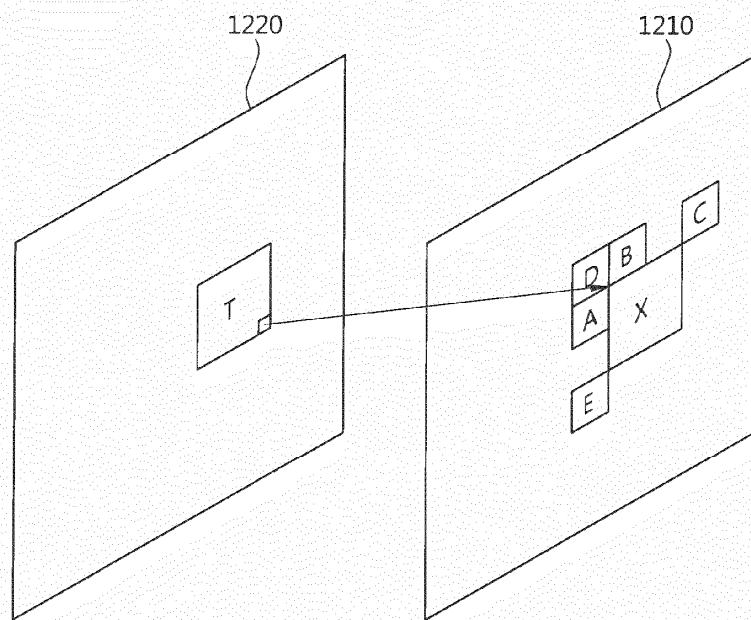


FIG. 13

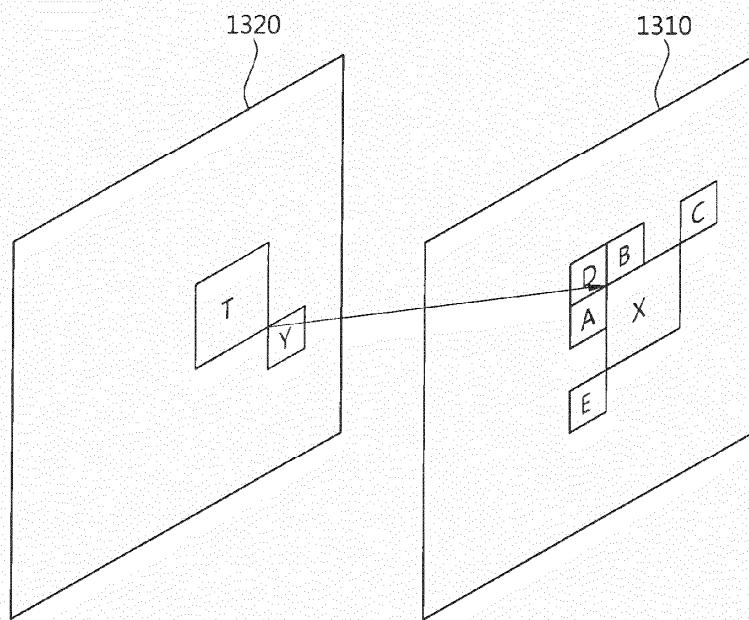


FIG. 14

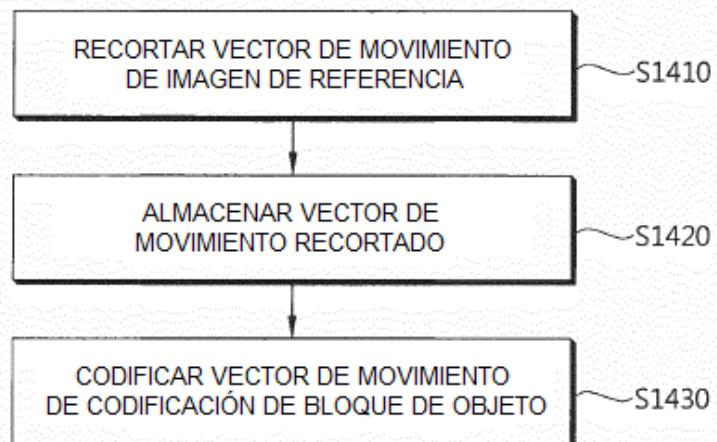


FIG. 15

