

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 003 036**

51 Int. Cl.:

H04N 19/52 (2014.01)
H04N 19/105 (2014.01)
H04N 19/146 (2014.01)
H04N 19/156 (2014.01)
H04N 19/167 (2014.01)
H04N 19/436 (2014.01)
H04N 19/53 (2014.01)
H04N 19/533 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **20.09.2018** **PCT/JP2018/034793**

87 Fecha y número de publicación internacional: **04.04.2019** **WO19065444**

96 Fecha de presentación y número de la solicitud europea: **20.09.2018** **E 18862113 (0)**

97 Fecha y número de publicación de la concesión europea: **27.11.2024** **EP 3691273**

54 Título: **Dispositivo de decodificación y procedimiento de decodificación**

30 Prioridad:

26.09.2017 US 201762563235 P

45 Fecha de publicación y mención en BOPI de la
traducción de la patente:
10.03.2025

73 Titular/es:

**PANASONIC INTELLECTUAL PROPERTY
CORPORATION OF AMERICA (100.00%)
20000 Mariner Avenue, Suite 200
Torrance, CA 90503, US**

72 Inventor/es:

**ABE, KIYOFUMI;
NISHI, TAKAHIRO;
TOMA, TADAMASA;
KANO, RYUICHI y
HASHIMOTO, TAKASHI**

74 Agente/Representante:

GONZÁLEZ PECES, Gustavo Adolfo

ES 3 003 036 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

Dispositivo de decodificación y procedimiento de decodificación

Campo técnico

5 La presente divulgación se refiere a un codificador, un decodificador, un procedimiento de codificación y un procedimiento de decodificación.

Técnica antecedente

Convencionalmente, H.265 se conoce como un estándar para codificar vídeos. H.265 también se conoce como codificación de vídeo de alta eficiencia (HEVC).

10 El documento WO 2017/157281 A1 se refiere a la derivación de vectores de movimiento para la codificación de vídeo. En particular, en los modos de fusión de coincidencia bilateral y temporal, los vectores de movimiento LIST_0 y LIST_1 en el candidato de fusión se utilizan como vectores de movimiento de inicio. Además, en los modos de fusión de coincidencia bilateral y temporal, se señala un índice de fusión, y la coincidencia bilateral o la coincidencia de plantilla realiza una búsqueda de refinamiento alrededor del candidato de fusión señalado.

15 Chen et al, "Output Algorithm of Joint Exploration Test Model 7 (JEM 7), JVET-G1001-v1 describen la derivación de un vector de movimiento con coincidencia de patrón. Con respecto al procesamiento de DMVR para derivar un vector de movimiento, para cada uno de los dos cuadros de referencia, el MV que produce el coste de plantilla mínimo se considera como el MV actualizado de esa lista para reemplazar el original.

20 Chen et al, "Further Improvement to HMkTA-1.0", VCEG-AZ07_v2) también describen la derivación de un vector de movimiento con coincidencia de patrón. En este, como punto de partida, se selecciona un candidato con el coste de coincidencia más pequeño de la lista de fusión de PU y se deriva un vector de movimiento a través de una búsqueda alrededor del punto de partida.

25 El documento US 2016/286230 A1 se refiere a la codificación de vídeo. En particular, un codificador de vídeo genera una lista de candidatos de vector de movimiento y utiliza uno o más de los candidatos de esta lista de candidatos como un vector de movimiento inicial. Los candidatos de vector de movimiento se asocian con bloques vecinos. El codificador refina aún más el candidato al realizar una búsqueda de movimiento adicional en un área indicada por el candidato seleccionado para determinar un vector de movimiento derivado.

Lista de citas**Literatura no de patente**

NPL 1: H.265 (ISO/IEC 23008-2 HEVC (High Efficiency Video Coding))

30 **Sumario de la invención****Problema técnico**

Se desea que dicho procedimiento de codificación y procedimiento de decodificación puedan reducir el retardo de procesamiento.

35 La presente divulgación tiene como objeto proporcionar un decodificador, un codificador, un procedimiento de decodificación y un procedimiento de codificación capaces de reducir el retardo de procesamiento. La invención se establece en el conjunto de reivindicaciones adjunto.

Soluciones al problema

40 Un codificador de acuerdo con un aspecto de la presente divulgación incluye circuitería y memoria. Utilizando la memoria, la circuitería: al codificar un bloque actual en un modo de predicción inter en el que un decodificador realiza una estimación de movimiento, deriva un primer vector de movimiento del bloque actual; almacena, en la memoria, el primer vector de movimiento derivado; deriva un segundo vector de movimiento del bloque actual; y genera una imagen de predicción del bloque actual al realizar una compensación de movimiento utilizando el segundo vector de movimiento. Al derivar el primer vector de movimiento, el primer vector de movimiento del bloque actual se deriva utilizando un primer vector de movimiento de un bloque procesado.

45 Un decodificador de acuerdo con un aspecto de la presente divulgación incluye circuitería y memoria. Utilizando la memoria, la circuitería: al decodificar un bloque actual en un modo de predicción inter en el que el decodificador realiza una estimación de movimiento, deriva un primer vector de movimiento del bloque actual; almacena, en la memoria, el primer vector de movimiento derivado; deriva un segundo vector de movimiento del bloque actual; y genera una imagen de predicción del bloque actual al realizar una compensación de movimiento utilizando el segundo vector de movimiento. Al derivar el primer vector de movimiento, el primer

50

vector de movimiento del bloque actual se deriva utilizando un primer vector de movimiento de un bloque procesado.

- 5 Tenga en cuenta que estos aspectos generales o específicos se pueden implementar mediante un sistema, un dispositivo, un procedimiento, un circuito integrado, un programa informático o un medio de grabación legible por ordenador no transitorio tal como una memoria de solo lectura de disco compacto (CD-ROM), o mediante cualquier combinación de sistemas, dispositivos, procedimientos, circuitos integrados, programas informáticos o medios de grabación.

Efecto ventajoso de la invención

- 10 La presente divulgación puede proporcionar un decodificador, un codificador, un procedimiento de decodificación y un procedimiento de codificación capaces de reducir el retardo de procesamiento.

Breve descripción de los dibujos

- La FIG. 1 es un diagrama de bloques que ilustra una configuración funcional de un codificador de acuerdo con la Realización 1.
- La FIG. 2 ilustra un ejemplo de división de bloques de acuerdo con la Realización 1.
- 15 La FIG. 3 es un gráfico que indica funciones de base de transformación para cada tipo de transformación.
- La FIG. 4A ilustra un ejemplo de una forma de filtro utilizada en ALF.
- La FIG. 4B ilustra otro ejemplo de una forma de filtro utilizada en ALF.
- La FIG. 4C ilustra otro ejemplo de una forma de filtro utilizada en ALF.
- La FIG. 5A ilustra 67 modos de predicción intra utilizados en la predicción intra.
- 20 La FIG. 5B es un diagrama de flujo para ilustrar un esquema de un procedimiento de corrección de imagen de predicción realizado a través del procesamiento de OBMC.
- La FIG. 5C es un diagrama conceptual para ilustrar un esquema de un procedimiento de corrección de imagen de predicción realizado a través del procesamiento de OBMC.
- La FIG. 5D ilustra un ejemplo de FRUC.
- 25 La FIG. 6 es para ilustrar la coincidencia de patrones (coincidencia bilateral) entre dos bloques a lo largo de una trayectoria de movimiento.
- La FIG. 7 es para ilustrar la coincidencia de patrones (coincidencia de plantillas) entre una plantilla en el cuadro actual y un bloque en un cuadro de referencia.
- La FIG. 8 es para ilustrar un modelo que supone un movimiento lineal uniforme.
- 30 La FIG. 9A es para ilustrar la derivación de un vector de movimiento de cada subbloque en base a vectores de movimiento de bloques vecinos.
- La FIG. 9B es para ilustrar un esquema de un procedimiento para derivar un vector de movimiento a través del modo de fusión.
- La FIG. 9C es un diagrama conceptual para ilustrar un esquema del procesamiento de DMVR.
- 35 La FIG. 9D es para ilustrar un esquema de un procedimiento de generación de imágenes de predicción que utiliza un procedimiento de corrección de luminancia realizado a través del procesamiento de LIC.
- La FIG. 10 es un diagrama de bloques que ilustra una configuración funcional de un decodificador de acuerdo con la Realización 1.
- 40 La FIG. 11 es un diagrama esquemático que ilustra un primer ejemplo de una configuración de canalización de acuerdo con la Realización 1.
- La FIG. 12 es un diagrama esquemático que ilustra un ejemplo de partición de bloques utilizada en la descripción del procesamiento de canalización de acuerdo con la Realización 1.
- La FIG. 13 es un diagrama de temporización que ilustra un ejemplo de temporización de procesamiento en el primer ejemplo de la configuración de canalización de acuerdo con la Realización 1.

La FIG. 14 es un diagrama de flujo del procesamiento de predicción inter en el primer ejemplo de la configuración de canalización de acuerdo con la Realización 1.

La FIG. 15 es un diagrama esquemático que ilustra un segundo ejemplo de la configuración de canalización de acuerdo con la Realización 1.

5 La FIG. 16 es un diagrama de temporización que ilustra un ejemplo de temporización de procesamiento en el segundo ejemplo de la configuración de canalización de acuerdo con la Realización 1.

La FIG. 17 es un diagrama de flujo del procesamiento de predicción inter en el segundo ejemplo de la configuración de canalización de acuerdo con la Realización 1.

10 La FIG. 18 es un diagrama esquemático que ilustra un tercer ejemplo de la configuración de canalización de acuerdo con la Realización 1.

La FIG. 19 es un diagrama de temporización que ilustra un ejemplo de temporización de procesamiento en el tercer ejemplo de la configuración de canalización de acuerdo con la Realización 1.

La FIG. 20 es un diagrama de flujo del procesamiento de predicción entre en el tercer ejemplo de la configuración de canalización de acuerdo con la Realización 1.

15 La FIG. 21 es un diagrama que ilustra un vector de movimiento al que se hace referencia de acuerdo con la Realización 1.

La FIG. 22 es un diagrama que ilustra un vector de movimiento al que se hace referencia de acuerdo con la Realización 1.

La FIG. 23 es un diagrama de bloques que ilustra un ejemplo de implementación de un codificador.

20 La FIG. 24 es un diagrama de bloques que ilustra un ejemplo de implementación de un decodificador.

La FIG. 25 ilustra una configuración general de un sistema de provisión de contenido para implementar un servicio de distribución de contenido.

La FIG. 26 ilustra un ejemplo de una estructura de codificación en codificación escalable.

La FIG. 27 ilustra un ejemplo de una estructura de codificación en codificación escalable.

25 La FIG. 28 ilustra un ejemplo de una pantalla de visualización de una página web.

La FIG. 29 ilustra un ejemplo de una pantalla de visualización de una página web.

La FIG. 30 ilustra un ejemplo de un teléfono inteligente.

La FIG. 31 es un diagrama de bloques que ilustra un ejemplo de configuración de un teléfono inteligente.

Descripción de realizaciones ejemplares

30 Un codificador de acuerdo con un aspecto de la presente divulgación incluye circuitería y memoria. Utilizando la memoria, la circuitería: cuando codifican un bloque actual en un modo de predicción inter en el que un decodificador realiza una estimación de movimiento, deriva un primer vector de movimiento del bloque actual; almacena, en la memoria, el primer vector de movimiento derivado; deriva un segundo vector de movimiento del bloque actual; y genera una imagen de predicción del bloque actual al realizar una compensación de movimiento utilizando el segundo vector de movimiento. Al derivar el primer vector de movimiento, el primer vector de movimiento del bloque actual se deriva utilizando un primer vector de movimiento de un bloque procesado.

40 De acuerdo con lo anterior, el decodificador puede comenzar a derivar el primer vector de movimiento del bloque actual, por ejemplo, después de que se derive el primer vector de movimiento de un bloque vecino en el control de canalización, sin esperar a que se complete la derivación del segundo vector de movimiento del bloque vecino. Por lo tanto, dado que el tiempo de espera que se produce en el control de canalización realizado por el decodificador se puede reducir en comparación con el caso de derivar el primer vector de movimiento utilizando el segundo vector de movimiento del bloque vecino, es posible reducir el retardo de procesamiento.

45 Por ejemplo, al derivar el primer vector de movimiento, (i) se puede generar una lista de predictores de vector de movimiento que indica una pluralidad de predictores de vector de movimiento utilizando el primer vector de movimiento del bloque procesado, y (ii) el primer vector de movimiento del bloque actual se puede determinar a partir de la pluralidad de predictores de vector de movimiento indicados en la lista de predictores de vector de movimiento.

Por ejemplo, el modo de predicción inter en el que el decodificador realiza la estimación de movimiento puede ser un modo de fusión, y al derivar el segundo vector de movimiento, el segundo vector de movimiento se puede derivar al realizar una estimación de movimiento en la vecindad del primer vector de movimiento.

5 Por ejemplo, el modo de predicción inter en el que el decodificador realiza la estimación de movimiento puede ser un modo de conversión ascendente de la velocidad del fotograma (FRUC), y al derivar el segundo vector de movimiento, el segundo vector de movimiento se puede derivar al realizar una estimación de movimiento en la vecindad del primer vector de movimiento.

10 Por ejemplo, el modo de predicción inter en el que el decodificador realiza la estimación de movimiento puede ser un modo de conversión ascendente de la velocidad del fotograma (FRUC), y al derivar el segundo vector de movimiento, (i) un tercer vector de movimiento se puede determinar a partir de la pluralidad de predictores de vector de movimiento indicados en la lista de predictores de vector de movimiento, y (ii) el segundo vector de movimiento se puede derivar al realizar una estimación de movimiento en la vecindad del tercer vector de movimiento.

15 Por ejemplo, al determinar el primer vector de movimiento, el primer vector de movimiento se puede derivar en base a, para cada una de las direcciones de predicción, un promedio o una mediana de la pluralidad de predictores de vector de movimiento indicados en la lista de predictores de vector de movimiento.

20 Por ejemplo, al determinar el primer vector de movimiento, entre la pluralidad de predictores de vector de movimiento indicados en la lista de predictores de vector de movimiento, un predictor de vector de movimiento indicado en la parte superior de la lista de predictores de vector de movimiento se puede determinar como el primer vector de movimiento.

25 Por ejemplo, al generar la lista de predictores de vector de movimiento, cada uno de la pluralidad de predictores de vector de movimiento se puede derivar utilizando el primer vector de movimiento del bloque procesado o un segundo vector de movimiento del bloque procesado, y al determinar el primer vector de movimiento, el primer vector de movimiento se puede determinar a partir de, entre la pluralidad de predictores de vector de movimiento indicados en la lista de predictores de vector de movimiento, uno o más candidatos de predictor de vector de movimiento derivados utilizando el segundo vector de movimiento del bloque procesado.

De acuerdo con lo anterior, puesto que el codificador puede determinar el primer vector de movimiento utilizando el segundo vector de movimiento altamente fiable, es posible suprimir una disminución en la fiabilidad del primer vector de movimiento.

30 Por ejemplo, al generar la lista de predictores de vector de movimiento, cada uno de la pluralidad de predictores de vector de movimiento se puede derivar utilizando el primer vector de movimiento del bloque procesado o un segundo vector de movimiento del bloque procesado, un predictor de vector de movimiento se puede derivar utilizando el primer vector de movimiento del bloque procesado cuando el bloque procesado se incluye en un cuadro que incluye el bloque actual, y un predictor de vector de movimiento se puede derivar utilizando el
35 segundo vector de movimiento del bloque procesado cuando el bloque procesado se incluye en un cuadro diferente del cuadro que incluye el bloque actual.

De acuerdo con lo anterior, cuando el bloque procesado se incluye en un cuadro diferente del cuadro que incluye el bloque actual, el codificador utiliza el segundo vector de movimiento y, como resultado, se puede mejorar la fiabilidad del predictor de vector de movimiento.

40 Por ejemplo, al generar la lista de predictores de vectores de movimiento, cada uno de la pluralidad de predictores de vectores de movimiento se puede derivar utilizando el primer vector de movimiento del bloque procesado o un segundo vector de movimiento del bloque procesado, y si se debe utilizar el primer vector de movimiento del bloque procesado o el segundo vector de movimiento del bloque procesado para realizar la derivación del predictor de vectores de movimiento se puede determinar de acuerdo con una posición del
45 bloque procesado relativa al bloque actual.

Por ejemplo, al generar la lista de predictores de vector de movimiento, el predictor de vector de movimiento se puede derivar de un primer bloque procesado entre una pluralidad de bloques procesados incluidos en un cuadro que incluye el bloque actual utilizando un primer vector de movimiento del primer bloque procesado, el primer bloque procesado es de N bloques antes del bloque actual en un orden de procesamiento, el predictor
50 de vector de movimiento se puede derivar de un segundo bloque procesado entre la pluralidad de bloques procesados utilizando un primer vector de movimiento del segundo bloque procesado, el segundo bloque procesado está después del primer bloque procesado en el orden de procesamiento, y el predictor de vector de movimiento se puede derivar de un tercer bloque procesado entre la pluralidad de bloques procesados utilizando un segundo vector de movimiento del tercer bloque procesado, el tercer bloque procesado está antes
55 del primer bloque procesado en el orden de procesamiento.

De acuerdo con lo anterior, el codificador utiliza el segundo vector de movimiento del tercer bloque procesado que está antes del primer bloque procesado en el orden de procesamiento, y como resultado, se puede mejorar la fiabilidad del predictor de vector de movimiento.

Por ejemplo, N puede ser 1.

- 5 Por ejemplo, el primer vector de movimiento se puede utilizar además en otro procesamiento diferente de la derivación del predictor del vector de movimiento.

Por ejemplo, el otro procesamiento puede ser el filtrado de bucle.

Por ejemplo, el segundo vector de movimiento se puede utilizar en el filtrado de bucle.

- 10 Por ejemplo, al codificar el bloque actual en un modo de retardo bajo, el primer vector de movimiento del bloque actual se puede derivar utilizando el primer vector de movimiento del bloque procesado.

De acuerdo con lo anterior, el codificador puede realizar un procesamiento apropiado de acuerdo con si se utiliza o no el modo de retardo bajo.

- 15 Por ejemplo, la información que indica si se debe codificar el bloque actual en el modo de retardo bajo se puede codificar en una región de encabezado de secuencia, una región de encabezado de cuadro, una región de encabezado de segmento o una región de información auxiliar.

Por ejemplo, la codificación del bloque actual en el modo de retardo bajo se puede intercambiar de acuerdo con el tamaño de un cuadro actual que incluye el bloque actual.

Por ejemplo, se puede intercambiar si se codifica el bloque actual en el modo de retardo bajo de acuerdo con la capacidad de procesamiento del decodificador.

- 20 Por ejemplo, se puede intercambiar si se codifica el bloque actual en el modo de bajo retardo de acuerdo con la información de perfil o nivel asignada a un flujo actual que se va a codificar.

Un decodificador de acuerdo con un aspecto de la presente divulgación incluye: circuitería y memoria. Utilizando la memoria, la circuitería: cuando decodifica un bloque actual en un modo de predicción inter en el que el decodificador realiza una estimación de movimiento, deriva un primer vector de movimiento del bloque actual; almacena, en la memoria, el primer vector de movimiento derivado; deriva un segundo vector de movimiento del bloque actual; y genera una imagen de predicción del bloque actual al realizar una compensación de movimiento utilizando el segundo vector de movimiento. Al derivar el primer vector de movimiento, el primer vector de movimiento del bloque actual se deriva utilizando un primer vector de movimiento de un bloque procesado.

- 30 De acuerdo con lo anterior, el decodificador puede comenzar a derivar el primer vector de movimiento del bloque actual, por ejemplo, después de que se derive el primer vector de movimiento de un bloque vecino en el control de canalización, sin esperar a que se complete la derivación del segundo vector de movimiento del bloque vecino. Por lo tanto, dado que el tiempo de espera que se produce en el control de canalización realizado por el decodificador se puede reducir en comparación con el caso de derivar el primer vector de movimiento utilizando el segundo vector de movimiento del bloque vecino, es posible reducir el retardo de procesamiento.

- 40 Por ejemplo, al derivar el primer vector de movimiento, (i) se puede generar una lista de predictores de vector de movimiento que indica una pluralidad de predictores de vector de movimiento utilizando el primer vector de movimiento del bloque procesado, y (ii) el primer vector de movimiento del bloque actual se puede determinar a partir de la pluralidad de predictores de vector de movimiento indicados en la lista de predictores de vector de movimiento.

Por ejemplo, el modo de predicción inter en el que el decodificador realiza la estimación de movimiento puede ser un modo de fusión, y al derivar el segundo vector de movimiento, el segundo vector de movimiento se puede derivar al realizar la estimación de movimiento en la vecindad del primer vector de movimiento.

- 45 Por ejemplo, el modo de predicción inter en el que el decodificador realiza la estimación de movimiento puede ser un modo de conversión ascendente de la velocidad del fotograma (FRUC), y al derivar el segundo vector de movimiento, el segundo vector de movimiento se puede derivar al realizar la estimación de movimiento en la vecindad del primer vector de movimiento.

- 50 Por ejemplo, el modo de predicción inter en el que el decodificador realiza la estimación de movimiento puede ser un modo de conversión ascendente de la velocidad del fotograma (FRUC), y al derivar el segundo vector de movimiento, (i) un tercer vector de movimiento se puede determinar a partir de la pluralidad de predictores de vector de movimiento indicados en la lista de predictores de vector de movimiento, y (ii) el segundo vector de movimiento se puede derivar al realizar una estimación de movimiento en la vecindad del tercer vector de movimiento.

Por ejemplo, al determinar el primer vector de movimiento, el primer vector de movimiento se puede derivar en base a, para cada una de las direcciones de predicción, de un promedio o una mediana de la pluralidad de predictores de vector de movimiento indicados en la lista de predictores de vector de movimiento.

- 5 Por ejemplo, al determinar el primer vector de movimiento, entre la pluralidad de predictores de vector de movimiento indicados en la lista de predictores de vector de movimiento, un predictor de vector de movimiento indicado en la parte superior de la lista de predictores de vector de movimiento se puede determinar como el primer vector de movimiento.

- 10 Por ejemplo, al generar la lista de predictores de vectores de movimiento, cada uno de la pluralidad de predictores de vectores de movimiento se puede derivar utilizando el primer vector de movimiento del bloque procesado o un segundo vector de movimiento del bloque procesado, y al determinar el primer vector de movimiento, el primer vector de movimiento se puede determinar a partir de, entre la pluralidad de predictores de vectores de movimiento indicados en la lista de predictores de vectores de movimiento, uno o más candidatos de predictores de vectores de movimiento derivados utilizando el segundo vector de movimiento del bloque procesado.

- 15 De acuerdo con lo anterior, dado que el decodificador puede determinar el primer vector de movimiento utilizando el segundo vector de movimiento altamente confiable, es posible suprimir una disminución en la confiabilidad del primer vector de movimiento.

- 20 Por ejemplo, al generar la lista de predictores de vector de movimiento, cada uno de la pluralidad de predictores de vector de movimiento se puede derivar utilizando el primer vector de movimiento del bloque procesado o un segundo vector de movimiento del bloque procesado, un predictor de vector de movimiento se puede derivar utilizando el primer vector de movimiento del bloque procesado cuando el bloque procesado se incluye en un cuadro que incluye el bloque actual, y un predictor de vector de movimiento se puede derivar utilizando el segundo vector de movimiento del bloque procesado cuando el bloque procesado se incluye en un cuadro diferente del cuadro que incluye el bloque actual.

- 25 De acuerdo con lo anterior, cuando el bloque procesado se incluye en un cuadro diferente del cuadro que incluye el bloque actual, el decodificador utiliza el segundo vector de movimiento y, como resultado, se puede mejorar la fiabilidad del predictor de vector de movimiento.

- 30 Por ejemplo, al generar la lista de predictores de vector de movimiento, cada uno de la pluralidad de predictores de vector de movimiento se puede derivar utilizando el primer vector de movimiento del bloque procesado o un segundo vector de movimiento del bloque procesado, y si se debe utilizar el primer vector de movimiento del bloque procesado o el segundo vector de movimiento del bloque procesado para realizar la derivación del predictor de vector de movimiento se puede determinar de acuerdo con una posición del bloque procesado relativo al bloque actual.

- 35 Por ejemplo, al generar la lista de predictores de vector de movimiento, el predictor de vector de movimiento se puede derivar de un primer bloque procesado entre una pluralidad de bloques procesados incluidos en un cuadro que incluye el bloque actual utilizando un primer vector de movimiento del primer bloque procesado, el primer bloque procesado es de N bloques antes del bloque actual en un orden de procesamiento, el predictor de vector de movimiento se puede derivar de un segundo bloque procesado entre la pluralidad de bloques procesados utilizando un primer vector de movimiento del segundo bloque procesado, el segundo bloque procesado está después del primer bloque procesado en el orden de procesamiento, y el predictor de vector de movimiento se puede derivar de un tercer bloque procesado entre la pluralidad de bloques procesados utilizando un segundo vector de movimiento del tercer bloque procesado, el tercer bloque procesado está antes del primer bloque procesado en el orden de procesamiento.

- 45 De acuerdo con lo anterior, el decodificador utiliza el segundo vector de movimiento del tercer bloque procesado que está antes del primer bloque procesado en el orden de procesamiento y, como resultado, se puede mejorar la fiabilidad del predictor del vector de movimiento.

Por ejemplo, N puede ser 1.

Por ejemplo, el primer vector de movimiento se puede utilizar además en otro procesamiento diferente de la derivación del predictor del vector de movimiento.

- 50 Por ejemplo, el otro procesamiento puede ser el filtrado de bucle.

Por ejemplo, el segundo vector de movimiento se puede utilizar en el filtrado de bucle.

Por ejemplo, al decodificar el bloque actual en un modo de bajo retardo, el primer vector de movimiento del bloque actual se puede derivar utilizando el primer vector de movimiento del bloque procesado.

De acuerdo con lo anterior, el decodificador puede realizar un procesamiento adecuado de acuerdo con si o no se utiliza el modo de bajo retardo.

Por ejemplo, la información que indica si se debe decodificar el bloque actual en el modo de bajo retardo se puede analizar a partir de una región de encabezado de secuencia, una región de encabezado de cuadro, una región de encabezado de segmento o una región de información auxiliar, y se puede determinar si se debe decodificar el bloque actual en el modo de bajo retardo en base a la información.

Por ejemplo, una configuración de canalización del decodificador puede incluir: una primera etapa en la que se deriva el primer vector de movimiento del bloque actual; y una segunda etapa que es diferente de la primera etapa y en la que se deriva el segundo vector de movimiento del bloque actual, y el procesamiento de la primera etapa para el bloque actual puede comenzar cuando se completa el procesamiento de la primera etapa para un bloque inmediatamente anterior al bloque actual en un orden de procesamiento, sin esperar a que se complete el procesamiento de la segunda etapa para un bloque que está M bloques antes del bloque actual en el orden de procesamiento.

Por ejemplo, una configuración de canalización del decodificador puede incluir: una primera etapa en la que se deriva el primer vector de movimiento del bloque actual; y una segunda etapa que es diferente de la primera etapa y en la que se deriva el segundo vector de movimiento del bloque actual, y el procesamiento de la primera etapa para el bloque actual puede comenzar cuando se deriva un primer vector de movimiento de un bloque que está M bloques antes del bloque actual en un orden de procesamiento, sin esperar a que se complete el procesamiento de la segunda etapa para el bloque que está M bloques antes del bloque actual en el orden de procesamiento.

Por ejemplo, M puede ser 1.

Un procedimiento de codificación de acuerdo con un aspecto de la presente divulgación incluye: cuando se codifica un bloque actual en un modo de predicción inter en el que un decodificador realiza una estimación de movimiento, derivar un primer vector de movimiento del bloque actual; almacenar, en la memoria, el primer vector de movimiento derivado; derivar un segundo vector de movimiento del bloque actual; y generar una imagen de predicción del bloque actual al realizar una compensación de movimiento utilizando el segundo vector de movimiento. Al derivar el primer vector de movimiento, el primer vector de movimiento del bloque actual se deriva utilizando un primer vector de movimiento de un bloque procesado.

De acuerdo con lo anterior, el decodificador puede comenzar a derivar el primer vector de movimiento del bloque actual, por ejemplo, después de que el primer vector de movimiento de un bloque vecino se deriva en el control de canalización, sin esperar a que se complete la derivación del segundo vector de movimiento del bloque vecino. Por lo tanto, dado que el tiempo de espera que se produce en el control de canalización realizado por el decodificador se puede reducir en comparación con el caso de derivar el primer vector de movimiento utilizando el segundo vector de movimiento del bloque vecino, es posible reducir el retardo de procesamiento.

Un procedimiento de decodificación de acuerdo con un aspecto de la presente divulgación incluye: cuando se decodifica un bloque actual en un modo de predicción inter en el que un decodificador realiza una estimación de movimiento, derivar un primer vector de movimiento del bloque actual; almacenar, en la memoria, el primer vector de movimiento derivado; derivar un segundo vector de movimiento del bloque actual; y generar una imagen de predicción del bloque actual al realizar una compensación de movimiento utilizando el segundo vector de movimiento. Al derivar el primer vector de movimiento, el primer vector de movimiento del bloque actual se deriva utilizando un primer vector de movimiento de un bloque procesado.

De acuerdo con lo anterior, el procedimiento de decodificación hace posible comenzar a derivar el primer vector de movimiento del bloque actual, por ejemplo después de que se derive el primer vector de movimiento de un bloque vecino en el control de canalización, sin esperar a que se complete la derivación del segundo vector de movimiento del bloque vecino. Por lo tanto, dado que el tiempo de espera que se produce en el control de canalización realizado por el decodificador se puede reducir en comparación con el caso de derivar el primer vector de movimiento utilizando el segundo vector de movimiento del bloque vecino, es posible reducir el retardo de procesamiento.

Tenga en cuenta que estos aspectos generales o específicos se pueden implementar mediante un sistema, un dispositivo, un procedimiento, un circuito integrado, un programa informático o un medio de grabación legible por ordenador no transitorio, tal como un CD-ROM, o mediante cualquier combinación de sistemas, dispositivos, procedimientos, circuitos integrados, programas informáticos o medios de grabación.

A continuación, se describirán realizaciones con referencia a los dibujos.

Tenga en cuenta que las realizaciones descritas a continuación muestran cada una un ejemplo general o específico. Los valores numéricos, formas, materiales, componentes, la disposición y conexión de los componentes, fases, orden de las fases, etc., indicados en las siguientes realizaciones son solo ejemplos, y por lo tanto no pretenden limitar el alcance de las reivindicaciones. Por lo tanto, entre los componentes de las

siguientes realizaciones, aquellos no enumerados en ninguna de las reivindicaciones independientes que definen los conceptos inventivos más amplios se describen como componentes opcionales.

Realización 1

En primer lugar, se presentará un esquema de la Realización 1. La Realización 1 es un ejemplo de un codificador y un decodificador al que son aplicables los procedimientos y/o configuraciones presentados en la descripción posterior de aspectos de la presente divulgación. Tenga en cuenta que la Realización 1 es simplemente un ejemplo de un codificador y un decodificador al que son aplicables los procedimientos y/o configuraciones presentados en la descripción de aspectos de la presente divulgación. Los procedimientos y/o configuraciones presentados en la descripción de aspectos de la presente divulgación también se pueden implementar en un codificador y un decodificador diferentes a los de la Realización 1.

Cuando los procedimientos y/o configuraciones presentados en la descripción de aspectos de la presente divulgación se aplican a la Realización 1, por ejemplo, se puede realizar cualquiera de los siguientes.

(1) en relación con el codificador o el decodificador de acuerdo con la Realización 1, entre los componentes incluidos en el codificador o el decodificador de acuerdo con la Realización 1, sustituir un componente correspondiente a un componente presentado en la descripción de aspectos de la presente divulgación por un componente presentado en la descripción de aspectos de la presente divulgación;

(2) en relación con el codificador o el decodificador de acuerdo con la Realización 1, implementar cambios discrecionales en funciones o procedimientos implementados realizados por uno o más componentes incluidos en el codificador o el decodificador de acuerdo con la Realización 1, tal como la adición, sustitución o eliminación, etc., de dichas funciones o procedimientos implementados, y luego sustituir un componente correspondiente a un componente presentado en la descripción de aspectos de la presente divulgación con un componente presentado en la descripción de aspectos de la presente divulgación;

(3) en relación con el procedimiento implementado por el codificador o el decodificador de acuerdo con la Realización 1, implementar cambios discrecionales tales como la adición de procedimientos y/o la sustitución, la eliminación de uno o más de los procedimientos incluidos en el procedimiento y luego sustituir un procedimiento correspondiente a un procedimiento presentado en la descripción de aspectos de la presente divulgación con un procedimiento presentado en la descripción de aspectos de la presente divulgación;

(4) combinar uno o más componentes incluidos en el codificador o el decodificador de acuerdo con la Realización 1 con un componente presentado en la descripción de aspectos de la presente divulgación, un componente que incluye una o más funciones incluidas en un componente presentado en la descripción de aspectos de la presente divulgación, o un componente que implementa uno o más procedimientos implementados por un componente presentado en la descripción de aspectos de la presente divulgación;

(5) combinar un componente que incluye una o más funciones incluidas en uno o más componentes incluidos en el codificador o el decodificador de acuerdo con la Realización 1, o un componente que implementa uno o más procedimientos implementados por uno o más componentes incluidos en el codificador o el decodificador de acuerdo con la Realización 1 con un componente presentado en la descripción de aspectos de la presente divulgación, un componente que incluye una o más funciones incluidas en un componente presentado en la descripción de aspectos de la presente divulgación, o un componente que implementa uno o más procedimientos implementados por un componente presentado en la descripción de aspectos de la presente divulgación;

(6) con respecto al procedimiento implementado por el codificador o el decodificador de acuerdo con la Realización 1, entre los procedimientos incluidos en el procedimiento, sustituir un procedimiento correspondiente a un procedimiento presentado en la descripción de aspectos de la presente divulgación con un procedimiento presentado en la descripción de aspectos de la presente divulgación; y

(7) combinar uno o más procedimientos incluidos en el procedimiento implementado por el codificador o el decodificador de acuerdo con la Realización 1 con un procedimiento presentado en la descripción de aspectos de la presente divulgación.

Tenga en cuenta que la implementación de los procedimientos y/o configuraciones presentados en la descripción de aspectos de la presente divulgación no se limita a los ejemplos anteriores. Por ejemplo, los procedimientos y/o configuraciones presentados en la descripción de aspectos de la presente divulgación se pueden implementar en un dispositivo utilizado para un propósito diferente del codificador de cuadro en movimiento/cuadro o el decodificador de cuadro en movimiento/cuadro divulgado en la Realización 1. Más aún, los procedimientos y/o configuraciones presentados en la descripción de aspectos de la presente divulgación se pueden implementar de forma independiente. Más aun, se pueden combinar los procedimientos y/o configuraciones descritos en diferentes aspectos.

[Esquema del codificador]

En primer lugar, se describirá el codificador de acuerdo con la Realización 1. La FIG. 1 es un diagrama de bloques que ilustra una configuración funcional del codificador 100 de acuerdo con la Realización 1. El codificador 100 es un codificador de cuadro en movimiento/cuadro que codifica un cuadro en movimiento/cuadro bloque por bloque.

Como se ilustra en la FIG. 1, el codificador 100 es un dispositivo que codifica un cuadro bloque por bloque, e incluye un divisor 102, sustractor 104, transformador 106, cuantificador 108, codificador de entropía 110, cuantificador inverso 112, transformador inverso 114, sumador 116, memoria de bloque 118, filtro de bucle 120, memoria de fotograma 122, predictor intra 124, predictor inter 126 y controlador de predicción 128.

El codificador 100 se realiza, por ejemplo, como un procesador y una memoria genéricos. En este caso, cuando un programa de software almacenado en la memoria es ejecutado por el procesador, el procesador funciona como divisor 102, sustractor 104, transformador 106, cuantificador 108, codificador de entropía 110, cuantificador inverso 112, transformador inverso 114, sumador 116, filtro de bucle 120, predictor intra 124, predictor inter 126 y controlador de predicción 128. Alternativamente, el codificador 100 se puede realizar como uno o más circuitos electrónicos dedicados correspondientes al divisor 102, sustractor 104, transformador 106, cuantificador 108, codificador de entropía 110, cuantificador inverso 112, transformador inverso 114, sumador 116, filtro de bucle 120, predictor intra 124, predictor inter 126 y controlador de predicción 128.

A continuación, se describirá cada componente incluido en el codificador 100.

[Divisor]

El divisor 102 divide cada cuadro incluido en un cuadro en movimiento de entrada en bloques, y envía cada bloque al sustractor 104. Por ejemplo, el divisor 102 divide primero un cuadro en bloques de un tamaño fijo (por ejemplo, 128×128). El bloque de tamaño fijo también se denomina unidad de árbol de codificación (CTU). El divisor 102 divide luego cada bloque de tamaño fijo en bloques de tamaños variables (por ejemplo, 64×64 o más pequeños), en base a la división recursiva de bloques de árbol binario y/o cuaternario. El bloque de tamaño variable también se denomina unidad de codificación (CU), unidad de predicción (PU) o unidad de transformación (TU). Tenga en cuenta que en esta realización, no hay necesidad de diferenciar entre CU, PU y TU; todos o algunos de los bloques en un cuadro se pueden procesar por CU, PU o TU.

La FIG. 2 ilustra un ejemplo de división de bloques de acuerdo con la Realización 1. En la FIG. 2, las líneas continuas representan límites de bloques de bloques divididos por división de bloques de árbol cuaternario, y las líneas discontinuas representan límites de bloques de bloques divididos por división de bloques de árbol binario.

En la presente memoria, el bloque 10 es un bloque cuadrado de 128×128 píxeles (bloque de 128×128). Este bloque de 128×128 primero se divide en cuatro bloques cuadrados de 64×64 (división de bloques de árbol cuaternario).

El bloque superior izquierdo de 64×64 se divide verticalmente en dos bloques rectangulares de 32×64 , y el bloque izquierdo de 32×64 se divide verticalmente en dos bloques rectangulares de 16×64 (división de bloques de árbol binario). Como resultado, el bloque superior izquierdo de 64×64 se divide en dos bloques de 16×64 11 y 12 y un bloque de 32×64 13.

El bloque superior derecho de 64×64 se divide horizontalmente en dos bloques rectangulares de 64×32 14 y 15 (división de bloques de árbol binario).

El bloque inferior izquierdo de 64×64 se divide primero en cuatro bloques cuadrados de 32×32 (división de bloques de árbol cuádruple). El bloque superior izquierdo y el bloque inferior derecho entre los cuatro bloques de 32×32 se dividen aún más. El bloque superior izquierdo de 32×32 se divide verticalmente en dos bloques rectangulares de 16×32 , y el bloque derecho de 16×32 se divide aún más horizontalmente en dos bloques de 16×16 (división de bloques de árbol binario). El bloque inferior derecho de 32×32 se divide horizontalmente en dos bloques de 32×16 (división de bloques de árbol binario). Como resultado, el bloque inferior izquierdo de 64×64 se divide en el bloque de 16×32 16, dos bloques de 16×16 17 y 18, dos bloques de 32×32 19 y 20, y dos bloques de 32×16 21 y 22.

El bloque inferior derecho de 64×64 23 no se divide.

Como se describió anteriormente, en la FIG. 2, el bloque 10 se divide en 13 bloques de tamaño variable 11 a 23 en base a la división recursiva de bloques de árbol binario y árbol cuádruple. Este tipo de división también se denomina como división de árbol cuádruple más árbol binario (QTBT).

Tenga en cuenta que en la FIG. 2, un bloque se divide en cuatro o dos bloques (división de bloques de árbol binario o árbol cuádruple), pero la división no se limita a este ejemplo. Por ejemplo, un bloque se puede dividir

en tres bloques (división de bloques ternarios). La división que incluye dicha división de bloques ternarios también se denomina como división de árbol multitypo (MBT).

[Sustractor]

5 El sustractor 104 resta una señal de predicción (muestra de predicción) de una señal original (muestra original) por cada bloque dividido por el divisor 102. En otras palabras, el sustractor 104 calcula errores de predicción (también denominados como residuos) de un bloque que se va a codificar (en adelante denominado bloque actual). El sustractor 104 luego envía los errores de predicción calculados al transformador 106.

10 La señal original es una señal que se ingresa al codificador 100, y es una señal que representa una imagen para cada cuadro incluido en un cuadro en movimiento (por ejemplo, una señal de luma y dos señales de croma). En adelante, una señal que representa una imagen también se denominará como muestra.

[Transformador]

15 El transformador 106 transforma los errores de predicción del dominio espacial en coeficientes de transformación del dominio de frecuencia y envía los coeficientes de transformación al cuantificador 108. Más específicamente, el transformador 106 aplica, por ejemplo, una transformada de coseno discreta (DCT) o una transformada de seno discreta (DST) predefinidas a los errores de predicción del dominio espacial.

20 Tenga en cuenta que el transformador 106 puede seleccionar de forma adaptativa un tipo de transformación de entre una pluralidad de tipos de transformación y transformar los errores de predicción en coeficientes de transformación utilizando una función de base de transformación correspondiente al tipo de transformación seleccionado. Este tipo de transformación también se conoce como transformación de núcleo múltiple explícita (EMT) o transformación múltiple adaptativa (AMT).

25 Los tipos de transformación incluyen, por ejemplo, DCT-II, DCT-V, DCT-VIII, DST-I y DST-VII. La FIG. 3 es un gráfico que indica las funciones de base de transformación para cada tipo de transformación. En la FIG. 3, N indica el número de píxeles de entrada. Por ejemplo, la selección de un tipo de transformación de entre la pluralidad de tipos de transformación puede depender del tipo de predicción (predicción intra e predicción inter), y puede depender del modo de predicción intra.

30 La información que indica si se debe aplicar dicha EMT o AMT (denominada como, por ejemplo, un indicador de AMT) y la información que indica el tipo de transformación seleccionado se señalan en el nivel de CU. Tenga en cuenta que la señalización de dicha información no se necesita realizar en el nivel de CU, y se puede realizar en otro nivel (por ejemplo, en el nivel de secuencia, nivel de cuadro, nivel de segmento, nivel de mosaico o nivel de CTU).

35 Más aún, el transformador 106 puede aplicar una transformación secundaria a los coeficientes de transformación (resultado de la transformación). Dicha transformación secundaria también se denomina transformación secundaria adaptativa (AST) o transformación secundaria no separable (NSST). Por ejemplo, el transformador 106 aplica una transformación secundaria a cada subbloque (por ejemplo, cada subbloque 4 x 4) incluido en el bloque de los coeficientes de transformación correspondientes a los errores de predicción intra. La información que indica si se debe aplicar NSST y la información relacionada con la matriz de transformación utilizada en NSST se señalan en el nivel de CU. Tenga en cuenta que la señalización de dicha información no se necesita realizar en el nivel de CU, y se puede realizar en otro nivel (por ejemplo, en el nivel de secuencia, nivel de cuadro, nivel de segmento, nivel de mosaico o nivel de CTU).

40 En la presente memoria, una transformación separable es un procedimiento en el que una transformación se realiza una pluralidad de veces al realizar por separado una transformación para cada dirección de acuerdo con el número de dimensiones de entrada. Una transformación no separable es un procedimiento para realizar una transformación colectiva en el que dos o más dimensiones en una entrada multidimensional se consideran colectivamente como una única dimensión.

45 En un ejemplo de una transformación no separable, cuando la entrada es un bloque de 4 x 4, el bloque de 4 x 4 se considera como un arreglo único que incluye 16 componentes, y la transformación aplica una matriz de transformación de 16 x 16 al arreglo.

50 Más aún, de manera similar a lo anterior, después de que un bloque de entrada 4 x 4 se considera como un arreglo único que incluye 16 componentes, una transformación que realiza una pluralidad de rotaciones de Givens en el arreglo (es decir, una Transformación de Givens de Hipercubo) es también un ejemplo de una transformación no separable.

[Cuantificador]

El cuantificador 108 cuantifica los coeficientes de transformación que salen del transformador 106. Más específicamente, el cuantificador 108 escanea, en un orden de escaneo predeterminado, los coeficientes de

transformación del bloque actual, y cuantifica los coeficientes de transformación escaneados en base a los parámetros de cuantificación (QP) correspondientes a los coeficientes de transformación. El cuantificador 108 envía entonces los coeficientes de transformación cuantificados (en adelante denominados coeficientes cuantificados) del bloque actual al codificador de entropía 110 y al cuantificador inverso 112.

- 5 Un orden predeterminado es un orden para los coeficientes de transformación de cuantificación/cuantificación inversa. Por ejemplo, un orden de escaneo predeterminado se define como un orden ascendente de frecuencia (de baja a alta frecuencia) o un orden descendente de frecuencia (desde alta hasta baja frecuencia).

- 10 Un parámetro de cuantificación es un parámetro que define un tamaño de fase de cuantificación (ancho de cuantificación). Por ejemplo, si el valor del parámetro de cuantificación aumenta, el tamaño de la fase de cuantificación también aumenta. En otras palabras, si aumenta el valor del parámetro de cuantificación, aumenta el error de cuantificación.

[Codificador de entropía]

- 15 El codificador de entropía 110 genera una señal codificada (flujo de bits codificado) mediante la codificación de longitud variable de coeficientes cuantificados, que son entradas del cuantificador 108. Más específicamente, el codificador de entropía 110, por ejemplo, binariza los coeficientes cuantificados y codifica aritméticamente la señal binaria.

[Cuantificador inverso]

- 20 El cuantificador inverso 112 cuantifica de forma inversa los coeficientes cuantificados, que son entradas del cuantificador 108. Más específicamente, el cuantificador inverso 112 cuantifica de forma inversa, en un orden de exploración predeterminado, los coeficientes cuantificados del bloque actual. El cuantificador inverso 112 envía entonces los coeficientes de transformación cuantificados inversos del bloque actual al transformador inverso 114.

[Transformador inverso]

- 25 El transformador inverso 114 restaura los errores de predicción al transformar de forma inversa los coeficientes de transformación, que son entradas del cuantificador inverso 112. Más específicamente, el transformador inverso 114 restaura los errores de predicción del bloque actual al aplicar una transformación inversa correspondiente a la transformación aplicada por el transformador 106 sobre los coeficientes de transformación. El transformador inverso 114 envía entonces los errores de predicción restaurados al sumador 116.

- 30 Tenga en cuenta que, dado que la información se pierde en la cuantificación, los errores de predicción restaurados no coinciden con los errores de predicción calculados por el sustractor 104. En otras palabras, los errores de predicción restaurados incluyen errores de cuantificación.

[Sumador]

- 35 El sumador 116 reconstruye el bloque actual al sumar los errores de predicción, que son entradas del transformador inverso 114, y las muestras de predicción, que son entradas del controlador de predicción 128. A continuación, el sumador 116 envía el bloque reconstruido a la memoria de bloques 118 y al filtro de bucle 120. Un bloque reconstruido también se denomina bloque decodificado local.

[Memoria de bloques]

- 40 La memoria de bloques 118 es un almacenamiento para almacenar bloques en un cuadro que se va a codificar (en adelante denominado cuadro actual) para referencia en la predicción intra. Más específicamente, la memoria de bloques 118 almacena los bloques reconstruidos que genera el sumador 116.

[Filtro de bucle]

- 45 El filtro de bucle 120 aplica un filtro de bucle a los bloques reconstruidos por el sumador 116, y envía los bloques reconstruidos filtrados a la memoria de fotograma 122. Un filtro de bucle es un filtro utilizado en un bucle de codificación (filtro en bucle), e incluye, por ejemplo, un filtro de desbloqueo (DF), un desplazamiento adaptativo de muestra (SAO) y un filtro de bucle adaptativo (ALF).

En ALF, se aplica un filtro de error de mínimos cuadrados para eliminar artefactos de compresión. Por ejemplo, se selecciona un filtro de entre una pluralidad de filtros para cada subbloque 2 x 2 en el bloque actual en base a la dirección y la actividad de los gradientes locales, y se aplica.

- 50 Más específicamente, en primer lugar, cada subbloque (por ejemplo, cada subbloque 2 x 2) se categoriza en una de una pluralidad de clases (por ejemplo, 15 o 25 clases). La clasificación del subbloque se basa en la direccionalidad y la actividad del gradiente. Por ejemplo, el índice de clasificación C se deriva en base a la direccionalidad del gradiente D (por ejemplo, 0 a 2 o 0 a 4) y la actividad del gradiente A (por ejemplo, 0 a 4)

(por ejemplo, $C = 5D + A$). A continuación, en base al índice de clasificación C, cada subbloque se categoriza en una de una pluralidad de clases (por ejemplo, 15 o 25 clases).

Por ejemplo, la direccionalidad de gradiente D se calcula al comparar gradientes de una pluralidad de direcciones (por ejemplo, la dirección horizontal, vertical y dos diagonales). Más aún, por ejemplo, la actividad de gradiente A se calcula al sumar gradientes de una pluralidad de direcciones y cuantificar la suma.

El filtro que se utilizará para cada subbloque se determina de entre la pluralidad de filtros en base al resultado de dicha categorización.

La forma de filtro que se utilizará en ALF es, por ejemplo, una forma de filtro simétrica circular. Las FIG. 4A a FIG. 4C ilustran ejemplos de formas de filtro utilizadas en ALF. La FIG. 4A ilustra un filtro con forma de diamante de 5×5 , la FIG. 4B ilustra un filtro con forma de diamante de 7×7 y la FIG. 4C ilustra un filtro con forma de diamante de 9×9 . La información que indica la forma del filtro se señala a nivel de cuadro. Tenga en cuenta que la señalización de información que indica la forma del filtro no se necesita realizar a nivel de cuadro, y se puede realizar a otro nivel (por ejemplo, a nivel de secuencia, nivel de segmento, nivel de mosaico, nivel de CTU o nivel de CU).

La habilitación o deshabilitación de ALF se determina a nivel de cuadro o nivel de CU. Por ejemplo, para luma, la decisión de aplicar ALF o no se realiza a nivel de CU, y para croma, la decisión de aplicar ALF o no se realiza a nivel de cuadro. La información que indica si ALF está habilitado o deshabilitado se señala a nivel de cuadro o nivel de CU. Tenga en cuenta que la señalización de información que indica si ALF está habilitado o deshabilitado no se necesita realizar a nivel de cuadro o nivel de CU, y se puede realizar a otro nivel (por ejemplo, a nivel de secuencia, nivel de segmento, nivel de mosaico o nivel de CTU).

Los coeficientes establecidos para la pluralidad de filtros seleccionables (por ejemplo, 15 o 25 filtros) se señalizan a nivel de cuadro. Tenga en cuenta que la señalización del conjunto de coeficientes no se necesita realizar en el nivel de cuadro, y se puede realizar en otro nivel (por ejemplo, en el nivel de secuencia, nivel de segmento, nivel de mosaico, nivel de CTU, nivel de CU o nivel de subbloque).

25 [Memoria de fotograma]

La memoria de fotograma 122 es un almacenamiento para almacenar cuadros de referencia utilizados en la predicción inter y también se denomina búfer de fotograma. Más específicamente, la memoria de fotograma 122 almacena bloques reconstruidos filtrados por el filtro de bucle 120.

[Predictor intra]

El predictor intra 124 genera una señal de predicción (señal de predicción intra) mediante la predicción intra del bloque actual con referencia a un bloque o bloques en el cuadro actual y almacenados en la memoria de bloques 118 (también denominada como predicción intra de fotograma). Más específicamente, el predictor intra 124 genera una señal de predicción intra mediante predicción intra con referencia a muestras (por ejemplo, valores de luma y/o croma) de un bloque o bloques vecinos al bloque actual, y luego envía la señal de predicción intra al controlador de predicción 128.

Por ejemplo, el predictor intra 124 realiza la predicción intra al utilizar un modo de entre una pluralidad de modos de predicción intra predefinidos. Los modos de predicción intra incluyen uno o más modos de predicción no direccional y una pluralidad de modos de predicción direccional.

El uno o más modos de predicción no direccional incluyen, por ejemplo, el modo de predicción planar y el modo de predicción DC definidos en el estándar H.265/codificación de vídeo de alta eficiencia (HEVC) (véase NPL 1).

La pluralidad de modos de predicción direccional incluyen, por ejemplo, los 33 modos de predicción direccional definidos en el estándar H.265/HEVC. Tenga en cuenta que la pluralidad de modos de predicción direccional puede incluir además 32 modos de predicción direccional además de los 33 modos de predicción direccional (para un total de 65 modos de predicción direccional). La FIG. 5A ilustra 67 modos de predicción intra utilizados en la predicción intra (dos modos de predicción no direccional y 65 modos de predicción direccional). Las flechas sólidas representan las 33 direcciones definidas en el estándar H.265/HEVC, y las flechas discontinuas representan las 32 direcciones adicionales.

Tenga en cuenta que se puede hacer referencia a un bloque de luma en la predicción intra de un bloque de croma. En otras palabras, se puede predecir un componente de croma del bloque actual en base a un componente de luma del bloque actual. Dicha predicción intra también se conoce como predicción de modelo lineal de componentes cruzados (CCLM). Dicho modo de predicción intra de un bloque de croma que hace referencia a un bloque de luma (denominado, por ejemplo, como modo CCLM) se puede agregar como uno de los modos de predicción intra de un bloque de croma.

El predictor intra 124 puede corregir los valores de píxel posteriores a la predicción intra en base a gradientes de píxel de referencia horizontales/verticales. La predicción intra acompañada de esta clase de corrección también se denomina como combinación de predicción intra dependiente de la posición (PDPC). La información que indica si se debe aplicar PDPC o no (denominada, por ejemplo, como un indicador PDPC) se señala, por ejemplo, en el nivel de CU. Tenga en cuenta que la señalización de esta información no se necesita realizar en el nivel de CU, y se puede realizar en otro nivel (por ejemplo, en el nivel de secuencia, nivel de cuadro, nivel de segmento, nivel de mosaico o nivel de CTU).

[Predictor inter]

El predictor inter 126 genera una señal de predicción (señal de predicción inter) al interpredicir el bloque actual con referencia a un bloque o bloques en un cuadro de referencia, que es diferente del cuadro actual y se almacena en la memoria de fotogramas 122 (también denominada como predicción entre fotogramas). La predicción inter se realiza por bloque actual o por subbloque (por ejemplo, por bloque de 4 x 4) en el bloque actual. Por ejemplo, el predictor inter 126 realiza una estimación de movimiento en un cuadro de referencia para el bloque o subbloque actual. El predictor inter 126 genera entonces una señal de predicción inter del bloque o subbloque actual mediante compensación de movimiento al utilizar la información de movimiento (por ejemplo, un vector de movimiento) obtenida a partir de la estimación de movimiento. El predictor inter 126 envía entonces la señal de predicción inter generada al controlador de predicción 128.

Se señala la información de movimiento utilizada en la compensación de movimiento. Se puede utilizar un predictor de vector de movimiento para la señalización del vector de movimiento. En otras palabras, se puede señalar la diferencia entre el vector de movimiento y el predictor de vector de movimiento.

Tenga en cuenta que la señal de predicción inter se puede generar utilizando información de movimiento para un bloque vecino además de información de movimiento para el bloque actual obtenida a partir de la estimación de movimiento. Más específicamente, la señal de predicción inter se puede generar por el subbloque en el bloque actual al calcular una suma ponderada de una señal de predicción en base a la información de movimiento obtenida a partir de la estimación de movimiento y una señal de predicción en base a la información de movimiento para un bloque vecino. Dicha predicción inter (compensación de movimiento) también se denomina como compensación de movimiento de bloque superpuesto (OBMC).

En dicho modo de OBMC, la información que indica el tamaño del subbloque para OBMC (denominada, por ejemplo, como tamaño de bloque de OBMC) se señala en el nivel de secuencia. Más aún, la información que indica si se debe aplicar o no el modo de OBMC (denominada, por ejemplo, como un indicador de OBMC) se señala en el nivel de CU. Tenga en cuenta que la señalización de dicha información no se necesita realizar en el nivel de secuencia y en el nivel de CU, y se puede realizar en otro nivel (por ejemplo, en el nivel de cuadro, nivel de segmento, nivel de mosaico, nivel de CTU o nivel de subbloque).

A continuación, se describirá el modo de OBMC con más detalle. La FIG. 5B es un diagrama de flujo y la FIG. 5C es un diagrama conceptual para ilustrar un esquema de un procedimiento de corrección de imagen de predicción realizado a través del procesamiento de OBMC.

Primero, se obtiene una imagen de predicción (Pred) a través de una compensación de movimiento típica utilizando un vector de movimiento (MV) asignado al bloque actual.

A continuación, se obtiene una imagen de predicción (Pred_L) al aplicar un vector de movimiento (MV_L) del bloque izquierdo vecino codificado al bloque actual, y se realiza una primera pasada de la corrección de la imagen de predicción al superponer la imagen de predicción y Pred_L.

De manera similar, se obtiene una imagen de predicción (Pred_U) al aplicar un vector de movimiento (MV_U) del bloque superior vecino codificado al bloque actual, y se realiza una segunda pasada de la corrección de la imagen de predicción al superponer la imagen de predicción resultante de la primera pasada y Pred_U. El resultado de la segunda pasada es la imagen de predicción final.

Tenga en cuenta que el ejemplo anterior es de un procedimiento de corrección de dos pasadas que utiliza los bloques vecinos izquierdo y superior, pero el procedimiento puede ser un procedimiento de corrección de tres pasadas o superior que también utiliza el bloque vecino derecho y/o inferior.

Tenga en cuenta que la región sujeta a superposición puede ser la región de píxeles completa del bloque y, alternativamente, puede ser una región de límite de bloque parcial.

Tenga en cuenta que en la presente memoria, el procedimiento de corrección de imagen de predicción se describe como en base a un único cuadro de referencia, pero lo mismo se aplica cuando una imagen de predicción se corrige en base a una pluralidad de cuadros de referencia. En dicho caso, después de obtener imágenes de predicción corregidas resultantes de realizar la corrección en base a cada uno de los cuadros de referencia, las imágenes de predicción corregidas obtenidas se superponen aún más para obtener la imagen de predicción final.

Tenga en cuenta que la unidad del bloque actual puede ser un bloque de predicción y, alternativamente, puede ser un subbloque obtenido al dividir aún más el bloque de predicción.

Un ejemplo de un procedimiento para determinar si se debe implementar el procesamiento de OBMC es al utilizar un `obmc_flag`, que es una señal que indica si se debe implementar el procesamiento de OBMC. Como un ejemplo específico, el codificador determina si el bloque actual pertenece a una región que incluye movimiento complicado. El codificador establece el `obmc_flag` en un valor de "1" cuando el bloque pertenece a una región que incluye movimiento complicado e implementa el procesamiento de OBMC al codificar, y establece el `obmc_flag` en un valor de "0" cuando el bloque no pertenece a una región que incluye movimiento complicado y codifica sin implementar el procesamiento de OBMC. El decodificador intercambia entre implementar el procesamiento de OBMC o no al decodificar el `obmc_flag` escrito en el flujo y al realizar la decodificación de acuerdo con el valor del indicador.

Tenga en cuenta que la información de movimiento se puede derivar en el lado del decodificador sin ser señalizada. Por ejemplo, se puede utilizar un modo de fusión definido en el estándar H.265/HEVC. Más aún, por ejemplo, la información de movimiento se puede derivar al realizar una estimación de movimiento en el lado del decodificador. En este caso, la estimación de movimiento se realiza sin utilizar los valores de píxel del bloque actual.

En la presente memoria, se describirá un modo para realizar una estimación de movimiento en el lado del decodificador. Un modo para realizar una estimación de movimiento en el lado del decodificador también se denomina modo de derivación de vector de movimiento coincidente con patrón (PMMVD) o modo de conversión ascendente de la velocidad del fotograma (FRUC).

Un ejemplo de procesamiento de FRUC se ilustra en la FIG. 5D. En primer lugar, se genera una lista de candidatos (una lista de candidatos puede ser una lista de fusión) de candidatos que incluyen cada uno un predictor de vector de movimiento con referencia a vectores de movimiento de bloques codificados que son vecinos espacial o temporalmente del bloque actual. A continuación, se selecciona el mejor MV candidato de entre una pluralidad de MV candidatos registrados en la lista de candidatos. Por ejemplo, se calculan los valores de evaluación para los candidatos incluidos en la lista de candidatos y se selecciona un candidato en base a los valores de evaluación calculados.

A continuación, se deriva un vector de movimiento para el bloque actual a partir del vector de movimiento del candidato seleccionado. Más específicamente, por ejemplo, el vector de movimiento para el bloque actual se calcula como el vector de movimiento del candidato seleccionado (mejor MV candidato), tal como está. Alternativamente, el vector de movimiento para el bloque actual se puede derivar mediante una coincidencia de patrones realizada en la vecindad de una posición en un cuadro de referencia correspondiente al vector de movimiento del candidato seleccionado. En otras palabras, cuando se busca la vecindad del mejor MV candidato a través del mismo procedimiento y se encuentra un MV que tiene un mejor valor de evaluación, el mejor MV candidato se puede actualizar al MV que tiene el mejor valor de evaluación, y el MV que tiene el mejor valor de evaluación se puede utilizar como el MV final para el bloque actual. Tenga en cuenta que también es aceptable una configuración en la que no se implementa este procesamiento.

Los mismos procedimientos se pueden realizar en casos en los que el procesamiento se realiza en unidades de subbloques.

Tenga en cuenta que un valor de evaluación se calcula al calcular la diferencia en la imagen reconstruida mediante la coincidencia de patrones realizada entre una región en un cuadro de referencia correspondiente a un vector de movimiento y una región predeterminada. Tenga en cuenta que el valor de evaluación se puede calcular al utilizar alguna otra información además de la diferencia.

La coincidencia de patrones utilizada es la primera coincidencia de patrones o la segunda coincidencia de patrones. La primera coincidencia de patrones y la segunda coincidencia de patrones también se denominan coincidencia bilateral y coincidencia de plantilla, respectivamente.

En la primera coincidencia de patrones, la coincidencia de patrones se realiza entre dos bloques a lo largo de la trayectoria de movimiento del bloque actual en dos cuadros de referencia diferentes. Por lo tanto, en la primera coincidencia de patrones, una región en otro cuadro de referencia que se ajusta a la trayectoria de movimiento del bloque actual se utiliza como la región predeterminada para el cálculo descrito anteriormente del valor de evaluación candidato.

La FIG. 6 es para ilustrar un ejemplo de coincidencia de patrones (coincidencia bilateral) entre dos bloques a lo largo de una trayectoria de movimiento. Como se ilustra en la FIG. 6, en la primera coincidencia de patrones, se obtienen dos vectores de movimiento (MV0, MV1) al encontrar la mejor coincidencia entre dos bloques a lo largo de la trayectoria de movimiento del bloque actual (bloque Cur) en dos cuadros de referencia diferentes (Ref0, Ref1). Más específicamente, se puede obtener una diferencia entre (i) una imagen reconstruida en una posición especificada en un primer cuadro de referencia codificado (Ref0) especificada por un MV candidato y (ii) un cuadro reconstruido en una posición especificada en un segundo cuadro de referencia codificado (Ref1)

especificada por un MV simétrico escalado en un intervalo de tiempo de visualización del MV candidato, y se puede calcular el valor de evaluación para el bloque actual al utilizar la diferencia obtenida. El MV candidato que tiene el mejor valor de evaluación entre la pluralidad de MV candidatos se puede seleccionar como el MV final

5 Bajo el supuesto de una trayectoria de movimiento continua, los vectores de movimiento (MV0, MV1) que apuntan a los dos bloques de referencia serán proporcionales a las distancias temporales (TD0, TD1) entre el cuadro actual (Cur Pic) y los dos cuadros de referencia (Ref0, Ref1). Por ejemplo, cuando el cuadro actual está temporalmente entre los dos cuadros de referencia y la distancia temporal desde el cuadro actual hasta los dos cuadros de referencia es la misma, la primera coincidencia de patrones deriva un vector de movimiento
10 bidireccional en base a espejo.

En la segunda coincidencia de patrones, la coincidencia de patrones se realiza entre una plantilla en el cuadro actual (bloques vecinos del bloque actual en el cuadro actual (por ejemplo, los bloques vecinos superiores y/o izquierdos)) y un bloque en un cuadro de referencia. Por lo tanto, en la segunda coincidencia de patrones, un bloque vecino al bloque actual en el cuadro actual se utiliza como la región predeterminada para el cálculo
15 descrito anteriormente del valor de evaluación del candidato.

La FIG. 7 es para ilustrar un ejemplo de coincidencia de patrones (coincidencia de plantilla) entre una plantilla en el cuadro actual y un bloque en un cuadro de referencia. Como se ilustra en la FIG. 7, en la segunda coincidencia de patrones, un vector de movimiento del bloque actual se deriva mediante la búsqueda de un cuadro de referencia (RefO) para encontrar el bloque que mejor coincide con los bloques vecinos del bloque actual (bloque Cur) en el cuadro actual (Cur Pic). Más específicamente, se puede derivar una diferencia entre
20 (i) una cuadro reconstruido de una región codificada que es ambas o una de las regiones vecinas izquierda y superior y (ii) un cuadro reconstruido en la misma posición en un cuadro de referencia codificada (RefO) especificada por un MV candidato, y el valor de evaluación para el bloque actual se puede calcular al utilizar la diferencia derivada. El MV candidato que tiene el mejor valor de evaluación entre la pluralidad de MV candidatos se puede seleccionar como el mejor MV candidato.
25

La información que indica si se debe aplicar o no el modo de FRUC (denominada, por ejemplo, un indicador de FRUC) se señala en el nivel de CU. Más aún, cuando se aplica el modo de FRUC (por ejemplo, cuando el indicador de FRUC se establece en verdadero), la información que indica el procedimiento de coincidencia de patrones (primera coincidencia de patrones o segunda coincidencia de patrones) se señala en el nivel de CU. Tenga en cuenta que la señalización de dicha información no se necesita realizar en el nivel de CU, y se puede
30 realizar en otro nivel (por ejemplo, en el nivel de secuencia, nivel de cuadro, nivel de segmento, nivel de mosaico, nivel de CTU o nivel de subbloque).

En la presente memoria, se describirá un modo para derivar un vector de movimiento en base a un modelo que supone un movimiento lineal uniforme. Este modo también se denomina modo de flujo óptico bidireccional (BIO).
35

La FIG. 8 es para ilustrar un modelo que supone un movimiento lineal uniforme. En la FIG. 8, (v_x, v_y) denota un vector de velocidad, y t_0 y t_1 denotan distancias temporales entre el cuadro actual (Cur Pic) y dos cuadros de referencia (Ref0, Ref1). (MV_{x0}, MV_{y0}) denota un vector de movimiento correspondiente al cuadro de referencia Ref0, y (MV_{x1}, MV_{y1}) denota un vector de movimiento correspondiente al cuadro de referencia Ref1.

40 En la presente memoria, bajo el supuesto de movimiento lineal uniforme exhibido por el vector de velocidad (v_x, v_y) , (MV_{x0}, MV_{y0}) y (MV_{x1}, MV_{y1}) se representan como $(v_x t_0, v_y t_0)$ y $(-v_x t_1, -v_y t_1)$, respectivamente, y se proporciona la siguiente ecuación de flujo óptico.

MAT. 1

$$\partial I^{(k)} / \partial t + v_x \partial I^{(k)} / \partial x + v_y \partial I^{(k)} / \partial y = 0. \quad (I)$$

45 En la presente memoria, $I^{(k)}$ denota un valor de luma del cuadro de referencia k ($k = 0, 1$) después de la compensación de movimiento. Esta ecuación de flujo óptico muestra que la suma de (i) la derivada temporal del valor de luma, (ii) el producto de la velocidad horizontal y el componente horizontal del gradiente espacial de un cuadro de referencia, y (iii) el producto de la velocidad vertical y el componente vertical del gradiente espacial de un cuadro de referencia es igual a cero. Un vector de movimiento de cada bloque obtenido a partir de, por ejemplo, una lista de fusión se corrige píxel por píxel en base a una combinación de la ecuación de flujo
50 óptico y la interpolación de Hermite.

Tenga en cuenta que un vector de movimiento se puede derivar en el lado del decodificador utilizando un procedimiento distinto al de derivar un vector de movimiento en base a un modelo que supone un movimiento lineal uniforme. Por ejemplo, se puede derivar un vector de movimiento para cada subbloque en base a vectores
55 de movimiento de bloques vecinos.

En la presente memoria, se describirá un modo en el que se deriva un vector de movimiento para cada subbloque en base a vectores de movimiento de bloques vecinos. Este modo también se conoce como modo de predicción de compensación de movimiento afin.

La FIG. 9A es para ilustrar la derivación de un vector de movimiento de cada subbloque en base a vectores de movimiento de bloques vecinos. En la FIG. 9A, el bloque actual incluye 16 subbloques de 4 x 4. En la presente memoria, el vector de movimiento v_0 del punto de control de la esquina superior izquierda en el bloque actual se deriva en base a vectores de movimiento de subbloques vecinos, y el vector de movimiento v_1 del punto de control de la esquina superior derecha en el bloque actual se deriva en base a vectores de movimiento de bloques vecinos. Luego, utilizando los dos vectores de movimiento v_0 y v_1 , el vector de movimiento (v_x, v_y) de cada subbloque en el bloque actual se deriva utilizando la Ecuación 2 a continuación.

MAT. 2

$$\begin{cases} v_x = \frac{(v_{1x} - v_{0x})}{w}x - \frac{(v_{1y} - v_{0y})}{w}y + v_{0x} \\ v_y = \frac{(v_{1y} - v_{0y})}{w}x + \frac{(v_{1x} - v_{0x})}{w}y + v_{0y} \end{cases} \quad (2)$$

En la presente memoria, x e y son las posiciones horizontal y vertical del subbloque, respectivamente, y w es un coeficiente ponderado predeterminado.

Dicho modo de predicción de compensación de movimiento afin puede incluir una serie de modos de diferentes procedimientos de derivación de los vectores de movimiento de los puntos de control de las esquinas superior izquierda y superior derecha. La información que indica dicho modo de predicción de compensación de movimiento afin de este tipo (denominada, por ejemplo, como un indicador afin) se señala en el nivel de CU. Tenga en cuenta que la señalización de información que indica el modo de predicción de compensación de movimiento afin no se necesita realizar en el nivel de CU, y se puede realizar en otro nivel (por ejemplo, en el nivel de secuencia, nivel de cuadro, nivel de segmento, nivel de mosaico, nivel de CTU o nivel de subbloque).

[Controlador de predicción]

El controlador de predicción 128 selecciona la señal de predicción intra o la señal de predicción inter, y envía la señal de predicción seleccionada al sustractor 104 y al sumador 116.

En la presente memoria, se dará un ejemplo de derivación de un vector de movimiento a través del modo de fusión en un cuadro actual. La FIG. 9B es para ilustrar un esquema de un procedimiento para derivar un vector de movimiento a través del modo de fusión.

Primero, se genera una lista de predictores de MV en la que se registran los predictores de MV candidatos. Los ejemplos de predictores de MV candidatos incluyen: predictores de MV espacialmente vecinos, que son MV de bloques codificados ubicados en la vecindad espacial del bloque actual; un predictor de MV temporalmente vecino, que es un MV de un bloque en un cuadro de referencia codificada que es vecino de un bloque en la misma ubicación que el bloque actual; un predictor de MV combinado, que es un MV generado al combinar los valores MV del predictor de MV vecino espacialmente y el predictor de MV vecino temporalmente; y un predictor de MV cero, que es un MV cuyo valor es cero.

A continuación, el MV del bloque actual se determina al seleccionar un predictor de MV de entre la pluralidad de predictores de MV registrados en la lista de predictores de MV.

Además, en el codificador de longitud variable, un `merge_idx`, que es una señal que indica qué predictor de MV está seleccionado, se escribe y codifica en el flujo.

Tenga en cuenta que los predictores de MV registrados en la lista de predictores de MV ilustrada en la FIG. 9B constituyen un ejemplo. El número de predictores de MV registrados en la lista de predictores de MV puede ser diferente del número ilustrado en la FIG. 9B, los predictores de MV registrados en la lista de predictores de MV pueden omitir uno o más de los tipos de predictores de MV dados en el ejemplo de la FIG. 9B, y los predictores de MV registrados en la lista de predictores de MV pueden incluir uno o más tipos de predictores de MV además de y diferentes de los tipos dados en el ejemplo de la FIG. 9B.

Tenga en cuenta que el MV final se puede determinar al realizar un procesamiento de DMVR (que se describirá más adelante) al utilizar el MV del bloque actual derivado a través del modo de fusión.

En la presente memoria, se dará un ejemplo de determinación de un MV utilizando el procesamiento de DMVR.

La FIG. 9C es un diagrama conceptual para ilustrar un esquema del procesamiento de DMVR.

En primer lugar, se considera que el conjunto de MVP más apropiado para el bloque actual es el MV candidato, se obtienen píxeles de referencia a partir de un primer cuadro de referencia, que es un cuadro procesado en la dirección L0 de acuerdo con el MV candidato, y un segundo cuadro de referencia, que es un cuadro procesado en la dirección L1 de acuerdo con el MV candidato, y se genera una plantilla al calcular el promedio de los píxeles de referencia.

A continuación, utilizando la plantilla, se buscan las regiones circundantes de los MV candidatos del primer y segundo cuadros de referencia, y se determina que el MV con el menor coste es el MV final. Tenga en cuenta que el valor del coste se calcula utilizando, por ejemplo, la diferencia entre cada valor de píxel en la plantilla y cada valor de píxel en las regiones buscadas, así como el valor del MV.

Tenga en cuenta que los esquemas de los procedimientos descritos en la presente memoria son fundamentalmente los mismos tanto en el codificador como en el decodificador.

Tenga en cuenta que se puede utilizar un procesamiento distinto del procesamiento exactamente como se ha descrito anteriormente, siempre que el procesamiento sea capaz de derivar el MV final mediante la búsqueda de los alrededores del MV candidato.

En la presente memoria, se dará un ejemplo de un modo que genera una imagen de predicción a utilizar el procesamiento de LIC.

La FIG. 9D es para ilustrar un esquema de un procedimiento de generación de imagen de predicción mediante el uso de un procedimiento de corrección de luminancia realizado mediante el procesamiento de LIC.

En primer lugar, se extrae un MV para obtener, a partir de un cuadro de referencia codificada, una imagen de referencia correspondiente al bloque actual.

A continuación, se extrae información que indica cómo cambió el valor de luminancia entre el cuadro de referencia y el cuadro actual y se calcula un parámetro de corrección de luminancia al utilizar los valores de píxel de luminancia para la región de referencia vecina izquierda codificada y la región de referencia vecina superior codificada, y el valor de píxel de luminancia en la misma ubicación en el cuadro de referencia especificada por el MV.

La imagen de predicción para el bloque actual se genera al realizar un procedimiento de corrección de luminancia al utilizar el parámetro de corrección de luminancia en la imagen de referencia en el cuadro de referencia especificada por el MV.

Tenga en cuenta que la forma de la región de referencia circundante ilustrada en la FIG. 9D es solo un ejemplo; la región de referencia circundante puede tener una forma diferente.

Más aún, aunque en este ejemplo se genera una imagen de predicción a partir de un único cuadro de referencia, en los casos en los que también se genera una imagen de predicción a partir de una pluralidad de cuadros de referencia, la imagen de predicción se genera después de realizar un procedimiento de corrección de luminancia, a través del mismo procedimiento, en las imágenes de referencia obtenidas a partir de los cuadros de referencia.

Un ejemplo de un procedimiento para determinar si se debe implementar el procesamiento de LIC es al utilizar un lic_flag, que es una señal que indica si se debe implementar el procesamiento de LIC. Como un ejemplo específico, el codificador determina si el bloque actual pertenece a una región de cambio de luminancia. El codificador establece el lic_flag en un valor de "1" cuando el bloque pertenece a una región de cambio de luminancia e implementa el procesamiento de LIC al codificar, y establece el lic_flag en un valor de "0" cuando el bloque no pertenece a una región de cambio de luminancia y codifica sin implementar el procesamiento de LIC. El decodificador intercambia entre implementar o no el procesamiento de LIC decodificando el lic_flag escrito en el flujo y al realizar la decodificación de acuerdo con el valor del indicador.

Un ejemplo de un procedimiento diferente para determinar si se debe implementar el procesamiento de LIC es determinarlo de acuerdo con si se determinó que el procesamiento de LIC se implementaría para un bloque circundante. En un ejemplo específico, cuando se utiliza el modo de fusión en el bloque actual, se puede determinar si se aplicó el procesamiento de LIC en la codificación del bloque codificado circundante seleccionado al derivar el MV en el procesamiento del modo de fusión, y se puede intercambiar si se debe implementar o no el procesamiento de LIC en base al resultado de la determinación. Tenga en cuenta que en este ejemplo, lo mismo se aplica al procesamiento realizado en el lado del decodificador.

[Esquema del decodificador]

A continuación, se describirá un decodificador capaz de decodificar una señal codificada (flujo de bits codificado) enviada por el codificador 100. La FIG. 10 es un diagrama de bloques que ilustra una configuración

funcional del decodificador 200 de acuerdo con la Realización 1. El decodificador 200 es un decodificador de cuadro en movimiento/cuadro que decodifica un cuadro en movimiento/cuadro bloque por bloque.

5 Como se ilustra en la FIG. 10, el decodificador 200 incluye un decodificador de entropía 202, cuantificador inverso 204, un transformador inverso 206, sumador 208, memoria de bloques 210, filtro de bucle 212, memoria de fotogramas 214, predictor intra 216, predictor inter 218 y controlador de predicción 220.

10 El decodificador 200 se realiza, por ejemplo, como un procesador y una memoria genéricos. En este caso, cuando un programa de software almacenado en la memoria es ejecutado por el procesador, el procesador funciona como decodificador de entropía 202, cuantificador inverso 204, transformador inverso 206, sumador 208, filtro de bucle 212, predictor intra 216, predictor inter 218 y controlador de predicción 220. Alternativamente, el decodificador 200 se puede realizar como uno o más circuitos electrónicos dedicados correspondientes al decodificador de entropía 202, cuantificador inverso 204, transformador inverso 206, sumador 208, filtro de bucle 212, predictor intra 216, predictor inter 218 y controlador de predicción 220.

A continuación, se describirá cada componente incluido en el decodificador 200.

[Decodificador de entropía]

15 El decodificador de entropía 202 decodifica por entropía un flujo de bits codificado. Más específicamente, por ejemplo, el decodificador de entropía 202 decodifica aritméticamente un flujo de bits codificado en una señal binaria. A continuación, el decodificador de entropía 202 desbinariza la señal binaria. Con esto, el decodificador de entropía 202 envía coeficientes cuantificados de cada bloque al cuantificador inverso 204.

[Cuantificador inverso]

20 El cuantificador inverso 204 cuantifica inversamente los coeficientes cuantificados de un bloque que se va a decodificar (en adelante denominado como bloque actual), que son entradas del decodificador de entropía 202. Más específicamente, el cuantificador inverso 204 cuantifica inversamente los coeficientes cuantificados del bloque actual en base a parámetros de cuantificación correspondientes a los coeficientes cuantificados. A continuación, el cuantificador inverso 204 envía los coeficientes cuantificados inversos (es decir, coeficientes de transformación) del bloque actual al transformador inverso 206.

[Transformador inverso]

El transformador inverso 206 restaura los errores de predicción mediante la transformación inversa de los coeficientes de transformación, que son entradas del cuantificador inverso 204.

30 Por ejemplo, cuando la información analizada a partir de un flujo de bits codificado indica la aplicación de EMT o AMT (por ejemplo, cuando el indicador de AMT está establecido en verdadero), el transformador inverso 206 transforma inversamente los coeficientes de transformación del bloque actual en base a la información que indica el tipo de transformación analizada.

35 Más aún, por ejemplo, cuando la información analizada a partir de un flujo de bits codificado indica la aplicación de NSST, el transformador inverso 206 aplica una transformación inversa secundaria a los coeficientes de transformación.

[Sumador]

40 El sumador 208 reconstruye el bloque actual al sumar los errores de predicción, que son entradas del transformador inverso 206, y las muestras de predicción, que son una entrada del controlador de predicción 220. A continuación, el sumador 208 envía el bloque reconstruido a la memoria de bloques 210 y al filtro de bucle 212.

[Memoria de bloques]

45 La memoria de bloques 210 es un almacenamiento para almacenar bloques en un cuadro que se va a decodificar (en adelante denominada como cuadro actual) para referencia en la predicción intra. Más específicamente, la memoria de bloques 210 almacena los bloques reconstruidos que se envían desde el sumador 208.

[Filtro de bucle]

El filtro de bucle 212 aplica un filtro de bucle a los bloques reconstruidos por el sumador 208, y envía los bloques reconstruidos filtrados a la memoria de fotogramas 214 y, por ejemplo, a un dispositivo de visualización.

50 Cuando la información que indica la habilitación o deshabilitación de ALF analizada a partir de un flujo de bits codificado indica que está habilitada, se selecciona un filtro de entre una pluralidad de filtros en base a la dirección y la actividad de los gradientes locales, y el filtro seleccionado se aplica al bloque reconstruido.

[Memoria de fotogramas]

La memoria de fotogramas 214 es un almacenamiento para almacenar cuadros de referencia utilizadas en la predicción inter y también se denomina búfer de fotogramas. Más específicamente, la memoria de fotograma 214 almacena bloques reconstruidos filtrados por el filtro de bucle 212.

5 [Predictor Intra]

El predictor intra 216 genera una señal de predicción (señal de predicción intra) mediante predicción intra con referencia a un bloque o bloques en el cuadro actual y almacenada en la memoria de bloques 210. Más específicamente, el predictor intra 216 genera una señal de predicción intra mediante predicción intra con referencia a muestras (por ejemplo, valores de luma y/o croma) de un bloque o bloques vecinos al bloque actual, y luego emite la señal de predicción intra al controlador de predicción 220.

Tenga en cuenta que cuando se selecciona un modo de predicción intra en el que se intrapredice un bloque de croma a partir de un bloque de luma, el predictor intra 216 puede predecir el componente de croma del bloque actual en base al componente de luma del bloque actual.

15 Más aún, cuando se analiza la información que indica la aplicación de PDPC a partir de un flujo de bits codificado, el predictor intra 216 corrige los valores de píxel posteriores a la predicción intra en base a gradientes de píxeles de referencia horizontales/verticales.

[Predictor inter]

20 El predictor inter 218 predice el bloque actual con referencia a un cuadro de referencia almacenada en la memoria de fotogramas 214. La predicción inter se realiza por bloque actual o por subbloque (por ejemplo, por bloque 4 x 4) en el bloque actual. Por ejemplo, el predictor inter 218 genera una señal de predicción inter del bloque o subbloque actual mediante compensación de movimiento utilizando información de movimiento (por ejemplo, un vector de movimiento) analizada a partir de un flujo de bits codificado, y envía la señal de predicción inter al controlador de predicción 220.

25 Tengas en cuenta que cuando la información analizada a partir del flujo de bits codificado indica la aplicación del modo de OBMC, el predictor inter 218 genera la señal de predicción inter utilizando información de movimiento para un bloque vecino además de información de movimiento para el bloque actual obtenida a partir de la estimación de movimiento.

30 Más aún, cuando la información analizada a partir del flujo de bits codificado indica la aplicación del modo de FRUC, el predictor inter 218 deriva información de movimiento al realizar una estimación de movimiento de acuerdo con el procedimiento de coincidencia de patrones (coincidencia bilateral o coincidencia de plantilla) analizado a partir del flujo de bits codificado. El predictor inter 218 realiza entonces una compensación de movimiento utilizando la información de movimiento derivada.

35 Más aún, cuando se va a aplicar el modo BIO, el predictor inter 218 deriva un vector de movimiento en base a un modelo que supone un movimiento lineal uniforme. Más aún, cuando la información analizada a partir del flujo de bits codificado indica que se va a aplicar el modo de predicción de compensación de movimiento afín, el predictor inter 218 deriva un vector de movimiento de cada subbloque en base a los vectores de movimiento de los bloques vecinos.

[Controlador de predicción]

40 El controlador de predicción 220 selecciona la señal de predicción intra o la señal de predicción inter, y envía la señal de predicción seleccionada al sumador 208.

[Primer ejemplo de procesamiento de predicción inter]

La FIG. 11 es un diagrama que ilustra un primer ejemplo del esquema de una configuración de canalización utilizada para el decodificador 200. Esta configuración de canalización incluye cuatro etapas, a saber, una primera etapa, una segunda etapa, una tercera etapa y una cuarta etapa.

45 En la primera etapa, el decodificador 200 realiza una decodificación de entropía en un flujo de entrada que se va a decodificar, y obtiene la información necesaria para la decodificación (S101).

50 En la segunda etapa, el decodificador 200 deriva, utilizando la información, un vector de movimiento (MV) que se va a utilizar en el procesamiento de predicción inter. Específicamente, en primer lugar, con referencia a un bloque decodificado vecino, el decodificador 200 deriva uno o más predictores de vector de movimiento (en adelante denominados como MVP) que son candidatos de vector de movimiento (S102). A continuación, el decodificador 200 transfiere una imagen de referencia a la memoria de acuerdo con uno o más MVP derivados (S103).

A continuación, cuando el modo de predicción inter es el modo de FRUC, el decodificador 200 determina un vector de movimiento al determinar el MVP más apropiado (S104) y al realizar una búsqueda en la vecindad del MVP más apropiado (S105). Cuando el modo de predicción inter es el modo de fusión, el decodificador 200 determina un vector de movimiento al realizar el procesamiento de DMVR (S106).

- 5 En la tercera etapa, el decodificador 200 decodifica una imagen residual al realizar una cuantificación inversa y un procesamiento de transformación inversa (S110). Cuando el bloque actual es un bloque intra, el decodificador 200 decodifica una imagen de predicción al realizar un procesamiento de predicción intra (S108). Cuando el bloque actual es un bloque inter, el decodificador 200 decodifica una imagen de predicción al realizar un procesamiento de compensación de movimiento, etc., utilizando el vector de movimiento derivado en la
10 segunda etapa (S107).

A continuación, el decodificador 200 selecciona una de las imágenes de predicción generadas a través del procesamiento de predicción intra y la imagen de predicción generada a través del procesamiento de predicción inter (S109), y genera una imagen reconstruida al agregar la imagen residual y la imagen de predicción seleccionada (S111).

- 15 En la cuarta etapa, el decodificador 200 genera una imagen decodificada al realizar un filtrado de bucle en la imagen reconstruida (S112).

- Dado que el vector de movimiento derivado en la segunda etapa se utiliza como un vector de movimiento de referencia circundante para derivar uno o más MVP en el procesamiento de decodificación en un bloque posterior, el vector de movimiento derivado en la segunda etapa se retroalimenta como una entrada para el
20 procesamiento de derivación de MVP (S102). Este procesamiento de retroalimentación debe realizarse en una etapa para hacer referencia a un vector de movimiento que pertenece a un bloque inmediatamente anterior al bloque actual en el orden de procesamiento. Como resultado, se realiza una gran cantidad de procesamiento en la segunda etapa como se ilustra en la FIG. 11, y la segunda etapa, por lo tanto, requiere una larga duración de procesamiento.

- 25 Tenga en cuenta que este esquema de la configuración de canalización es un solo ejemplo; se puede eliminar una parte del procesamiento descrito, se puede agregar un procesamiento no descrito o se pueden modificar los límites de las etapas.

- La FIG. 12 es un diagrama esquemático que ilustra un ejemplo de partición de bloques utilizado en la descripción del procesamiento de canalización. El ejemplo de partición de bloques ilustrado en la FIG. 12
30 muestra dos unidades de árbol de codificación. Una unidad de árbol de codificación incluye dos unidades de codificación CU0 y CU1, y la otra unidad de árbol de codificación incluye tres unidades de codificación CU2, CU3 y CU4.

- Las unidades de codificación CU0, CU1 y CU4 tienen el mismo tamaño. Las unidades de codificación CU2 y CU3 tienen el mismo tamaño. El tamaño de cada una de las unidades de codificación CU0, CU1 y CU4 es el
35 doble del tamaño de cada una de las unidades de codificación CU2 y CU3.

- La FIG. 13 ilustra, en series de tiempo, la temporización de procesamiento de las respectivas etapas para cada bloque que se va a decodificar, en el primer ejemplo del esquema de la configuración de canalización descrita con referencia a la FIG. 11. La FIG. 13 ilustra la temporización de procesamiento para cinco bloques que se
40 van a decodificar, a saber, las unidades de codificación CU0 a CU4 ilustradas en la FIG. 12. S1 a S4 en la FIG. 13 indican las duraciones de procesamiento de las etapas primera a cuarta ilustradas en la FIG. 11.

Dado que las unidades de codificación CU0, CU1 y CU4 son el doble de grandes que las unidades de codificación CU2 y CU3, las duraciones de procesamiento de las respectivas etapas para las unidades de codificación CU0, CU1 y CU4 también son el doble de grandes.

- 45 Dado que la segunda etapa tarda mucho tiempo en procesarse como se describe con referencia a la FIG. 11, la duración de procesamiento de la segunda etapa es el doble de larga que cada una de las duraciones de procesamiento de las otras etapas.

- El procesamiento de cada etapa comienza después del procesamiento de la misma etapa para el bloque inmediatamente anterior a la finalización del bloque actual en el orden de procesamiento. Por ejemplo, el
50 procesamiento de la segunda etapa para la unidad de codificación CU1 comienza en el tiempo t6 en el que finaliza la segunda etapa para la unidad de codificación CU0. En este momento, dado que la duración del procesamiento de la segunda etapa para la unidad de codificación CU0 es el doble de larga que la duración del procesamiento de la primera etapa para la unidad de codificación CU1, la unidad de codificación CU1 tiene un tiempo de espera desde el tiempo t4 en el que finaliza el procesamiento de la primera etapa hasta el tiempo t6 en el que comienza el procesamiento de la segunda etapa.

- 55 De esta manera, siempre hay un tiempo de espera antes de que comience la segunda etapa, y el tiempo de espera se acumula cada vez que se realiza el procesamiento para los bloques actuales. Como consecuencia,

la unidad de codificación CU4 tiene un tiempo de espera desde el tiempo t8 en el que finaliza el procesamiento de la primera etapa hasta el tiempo t14 en el que comienza el procesamiento de la segunda etapa.

Como resultado, en el momento de la finalización del procesamiento de decodificación en un cuadro, la duración del procesamiento, que incluye el tiempo de espera, se vuelve aproximadamente el doble de larga que la duración del procesamiento originalmente prevista. Esto puede dificultar la finalización del procesamiento en todos los bloques dentro de la duración del procesamiento asignada a un cuadro.

La FIG. 14 es un diagrama de flujo del procesamiento de predicción inter en el primer ejemplo del esquema de la configuración de canalización descrita con referencia a la FIG. 11. El procesamiento ilustrado en la FIG. 14 se realiza repetidamente para cada bloque de predicción, que es la unidad de procesamiento del procesamiento de predicción de fotogramas inter. El procesamiento ilustrado en la FIG. 14 se realiza mediante el codificador 100 y el decodificador 200. Aunque lo siguiente describe principalmente las operaciones del predictor inter 126 incluido en el codificador 100, lo mismo se aplica a las operaciones del predictor inter 218 incluido en el decodificador 200.

El predictor inter 126 selecciona un modo de predicción inter para un bloque actual, es decir, un bloque actual que se va a codificar o decodificar, de entre una pluralidad de modos (modo inter normal, modo de fusión, modo de FRUC, etc.). El predictor inter 126 deriva un vector de movimiento (MV) utilizando el modo de predicción inter seleccionado. Específicamente, el modo de predicción inter para el bloque actual se indica mediante la información del modo de predicción inter.

Cuando se utiliza el modo inter normal (modo inter normal en S201), el predictor inter 126 hace referencia a los vectores de movimiento de los bloques procesados vecinos y obtiene una pluralidad de predictores de vector de movimiento (MVP), y crea una lista de MVP normal que indica la pluralidad de MVP obtenidos. El predictor inter 126 especifica un MVP de la pluralidad de MVP indicados en la lista de MVP normal creada, y añade una diferencia de vector de movimiento (MVD) al MVP especificado para determinar un vector de movimiento final (S202). Específicamente, el codificador 100 genera la diferencia de vector de movimiento a partir del vector de movimiento y el MVP, y transmite la diferencia de vector de movimiento generada al decodificador 200. El decodificador 200 obtiene un vector de movimiento al agregar la diferencia de vector de movimiento recibida al predictor de vector de movimiento.

Cuando se utiliza el modo de fusión (modo de fusión en S201), el predictor inter 126 hace referencia a los vectores de movimiento de bloques procesados vecinos y obtiene uno o más MVP, y crea una lista de MVP de fusión que indica el uno o más MVP obtenidos. A continuación, el predictor inter 126 especifica un MVP de la lista de MVP de fusión creada como el MVP más apropiado (S203). A continuación, el predictor inter 126 determina un vector de movimiento final al realizar un procesamiento de DMVR en el que se busca una posición con el valor de coste más bajo en la vecindad del MVP más apropiado, utilizando un cuadro procesado (S204).

Cuando se utiliza el modo de FRUC (FRUC en S201), el predictor inter 126 hace referencia a vectores de movimiento de bloques procesados vecinos y obtiene una pluralidad de MVP, y crea una lista de MVP de FRUC que indica la pluralidad de MVP obtenidos (S205). A continuación, el predictor inter 126 deriva el MVP más apropiado con el valor de coste más bajo de la pluralidad de MVP indicados en la lista de MVP de FRUC, utilizando un procedimiento de coincidencia bilateral o un procedimiento de coincidencia de plantilla (S206). A continuación, el predictor inter 126 realiza además el mismo procesamiento para buscar una posición con el coste más bajo en la vecindad del MVP más apropiado derivado, y determina, como un vector de movimiento final, el vector de movimiento obtenido a través de la búsqueda (S207).

El vector de movimiento final derivado utilizando cada procedimiento se utiliza como el MV de referencia circundante para derivar uno o más MVP para un bloque posterior, y por tanto se almacena en la memoria de MV de referencia circundante.

Por último, el predictor inter 126 genera una imagen de predicción al realizar un procesamiento de compensación de movimiento, etc., utilizando el vector de movimiento final (S208).

Como se puede entender a partir de lo anterior, en el caso de procesar el bloque actual en el modo de fusión o el modo de FRUC, se requieren muchas más fases para derivar el vector de movimiento final en comparación con el caso de procesamiento en el otro modo. Esto prolonga la duración del procesamiento, lo que provoca un aumento en el tiempo de espera para esperar la siguiente etapa en el control de canalización descrito con referencia a la FIG. 13.

Tenga en cuenta que el flujo de procesamiento ilustrado en la presente memoria es un solo ejemplo; se puede eliminar una parte del procesamiento descrito, o se puede agregar un procesamiento no descrito.

Más aún, el flujo de procesamiento descrito en la presente memoria es básicamente común para el codificador 100 y el decodificador 200; la única diferencia es si la señal necesaria para el procesamiento se codifica en un flujo o se analiza a partir de un flujo.

[Segundo ejemplo de procesamiento de predicción inter]

La FIG. 15 es un diagrama que ilustra un segundo ejemplo del esquema de la configuración de canalización utilizada para el decodificador 200. El segundo ejemplo ilustrado en la FIG. 15 es diferente del primer ejemplo descrito con referencia a la FIG. 11 en que el decodificador 200 utiliza, como el vector de movimiento de referencia circundante de un bloque decodificado vecino que se utilizará para la derivación de MVP en el procesamiento de predicción inter, un vector de movimiento temporal (un primer vector de movimiento) generado utilizando uno o más MVP obtenidos en el procesamiento de derivación de MVP, en lugar del vector de movimiento final (un segundo vector de movimiento) obtenido al realizar todo el procesamiento relacionado con la derivación del vector de movimiento.

De esa manera, al utilizar el vector de movimiento temporal como el vector de movimiento de referencia circundante para derivar uno o más MVP en el procesamiento de decodificación en un bloque posterior, es posible reducir significativamente la longitud del bucle de retroalimentación. Como resultado, la segunda etapa, que es una etapa larga en el primer ejemplo, se puede dividir en dos etapas más cortas, a saber, la segunda etapa y la tercera etapa, mientras se satisface la condición de que el bucle de retroalimentación no se extienda sobre varias etapas.

Tenga en cuenta que este esquema de la configuración de canalización es un solo ejemplo; se puede eliminar una parte del procesamiento descrito, se puede agregar un procesamiento no descrito o se pueden modificar los límites de las etapas.

La FIG. 16 ilustra, en series de tiempo, la temporización del procesamiento de las respectivas etapas para cada bloque que se va a decodificar, en el segundo ejemplo del esquema de la configuración de canalización descrito con referencia a la FIG. 15.

Como en la FIG. 13, la FIG. 16 ilustra la temporización del procesamiento para cinco bloques que se van a decodificar, a saber, las unidades de codificación CU0 a CU4 ilustradas en el ejemplo de partición de bloques de la FIG. 12. En el primer ejemplo, la segunda etapa es una etapa larga, mientras que en el segundo ejemplo, la segunda etapa se divide en dos etapas más cortas, a saber, la segunda etapa y la tercera etapa. La longitud de cada una de las segundas y terceras etapas es la misma que la longitud de cada una de las otras etapas.

El procesamiento de cada etapa comienza después de que finalice el procesamiento de la misma etapa para el bloque inmediatamente anterior al bloque actual en el orden de procesamiento. Por ejemplo, el procesamiento de la segunda etapa para la unidad de codificación CU1 comienza en el tiempo t_4 en el que finaliza la segunda etapa para la unidad de codificación CU0. En este momento, dado que la duración del procesamiento de la segunda etapa para la unidad de codificación CU0 es la misma que la duración del procesamiento de cada una de las otras etapas, el procesamiento de la segunda etapa para la unidad de codificación CU1 puede comenzar sin tiempo de espera después de que finalice el procesamiento de la primera etapa para la unidad de codificación CU1.

Por otro lado, dado que la unidad de codificación CU2 es más pequeña en tamaño de bloque que la unidad de codificación CU1 que está inmediatamente anterior a la unidad de codificación CU2 en el orden de procesamiento, existe un tiempo de espera entre la primera etapa y la segunda etapa, pero este tiempo de espera no se acumula y se vuelve cero cuando comienza el procesamiento para la unidad de codificación CU4.

Como resultado, incluso con el tiempo de espera, la duración del procesamiento requerida para completar el procesamiento de decodificación para un cuadro es aproximadamente la misma que la duración del procesamiento originalmente prevista, y es más probable que se pueda completar el procesamiento en todos los bloques dentro de la duración del procesamiento asignada a un cuadro.

La FIG. 17 es un diagrama de flujo del procesamiento de predicción inter en el segundo ejemplo del esquema de la configuración de canalización descrita con referencia a la FIG. 15. El procesamiento ilustrado en la FIG. 17 se realiza repetidamente para cada bloque de predicción, que es la unidad de procesamiento del procesamiento de predicción de fotogramas inter. El procesamiento ilustrado en la FIG. 17 lo realizan el codificador 100 y el decodificador 200. Aunque lo siguiente describe principalmente las operaciones del predictor inter 126 incluido en el codificador 100, predictor inter 218 incluido en el decodificador 200.

El procesamiento ilustrado en la FIG. 17 es diferente del procesamiento de acuerdo con el primer ejemplo descrito con referencia a la FIG. 14 en el siguiente aspecto: En el procesamiento ilustrado en la FIG. 17, lo que se almacena en la memoria de vector de movimiento de referencia circundante no es el vector de movimiento final derivado utilizando cada procedimiento. En cambio, un vector de movimiento temporal derivado utilizando uno o más MVP obtenidos utilizando cada procedimiento se almacena en la memoria de vector de movimiento de referencia circundante.

Con esto, el vector de movimiento de referencia circundante utilizado para derivar uno o más MVP en el procesamiento en un bloque posterior se puede realimentar en un punto temprano en el tiempo en el flujo de

procesamiento. Por lo tanto, como se describe con referencia a la FIG. 16, es más probable que pueda reducir significativamente el tiempo de espera para esperar la siguiente etapa en el control de canalización.

El vector de movimiento temporal almacenado en la memoria de vector de movimiento de referencia circundante se deriva de la siguiente manera:

5 (1) En el modo inter normal, el predictor inter 126 determina, como el vector de movimiento temporal, el vector de movimiento final obtenido a través del procesamiento de derivación de vector de movimiento normal.

10 (2) En el modo de fusión, el predictor inter 126 determina, como el vector de movimiento temporal, el MVP (el MVP más apropiado) especificado con un índice de fusión entre la pluralidad de MVP indicados en la lista de MVP de fusión.

15 (3) En el modo de FRUC, el predictor inter 126 deriva el vector de movimiento temporal utilizando la pluralidad de MVP indicados en la lista de MVP de FRUC mediante uno de los siguientes procedimientos, por ejemplo: El predictor inter 126 determina, como el vector de movimiento temporal, un MVP registrado en la parte superior de la lista de MVP de FRUC. Alternativamente, el predictor inter 126 reduce cada uno de la pluralidad de MVP indicados en la lista de MVP de FRUC al intervalo de tiempo del cuadro de referencia temporalmente más cercano. El predictor inter 126 calcula, para cada una de las direcciones L0 y L1, el promedio o la mediana de la pluralidad de MVP obtenidos mediante el escalado, y determina el vector de movimiento calculado como el vector de movimiento temporal.

20 Tenga en cuenta que el predictor inter 126 puede excluir, de la pluralidad de MVP indicados en la lista de MVP de FRUC, uno o más MVP registrados con referencia al vector de movimiento temporal, y puede derivar un vector de movimiento temporal al llevar a cabo uno de los procedimientos descritos anteriormente en los MVP restantes.

25 Aunque, en la FIG. 17, los vectores de movimiento temporales almacenados en la memoria de vectores de movimiento de referencia circundante se utilizan cada uno como el vector de movimiento de referencia circundante para derivar uno o más MVP, los vectores de movimiento temporales se pueden utilizar cada uno como el vector de movimiento de referencia circundante en otro procesamiento, tal como el filtrado de bucle. Tenga en cuenta que en otro procesamiento, tal como el filtrado de bucle, se puede utilizar el vector de movimiento final utilizado para la compensación de movimiento, en lugar del vector de movimiento temporal. Específicamente, el vector de movimiento final se deriva en la tercera etapa ilustrada en la FIG. 15. Por lo tanto, el vector de movimiento final se puede utilizar en el procesamiento de la cuarta etapa y etapas posteriores.

30 En la configuración ilustrada en la FIG. 15, es decir, la configuración de canalización que supone este flujo de procesamiento, el vector de movimiento temporal se retroalimenta como el vector de movimiento de referencia circundante en la temporización inmediatamente posterior al procesamiento de derivación de MVP; sin embargo, el vector de movimiento temporal se puede retroalimentar en una temporización diferente, siempre que se pueda obtener la misma información que el vector de movimiento temporal descrito en la presente memoria.

35 Tenga en cuenta que este flujo de procesamiento es un solo ejemplo; se puede eliminar una parte del procesamiento descrito, o se puede agregar un procesamiento no descrito. Por ejemplo, en el modo de fusión, el vector de movimiento temporal puede ser el mismo que el vector de movimiento final en el caso en el que no se realiza la búsqueda en la vecindad del MVP más apropiado.

Más aún, el flujo de procesamiento descrito en la presente memoria es básicamente común para el codificador 100 y el decodificador 200; la única diferencia es si la señal necesaria para el procesamiento se codifica en un flujo o se analiza a partir de un flujo.

[Efectos ventajosos del segundo ejemplo de procesamiento de predicción inter]

45 Con la configuración descrita con referencia a la FIG. 15 a la FIG. 17, el vector de movimiento de referencia circundante utilizado para derivar uno o más MVP en el procesamiento en un bloque posterior se puede realimentar en un punto temprano en el tiempo en el flujo de procesamiento. De acuerdo con lo anterior, es posible reducir significativamente el tiempo de espera, que se produce en el primer ejemplo, para esperar la siguiente etapa en el control de canalización. Con esto, incluso un decodificador con un bajo rendimiento de procesamiento tiene más probabilidades de poder completar el procesamiento en todos los bloques dentro de la duración de procesamiento asignada a un cuadro.

[Tercer ejemplo de procesamiento de predicción inter]

55 La FIG. 18 es un diagrama que ilustra un tercer ejemplo del esquema de la configuración de canalización utilizada para el decodificador 200. El tercer ejemplo ilustrado en la FIG. 18 es diferente del primer ejemplo ilustrado en la FIG. 11 en que el decodificador 200 utiliza, como el vector de movimiento de referencia

circundante de un bloque decodificado vecino que se utilizará para la derivación de MVP en el procesamiento de predicción inter: el vector de movimiento temporal disponible antes de que se realice la búsqueda en la vecindad del MVP más apropiado cuando el modo de predicción inter es el modo de FRUC; y el vector de movimiento temporal disponible antes de que se realice el procesamiento de DMVR cuando el modo de predicción inter es el modo de fusión, en lugar del vector de movimiento final obtenido al realizar todo el procesamiento relacionado con la derivación del vector de movimiento.

De esa manera, la longitud del bucle de retroalimentación se vuelve relativamente corta al utilizar el vector de movimiento temporal como el vector de movimiento de referencia circundante para derivar uno o más MVP en el procesamiento de decodificación en un bloque posterior. Esto hace posible realimentar el vector de movimiento de referencia circundante en un punto temprano en el tiempo en la tercera etapa. De acuerdo con lo anterior, si el procesamiento de derivación de MVP para el bloque posterior puede comenzar después de que se determine el vector de movimiento temporal, la segunda etapa, que es una etapa larga en el primer ejemplo, se puede dividir en dos etapas más cortas, a saber, la segunda etapa y la tercera etapa.

Tenga en cuenta que este esquema de la configuración de canalización es un solo ejemplo; una parte del procesamiento descrito se puede eliminar, se puede agregar un procesamiento no descrito o se pueden modificar los límites de las etapas.

La FIG. 19 ilustra, en series de tiempo, la temporización de procesamiento de las respectivas etapas para cada bloque que se va a decodificar, en el tercer ejemplo del esquema de la configuración de canalización descrita con referencia a la figura 18.

Como en la FIG. 13, la FIG. 19 ilustra la temporización de procesamiento para cinco bloques que se van a decodificar, a saber, las unidades de codificación CU0 a CU4 ilustradas en el ejemplo de partición de bloques en la FIG. 12. En el primer ejemplo, la segunda etapa es una etapa larga, mientras que en el tercer ejemplo, la segunda etapa se divide en dos etapas más cortas, a saber, la segunda etapa y la tercera etapa. En la presente memoria, la segunda etapa es más corta que las otras etapas. Esto se debe a que el procesamiento en la segunda etapa solo incluye la derivación de MVP y la transferencia de memoria de la imagen de referencia, y por lo tanto implica una pequeña cantidad de procesamiento. Tenga en cuenta que en la presente memoria se supone que la velocidad de la transferencia de memoria de la imagen de referencia es suficientemente rápida.

El procesamiento de cada etapa comienza después de que finalice el procesamiento de la misma etapa para el bloque inmediatamente anterior al bloque actual en el orden de procesamiento. Sin embargo, el tercer ejemplo es un caso excepcional en el que la segunda etapa comienza después de que se determine el vector de movimiento temporal en la tercera etapa para el bloque inmediatamente anterior al bloque actual en el orden de procesamiento. Por lo tanto, por ejemplo, el procesamiento de la segunda etapa para la unidad de codificación CU1 comienza en el tiempo t4 en el que termina la primera mitad del procesamiento en la tercera etapa para la unidad de codificación CU0. En este momento, dado que la duración del procesamiento de la segunda etapa para la unidad de codificación CU0 es suficientemente corta en comparación con las duraciones del procesamiento de las otras etapas, el procesamiento de la segunda etapa para la unidad de codificación CU1 puede comenzar sin tiempo de espera después de que termina el procesamiento de la primera etapa para la unidad de codificación CU1.

Por otro lado, por ejemplo, aunque el procesamiento de la segunda etapa para la unidad de codificación CU3 comienza después de que termina la primera mitad del procesamiento de la tercera etapa para la unidad de codificación CU2, la duración del procesamiento de la segunda etapa para la unidad de codificación CU2 no es suficientemente corta en comparación con las duraciones del procesamiento de las otras etapas y, por lo tanto, hay un ligero aumento en el tiempo de espera.

Como resultado, en comparación con el segundo ejemplo descrito con referencia a la FIG. 16, hay un tiempo de espera desde el tiempo t8 hasta el tiempo t9 entre la primera etapa y la segunda etapa cuando se procesa la unidad de codificación CU4. Sin embargo, en comparación con el primer ejemplo descrito con referencia a la FIG. 13, el tiempo de espera se reduce significativamente, e incluso con el tiempo de espera, la duración de procesamiento requerida para completar el procesamiento de decodificación para un cuadro es aproximadamente la misma que la duración de procesamiento originalmente prevista. Por lo tanto, es más probable poder completar el procesamiento en todos los bloques dentro de la duración de procesamiento asignada a un cuadro.

La FIG. 20 es un diagrama de flujo del procesamiento de predicción inter en el tercer ejemplo del esquema de la configuración de canalización descrita con referencia a la FIG. 18. El procesamiento ilustrado en la FIG. 20 se realiza repetidamente para cada bloque de predicción, que es la unidad de procesamiento del procesamiento de predicción de fotograma inter. El procesamiento ilustrado en la FIG. 20 se realiza mediante el codificador 100 y el decodificador 200. Aunque lo siguiente describe principalmente las operaciones del predictor inter 126 incluido en el codificador 100, lo mismo se aplica a las operaciones del predictor inter 218 incluido en el decodificador 200.

El procesamiento ilustrado en la FIG. 20 es diferente del procesamiento de acuerdo con el primer ejemplo descrito con referencia a la FIG. 14 en el siguiente aspecto: En el procesamiento ilustrado en la FIG. 20, el vector de movimiento final derivado utilizando cada procedimiento no se utiliza como el vector de movimiento de referencia circundante. En cambio, el vector de movimiento temporal que es un valor disponible en medio del procesamiento de derivación del vector de movimiento realizado utilizando cada procedimiento se utiliza como el vector de movimiento de referencia circundante y se almacena en la memoria del vector de movimiento de referencia circundante.

Con esto, el vector de movimiento de referencia circundante utilizado para derivar uno o más MVP en el procesamiento en un bloque posterior se puede retroalimentar en un punto temprano en el tiempo en el flujo de procesamiento. Por lo tanto, como se describe con referencia a la FIG. 19, es más probable poder reducir significativamente el tiempo de espera para esperar la siguiente etapa en el control de canalización.

El vector de movimiento temporal almacenado en la memoria de vector de movimiento de referencia circundante se deriva de la siguiente manera:

(1) En el modo inter normal, el predictor inter 126 determina, como el vector de movimiento temporal, el vector de movimiento final obtenido a través del procesamiento de derivación de vector de movimiento normal.

(2) En el modo de fusión, el predictor inter 126 determina, como el vector de movimiento temporal, el MVP (el MVP más apropiado) especificado con un índice de fusión entre la pluralidad de MVP indicados en la lista de MVP de fusión.

(3) En el modo de FRUC, el predictor inter 126 determina, como el vector de movimiento temporal, el MVP (el MVP más apropiado) que se determina para hacer que el valor de coste sea el más bajo entre la pluralidad de MVP indicados en la lista de MVP de FRUC, utilizando el procedimiento de coincidencia bilateral o el procedimiento de coincidencia de plantilla.

Aunque, en la FIG. 20, los vectores de movimiento temporal almacenados en la memoria de vector de movimiento de referencia circundante se utilizan cada uno como el vector de movimiento de referencia circundante para derivar uno o más MVP, los vectores de movimiento temporal se pueden utilizar cada uno como el vector de movimiento de referencia circundante en otro procesamiento, tal como el filtrado de bucle. Tenga en cuenta que en otro procesamiento, tal como el filtrado de bucle, se puede utilizar el vector de movimiento final utilizado para la compensación de movimiento, en lugar del vector de movimiento temporal. Específicamente, el vector de movimiento final se deriva en la tercera etapa ilustrada en la FIG. 18. Por lo tanto, el vector de movimiento final se puede utilizar en el procesamiento de la cuarta etapa y etapas posteriores.

En la configuración ilustrada en la FIG. 18, es decir, la configuración de canalización que supone este flujo de procesamiento, el vector de movimiento temporal se retroalimenta como el vector de movimiento de referencia circundante en la temporización inmediatamente anterior a la búsqueda en la vecindad del MVP más apropiado y el procesamiento de DMVR; sin embargo, el vector de movimiento temporal se puede retroalimentar en una temporización diferente, siempre que se pueda obtener la misma información que el vector de movimiento temporal descrito en la presente memoria.

Tenga en cuenta que este flujo de procesamiento es un solo ejemplo; se puede eliminar una parte del procesamiento descrito, o se puede agregar un procesamiento no descrito. Por ejemplo, en el modo de fusión o el modo de FRUC, el vector de movimiento temporal puede ser el mismo que el vector de movimiento final en el caso en el que no se realiza la búsqueda en la vecindad del MVP más apropiado.

Más aún, el flujo de procesamiento descrito en la presente memoria es básicamente común al codificador 100 y al decodificador 200; la única diferencia es si la señal necesaria para el procesamiento se codifica en un flujo o se analiza a partir de un flujo.

[Efectos ventajosos del tercer ejemplo de procesamiento de predicción inter]

Con la configuración descrita con referencia a la FIG. 18 a la FIG. 20, el vector de movimiento de referencia circundante utilizado para derivar uno o más MVP en el procesamiento en un bloque posterior se puede realimentar en un punto temprano en el tiempo en el flujo de procesamiento. De acuerdo con lo anterior, es posible reducir significativamente el tiempo de espera, que ocurre en el primer ejemplo, para esperar la siguiente etapa en el control de canalización. Con esto, incluso un decodificador con bajo rendimiento de procesamiento tiene más probabilidades de poder completar el procesamiento en todos los bloques dentro de la duración de procesamiento asignada a un cuadro.

En comparación con el segundo ejemplo descrito con referencia a la FIG. 17, el vector de movimiento obtenido a través del procesamiento hasta el procesamiento de determinación de MVP más apropiado se puede utilizar como el vector de movimiento de referencia circundante cuando el modo de predicción inter es el modo de

FRUC. Por lo tanto, dado que es posible hacer referencia a un vector de movimiento más confiable, es más probable que se pueda mejorar la eficiencia de codificación.

[Vector de movimiento de referencia circundante que combina el vector de movimiento final y el vector de movimiento temporal]

5 Además de almacenar el vector de movimiento temporal en la memoria de vector de movimiento de referencia circundante de acuerdo con el segundo ejemplo descrito con referencia a la FIG. 17 y el tercer ejemplo descrito con referencia a la FIG. 20, el vector de movimiento final también se puede almacenar como en el primer ejemplo descrito con referencia a la FIG. 14.

10 Al hacerlo, en el procesamiento de derivación de MVP para un bloque posterior, el predictor inter 126 puede obtener, como el vector de movimiento de referencia circundante, el vector de movimiento final de, entre una pluralidad de bloques de referencia circundantes, un bloque del cual se puede obtener el vector de movimiento final, mientras que el predictor inter 126 puede obtener, como el vector de movimiento de referencia circundante, el vector de movimiento temporal de, entre la pluralidad de bloques de referencia circundantes, un bloque del cual no se puede obtener el vector de movimiento final.

15 La FIG. 21 y la FIG. 22 son diagramas que ilustran bloques vecinos a los que se hace referencia para derivar uno o más MVP para el bloque actual. La unidad de codificación CU4 es el bloque actual, las unidades de codificación CU0 a CU3 son bloques vecinos espacialmente en los cuales ya se ha completado el procesamiento, y la unidad de codificación CU col es un bloque vecino temporalmente que está colocado en otro cuadro en el cual ya se ha completado el procesamiento.

20 La FIG. 21 es un diagrama que ilustra el caso en el que el vector de movimiento final no se puede obtener de un bloque que está inmediatamente antes del bloque actual en el orden de procesamiento. En este ejemplo, el predictor inter 126 obtiene un vector de movimiento temporal como el vector de movimiento de referencia circundante de la unidad de codificación CU3 (primer bloque procesado), y obtiene vectores de movimiento final como los vectores de movimiento de referencia circundante de los otros bloques (tercer bloque procesado).

25 Tenga en cuenta que la unidad de codificación CU col es un bloque incluido en un cuadro en el que ya se ha completado el procesamiento. Por lo tanto, el predictor inter 126 obtiene un vector de movimiento final como el vector de movimiento de referencia circundante de la unidad de codificación CU col.

30 La FIG. 22 es un diagrama que ilustra el caso en el que los vectores de movimiento final no se pueden obtener de dos bloques que están inmediatamente antes del bloque actual en el orden de procesamiento. En este ejemplo, el predictor inter 126 obtiene vectores de movimiento temporales como vectores de movimiento de referencia circundantes de las unidades de codificación CU2 y CU3 (primer bloque procesado, segundo bloque procesado), y el predictor inter 126 obtiene vectores de movimiento finales como vectores de movimiento de referencia circundantes de los otros bloques (tercer bloque procesado). Tenga en cuenta que la unidad de codificación CU col es un bloque incluido en un cuadro en el que ya se ha completado el procesamiento. Por lo tanto, el predictor inter 126 obtiene un vector de movimiento final como vector de movimiento de referencia circundante de la unidad de codificación CU col.

35 Como se describió anteriormente, al obtener vectores de movimiento finales como vectores de movimiento de referencia circundantes de, entre los bloques de referencia circundantes, los bloques de los que se pueden obtener vectores de movimiento finales, el predictor inter 126 puede realizar la derivación de MVP con referencia a vectores de movimiento que son más confiables en comparación con el caso de obtener vectores de movimiento temporales de todos los bloques como vectores de movimiento de referencia circundantes. Esto hace que sea más probable poder mejorar la eficiencia de codificación.

40 Tenga en cuenta que el predictor inter 126 puede cambiar los vectores de movimiento a los que se puede hacer referencia dependiendo de si el límite del bloque actual es el límite de la CTU. Por ejemplo, cuando el bloque actual no es adyacente al límite superior de la CTU, el predictor inter 126 determina si se debe hacer referencia a un vector de movimiento final o a un vector de movimiento temporal de un bloque adyacente al extremo superior del bloque actual, utilizando dichos procedimientos como los descritos con referencia a la FIG. 21 y la FIG. 22. Por otra parte, cuando el bloque actual es adyacente al límite superior de la CTU, el predictor inter 126 hace referencia constantemente a un vector de movimiento final porque la derivación del vector de movimiento final se ha completado para el bloque adyacente al extremo superior del bloque actual. De la misma manera, cuando el bloque actual es adyacente al límite izquierdo de la CTU, el predictor inter 126 hace referencia constantemente a un vector de movimiento final de un bloque adyacente al extremo izquierdo del bloque actual.

[Combinación de los primero, segundo y tercero ejemplos]

45 Se puede utilizar un flujo de procesamiento que combina el primer ejemplo descrito con referencia a la FIG. 14, el segundo ejemplo descrito con referencia a la FIG. 17 y el tercer ejemplo descrito con referencia a la FIG. 20.

50 Un ejemplo específico es que cuando el modo de predicción inter es el modo de fusión, el predictor inter 126 puede utilizar el vector de movimiento final como el vector de movimiento de referencia circundante como en

el primer ejemplo, mientras que cuando el modo de predicción inter es el modo de FRUC, el predictor inter 126 puede utilizar, como el vector de movimiento de referencia circundante, un vector de movimiento temporal disponible antes de que se realice la búsqueda en la vecindad del MVP más apropiado como en el tercer ejemplo. Dado que el procesamiento en modo de fusión implica una menor cantidad de procesamiento en comparación con el procesamiento en modo de FRUC, el procesamiento para un bloque posterior se puede completar dentro de la duración del procesamiento en la que se requiere que se complete el procesamiento, incluso si un vector de movimiento final se retroalimenta como el vector de movimiento de referencia circundante después de ser finalizado. Como tal, existe la posibilidad de poder realizar el procesamiento sin la acumulación del tiempo de espera para esperar la siguiente etapa en el control de canalización. Más aún, al hacer que un vector de movimiento más confiable esté disponible como el vector de movimiento de referencia circundante en el caso del modo de fusión, es más probable que la eficiencia de codificación mejore.

[Intercambio de acuerdo con la señal del modo de bajo retardo]

El predictor inter 126 puede determinar si se debe o no procesar un flujo actual en un modo de bajo retardo, y cuando determina procesar el flujo actual en el modo de bajo retardo, el predictor inter 126 puede hacer referencia al vector de movimiento temporal como el vector de movimiento de referencia circundante para el procesamiento de derivación de MVP como se describe en el segundo ejemplo o el tercer ejemplo. Por otro lado, cuando determina no procesar el flujo actual en el modo de bajo retardo, el predictor inter 126 puede hacer referencia al vector de movimiento final como el vector de movimiento de referencia circundante para el procesamiento de derivación de MVP como se describe en el primer ejemplo.

Al hacerlo, la longitud del bucle de retroalimentación para el vector de movimiento de referencia circundante se acorta en el modo de retardo bajo y, por lo tanto, es más probable que el tiempo de espera para esperar la siguiente etapa en el control de canalización se reduzca significativamente. Por otro lado, cuando el modo no es el modo de retardo bajo, hay un tiempo de espera para esperar la siguiente etapa en el control de canalización, pero es más probable que la eficiencia de codificación mejore porque un vector de movimiento altamente confiable se puede denominar el vector de movimiento de referencia circundante.

El codificador 100 genera información que indica si se debe o no realizar el procesamiento en el modo de retardo bajo y codifica la información generada en un flujo. El decodificador 200 obtiene la información al analizar la información del flujo y determina si se debe o no realizar el procesamiento en el modo de retardo bajo en base a la información obtenida. Tenga en cuenta que la información se describe en una región de encabezado de secuencia, una región de encabezado de cuadro, una región de encabezado de segmento o una región de información auxiliar del flujo actual.

Por ejemplo, el codificador 100 puede intercambiar si se realiza o no el procesamiento en el modo de retardo bajo, de acuerdo con el tamaño de un cuadro actual que se va a codificar. Por ejemplo, cuando el tamaño del cuadro es pequeño, el número de bloques que se van a procesar es pequeño, lo que significa que hay tiempo suficiente para procesar los bloques y, por lo tanto, el codificador 100 no se establece en el modo de retardo bajo, mientras que cuando el tamaño del cuadro es grande, el número de bloques que se van a procesar es grande, lo que significa que no hay tiempo suficiente para procesar los bloques y, por lo tanto, el codificador 100 se establece en el modo de retardo bajo.

El codificador 100 puede intercambiar si se realiza o no el procesamiento en el modo de retardo bajo, de acuerdo con la capacidad de procesamiento del decodificador 200 al que se va a transmitir el flujo. Por ejemplo, cuando la capacidad de procesamiento del decodificador 200 es alta, el codificador 100 no se configura en el modo de retardo bajo porque el decodificador 200 es capaz de realizar una gran cantidad de procesamiento dentro de una duración de procesamiento determinada. Por otra parte, cuando la capacidad de procesamiento del decodificador 200 es baja, el codificador 100 se configura en el modo de retardo bajo porque el decodificador 200 no es capaz de realizar una gran cantidad de procesamiento dentro de una determinada duración de procesamiento.

El codificador 100 puede intercambiar si realiza o no el procesamiento en el modo de retardo bajo, de acuerdo con un perfil o información de nivel asignada al flujo actual. Por ejemplo, el codificador 100 no se establece en el modo de retardo bajo cuando el perfil o el nivel asignado al flujo suponen un decodificador que tiene una capacidad de procesamiento suficiente. Por otra parte, el codificador 100 se configura en el modo de retardo bajo cuando el perfil o el nivel asignado al flujo suponen un decodificador que tiene una capacidad de procesamiento insuficiente.

Tenga en cuenta que la información que indica si se debe o no realizar el procesamiento en el modo de bajo retardo y que está codificada en el flujo no necesita ser una señal que indique directamente si se debe o no realizar el procesamiento en el modo de bajo retardo, y puede ser una señal que tenga un significado diferente. Por ejemplo, la información que indica si se debe o no realizar el procesamiento en el modo de bajo retardo puede estar directamente asociada con el perfil y el nivel, de modo que se pueda determinar si se debe o no realizar el procesamiento en el modo de bajo retardo se basa únicamente en la señal que indica el perfil y el nivel.

- Como se describió anteriormente, el codificador 100 de acuerdo con la presente realización: cuando codifica un bloque actual en un modo de predicción inter en el que el decodificador 200 realiza una estimación de movimiento (por ejemplo, modo de fusión o modo de FRUC en S201 en la FIG. 17), deriva un primer vector de movimiento del bloque actual (S203 o S205); almacena, en la memoria, el primer vector de movimiento derivado; deriva un segundo vector de movimiento del bloque actual (S204 o S207); y genera una imagen de predicción del bloque actual al realizar una compensación de movimiento utilizando el segundo vector de movimiento (S208). Al derivar el primer vector de movimiento (S203 o S205), el codificador 100 deriva el primer vector de movimiento del bloque actual utilizando un primer vector de movimiento de un bloque procesado.
- De acuerdo con lo anterior, el decodificador 200 puede comenzar a derivar el primer vector de movimiento del bloque actual, por ejemplo, después de que se derive el primer vector de movimiento de un bloque vecino en el control de canalización, sin esperar a que se complete la derivación del segundo vector de movimiento del bloque vecino. Por lo tanto, dado que el tiempo de espera que se produce en el control de canalización realizado por el decodificador 200 se puede reducir en comparación con el caso de derivar el primer vector de movimiento utilizando el segundo vector de movimiento del bloque vecino, es posible reducir el retardo de procesamiento.
- Por ejemplo, al derivar el primer vector de movimiento, el codificador 100 (i) genera una lista de predictores de vector de movimiento que indica una pluralidad de predictores de vector de movimiento utilizando el primer vector de movimiento del bloque procesado, y (ii) determina el primer vector de movimiento del bloque actual a partir de la pluralidad de predictores de vector de movimiento indicados en la lista de predictores de vector de movimiento (por ejemplo, S203 o S205 en la FIG. 17).
- Por ejemplo, el modo de predicción inter en el que el decodificador 200 realiza la estimación de movimiento es un modo de fusión, y al derivar el segundo vector de movimiento, el codificador 100 deriva el segundo vector de movimiento al realizar la estimación de movimiento en la vecindad del primer vector de movimiento (por ejemplo, S204 en la FIG. 17).
- Por ejemplo, el modo de predicción inter en el que el decodificador 200 realiza la estimación de movimiento es un modo de conversión ascendente de la velocidad del fotograma (FRUC), y al derivar el segundo vector de movimiento, el codificador 100 deriva el segundo vector de movimiento al realizar la estimación de movimiento en la vecindad del primer vector de movimiento (por ejemplo, S207 en la FIG. 20).
- Por ejemplo, el modo de predicción inter en el que el decodificador 200 realiza la estimación de movimiento es un modo de conversión ascendente de la velocidad del fotograma (FRUC), y al derivar el segundo vector de movimiento, el codificador 100 (i) determina un tercer vector de movimiento (el MVP más apropiado) a partir de la pluralidad de predictores de vector de movimiento indicados en la lista de predictores de vector de movimiento (por ejemplo, S206 en la FIG. 17), y (ii) deriva el segundo vector de movimiento al realizar la estimación de movimiento en la vecindad del tercer vector de movimiento (por ejemplo, S207 en la FIG. 17).
- Por ejemplo, al determinar el primer vector de movimiento (por ejemplo, S205 en la FIG. 17), el codificador 100 deriva el primer vector de movimiento en base, para cada una de las direcciones de predicción, a un promedio o una mediana de la pluralidad de predictores de vector de movimiento indicados en la lista de predictores de vector de movimiento.
- Por ejemplo, al determinar el primer vector de movimiento (por ejemplo, S205 en la FIG. 17), entre la pluralidad de predictores de vector de movimiento indicados en la lista de predictores de vector de movimiento, el codificador 100 determina, como el primer vector de movimiento, un predictor de vector de movimiento indicado en la parte superior de la lista de predictores de vector de movimiento.
- Por ejemplo, al generar la lista de predictores de vector de movimiento (por ejemplo, S203 o S205 en la FIG. 17), el codificador 100 deriva cada uno de la pluralidad de predictores de vector de movimiento utilizando el primer vector de movimiento del bloque procesado o un segundo vector de movimiento del bloque procesado.
- Al determinar el primer vector de movimiento (por ejemplo, S205 en la FIG. 17), el codificador 100 determina el primer vector de movimiento a partir de, entre la pluralidad de predictores de vector de movimiento indicados en la lista de predictores de vector de movimiento, uno o más candidatos de predictor de vector de movimiento derivados utilizando el segundo vector de movimiento del bloque procesado. De acuerdo con lo anterior, dado que el primer vector de movimiento se puede determinar utilizando el segundo vector de movimiento confiable, es posible suprimir una disminución en la confiabilidad del primer vector de movimiento.
- Por ejemplo, al generar la lista de predictores de vector de movimiento (por ejemplo, S203 o S205 en la FIG. 17), el codificador 100 deriva un predictor de vector de movimiento utilizando el primer vector de movimiento del bloque procesado cuando el bloque procesado se incluye en un cuadro que incluye el bloque actual, y el codificador 100 deriva un predictor de vector de movimiento utilizando un segundo vector de movimiento del bloque procesado cuando el bloque procesado se incluye en un cuadro diferente del cuadro que incluye el bloque actual. De acuerdo con lo anterior, cuando el bloque procesado se incluye en un cuadro diferente del cuadro que incluye el bloque actual, el codificador 100 utiliza el segundo vector de movimiento y, como resultado, se puede mejorar la fiabilidad del predictor de vector de movimiento.

Por ejemplo, al generar la lista de predictores de vector de movimiento (por ejemplo, S203 o S205 en la FIG. 17), el codificador 100 determina si se debe utilizar el primer vector de movimiento del bloque procesado o un segundo vector de movimiento del bloque procesado para realizar la derivación del predictor de vector de movimiento, de acuerdo con la posición del bloque procesado con respecto al bloque actual.

- 5 Por ejemplo, al generar la lista de predictores de vector de movimiento (por ejemplo, S203 o S205 en la FIG. 17), el codificador 100 deriva un predictor de vector de movimiento utilizando el segundo vector de movimiento del bloque procesado cuando el bloque procesado se incluye en una unidad de procesamiento diferente de la unidad de procesamiento (por ejemplo, CTU) que incluye el bloque actual.

- 10 Por ejemplo, al generar la lista de predictores de vector de movimiento (por ejemplo, S203 o S205 en la FIG. 17), el codificador 100: deriva el predictor de vector de movimiento a partir de un primer bloque procesado entre una pluralidad de bloques procesados incluidos en un cuadro que incluye el bloque actual utilizando un primer vector de movimiento del primer bloque procesado, el primer bloque procesado es de N bloques antes del bloque actual en un orden de procesamiento; deriva el predictor de vector de movimiento a partir de un segundo bloque procesado entre la pluralidad de bloques procesados utilizando un primer vector de movimiento del segundo bloque procesado, el segundo bloque procesado está después del primer bloque procesado en el orden de procesamiento; y deriva el predictor de vector de movimiento a partir de un tercer bloque procesado entre la pluralidad de bloques procesados utilizando un segundo vector de movimiento del tercer bloque procesado, el tercer bloque procesado está antes del primer bloque procesado en el orden de procesamiento.

- 20 De acuerdo con lo anterior, el codificador 100 utiliza el segundo vector de movimiento del tercer bloque procesado que está antes del primer bloque procesado en el orden de procesamiento y, como resultado, se puede mejorar la fiabilidad del predictor del vector de movimiento.

Por ejemplo, N es 1.

Por ejemplo, el primer vector de movimiento se utiliza además en otro procesamiento diferente de la derivación del predictor del vector de movimiento. El otro procesamiento es el filtrado de bucle, por ejemplo.

- 25 Por ejemplo, el segundo vector de movimiento se utiliza en el filtrado de bucle.

Por ejemplo, al codificar el bloque actual en un modo de retardo bajo, el codificador 100 deriva el primer vector de movimiento del bloque actual utilizando el primer vector de movimiento del bloque procesado.

De acuerdo con lo anterior, el codificador 100 puede realizar un procesamiento adecuado de acuerdo con si o no se utiliza el modo de retardo bajo.

- 30 Por ejemplo, el codificador 100 codifica, en una región de encabezado de secuencia, una región de encabezado de cuadro, una región de encabezado de segmento o una región de información auxiliar, información que indica si se debe codificar el bloque actual en el modo de retardo bajo.

Por ejemplo, el codificador 100 se intercambia si se debe codificar el bloque actual en el modo de retardo bajo, de acuerdo con el tamaño de un cuadro actual que incluye el bloque actual.

- 35 Por ejemplo, el codificador 100 se intercambia si se debe codificar el bloque actual en el modo de retardo bajo, de acuerdo con la capacidad de procesamiento del decodificador.

Por ejemplo, el codificador 100 se intercambia si se debe codificar el bloque actual en el modo de retardo bajo, de acuerdo con un perfil o información de nivel asignada a un flujo actual que se va a codificar.

- 40 El decodificador 200 de acuerdo con la presente realización, al decodificar un bloque actual en un modo de predicción inter en el que el decodificador 200 realiza una estimación de movimiento (por ejemplo, modo de fusión o modo de FRUC en S201 en la FIG. 17): deriva un primer vector de movimiento del bloque actual (S203 o S205); almacena, en la memoria, el primer vector de movimiento derivado; deriva un segundo vector de movimiento del bloque actual (S204 o S207); y genera una imagen de predicción del bloque actual al realizar una compensación de movimiento utilizando el segundo vector de movimiento (S208). Al derivar el primer vector de movimiento (S203 o S205), el decodificador 200 deriva el primer vector de movimiento del bloque actual utilizando un primer vector de movimiento de un bloque procesado.

- 45 De acuerdo con lo anterior, el decodificador 200 puede comenzar a derivar el primer vector de movimiento del bloque actual, por ejemplo, después de que el primer vector de movimiento de un bloque vecino se deriva en el control de canalización, sin esperar a que se complete la derivación del segundo vector de movimiento del bloque vecino. Por lo tanto, dado que el tiempo de espera que se produce en el control de canalización realizado por el decodificador 200 se puede reducir en comparación con el caso de derivar el primer vector de movimiento utilizando el segundo vector de movimiento del bloque vecino, es posible reducir el retardo de procesamiento.

Por ejemplo, al derivar el primer vector de movimiento, el decodificador 200 (i) genera una lista de predictores de vector de movimiento que indica una pluralidad de predictores de vector de movimiento utilizando el primer

vector de movimiento del bloque procesado, y (ii) determina el primer vector de movimiento del bloque actual a partir de la pluralidad de predictores de vector de movimiento indicados en la lista de predictores de vector de movimiento (por ejemplo, S203 o S205 en la FIG. 17).

5 Por ejemplo, el modo de predicción inter en el que el decodificador 200 realiza la estimación de movimiento es un modo de fusión, y al derivar el segundo vector de movimiento, el decodificador 200 deriva el segundo vector de movimiento al realizar la estimación de movimiento en la vecindad del primer vector de movimiento (por ejemplo, S204 en la FIG. 17).

10 Por ejemplo, el modo de predicción inter en el que el decodificador 200 realiza la estimación de movimiento es un modo de conversión ascendente de la velocidad del fotograma (FRUC), y al derivar el segundo vector de movimiento, el decodificador 200 deriva el segundo vector de movimiento al realizar la estimación de movimiento en la vecindad del primer vector de movimiento (por ejemplo, S207 en la FIG. 20).

15 Por ejemplo, el modo de predicción inter en el que el decodificador 200 realiza la estimación de movimiento es un modo de conversión ascendente de la velocidad del fotograma (FRUC), y al derivar el segundo vector de movimiento, el decodificador 200 (i) determina un tercer vector de movimiento (el MVP más apropiado) a partir de la pluralidad de predictores de vector de movimiento indicados en la lista de predictores de vector de movimiento (por ejemplo, S206 en la FIG. 17), y (ii) deriva el segundo vector de movimiento al realizar una estimación de movimiento en la vecindad del tercer vector de movimiento (por ejemplo, S207 en la FIG. 17).

20 Por ejemplo, al determinar el primer vector de movimiento (por ejemplo, S205 en la FIG. 17), el decodificador 200 deriva el primer vector de movimiento en base, para cada una de las direcciones de predicción, a un promedio o una mediana de la pluralidad de predictores de vector de movimiento indicados en la lista de predictores de vector de movimiento.

25 Por ejemplo, al determinar el primer vector de movimiento (por ejemplo, S205 en la FIG. 17), entre la pluralidad de predictores de vector de movimiento indicados en la lista de predictores de vector de movimiento, el decodificador 200 determina, como el primer vector de movimiento, un predictor de vector de movimiento indicado en la parte superior de la lista de predictores de vector de movimiento.

30 Por ejemplo, al generar la lista de predictores de vector de movimiento (por ejemplo, S203 o S205 en la FIG. 17), el decodificador 200 deriva cada una de la pluralidad de predictores de vector de movimiento utilizando el primer vector de movimiento del bloque procesado o un segundo vector de movimiento del bloque procesado. Al determinar el primer vector de movimiento (por ejemplo, S205 en la FIG. 17), el decodificador 200 determina el primer vector de movimiento a partir de, entre la pluralidad de predictores de vector de movimiento indicados en la lista de predictores de vector de movimiento, uno o más candidatos de predictor de vector de movimiento derivados utilizando el segundo vector de movimiento del bloque procesado. De acuerdo con lo anterior, dado que el primer vector de movimiento se puede determinar utilizando el segundo vector de movimiento confiable, es posible suprimir una disminución en la confiabilidad del primer vector de movimiento.

35 Por ejemplo, al generar la lista de predictores de vector de movimiento (por ejemplo, S203 o S205 en la FIG. 17), el decodificador 200 obtiene un predictor de vector de movimiento utilizando el primer vector de movimiento del bloque procesado cuando el bloque procesado se incluye en un cuadro que incluye el bloque actual, y el decodificador 200 obtiene un predictor de vector de movimiento utilizando el segundo vector de movimiento del bloque procesado cuando el bloque procesado se incluye en un cuadro diferente del cuadro que incluye el bloque actual. De acuerdo con lo anterior, cuando el bloque procesado se incluye en un cuadro diferente del cuadro que incluye el bloque actual, el decodificador 200 utiliza el segundo vector de movimiento y, como resultado, se puede mejorar la fiabilidad del predictor de vector de movimiento.

40 Por ejemplo, al generar la lista de predictores de vector de movimiento (por ejemplo, S203 o S205 en la FIG. 17), el decodificador 200 determina si se debe utilizar el primer vector de movimiento del bloque procesado o un segundo vector de movimiento del bloque procesado para realizar la derivación del predictor de vector de movimiento, de acuerdo con la posición del bloque procesado relativa al bloque actual.

45 Por ejemplo, al generar la lista de predictores de vector de movimiento (por ejemplo, S203 o S205 en la FIG. 17), el decodificador 200 deriva un predictor de vector de movimiento utilizando el segundo vector de movimiento del bloque procesado cuando el bloque procesado se incluye en una unidad de procesamiento diferente de la unidad de procesamiento (por ejemplo, CTU) que incluye el bloque actual.

50 Por ejemplo, al generar la lista de predictores de vector de movimiento (por ejemplo, S203 o S205 en la FIG. 17), el decodificador 200: deriva el predictor de vector de movimiento a partir de un primer bloque procesado entre una pluralidad de bloques procesados incluidos en un cuadro que incluye el bloque actual utilizando un primer vector de movimiento del primer bloque procesado, el primer bloque procesado es de N bloques antes del bloque actual en un orden de procesamiento; deriva el predictor de vector de movimiento a partir de un segundo bloque procesado entre la pluralidad de bloques procesados utilizando un primer vector de movimiento del segundo bloque procesado, el segundo bloque procesado está después del primer bloque procesado en el orden de procesamiento; y deriva el predictor de vector de movimiento a partir de un tercer bloque procesado

entre la pluralidad de bloques procesados utilizando un segundo vector de movimiento del tercer bloque procesado, el tercer bloque procesado está antes del primer bloque procesado en el orden de procesamiento.

De acuerdo con lo anterior, el decodificador 200 utiliza el segundo vector de movimiento del tercer bloque procesado que está antes del primer bloque procesado en el orden de procesamiento y, como resultado, se puede mejorar la fiabilidad del predictor del vector de movimiento.

Por ejemplo, N es 1.

Por ejemplo, el primer vector de movimiento se utiliza además en otro procesamiento diferente de la derivación del predictor del vector de movimiento. El otro procesamiento es, por ejemplo, el filtrado de bucle.

Por ejemplo, el segundo vector de movimiento se utiliza en el filtrado de bucle.

Por ejemplo, al decodificar el bloque actual en un modo de bajo retardo, el decodificador 200 deriva el primer vector de movimiento del bloque actual utilizando el primer vector de movimiento del bloque procesado.

De acuerdo con lo anterior, el decodificador 200 puede realizar un procesamiento adecuado de acuerdo con si o no se utiliza el modo de bajo retardo.

Por ejemplo, el decodificador 200 analiza, a partir de una región de encabezado de secuencia, una región de encabezado de cuadro, una región de encabezado de segmento o una región de información auxiliar, información que indica si se debe decodificar el bloque actual en el modo de bajo retardo, y determina si se debe decodificar el bloque actual en el modo de bajo retardo en base a la información.

Por ejemplo, como se ilustra en la FIG. 15, una configuración de canalización del decodificador 200 incluye: una primera etapa en la que se deriva el primer vector de movimiento del bloque actual (segunda etapa en la FIG. 15); y una segunda etapa que es diferente de la primera etapa y en la que se deriva el segundo vector de movimiento del bloque actual (tercera etapa en la FIG. 15). El decodificador 200 comienza el procesamiento de la primera etapa para el bloque actual cuando se completa el procesamiento de la primera etapa para un bloque inmediatamente antes del bloque actual en un orden de procesamiento, sin esperar a que se complete el procesamiento de la segunda etapa para un bloque que está M bloques antes del bloque actual en el orden de procesamiento.

Por ejemplo, como se ilustra en la FIG. 18, una configuración de canalización del decodificador 200 incluye: una primera etapa en la que se deriva el primer vector de movimiento del bloque actual (segunda etapa en la FIG. 18); y una segunda etapa que es diferente de la primera etapa y en la que se deriva el segundo vector de movimiento del bloque actual (tercera etapa en la FIG. 18). El decodificador 200 comienza el procesamiento de la primera etapa para el bloque actual cuando se deriva un primer vector de movimiento de un bloque que está M bloques antes del bloque actual en un orden de procesamiento, sin esperar a que se complete el procesamiento de la segunda etapa para el bloque que está M bloques antes del bloque actual en el orden de procesamiento.

Por ejemplo, M es 1.

El codificador 100 de acuerdo con la presente Realización incluye: divisor 102 que divide una primera imagen en una pluralidad de bloques; predictor intra 124 que predice un bloque incluido en la primera imagen utilizando un bloque de referencia incluido en la primera imagen; predictor inter 126 que predice un bloque incluido en la primera imagen utilizando un bloque de referencia incluido en una segunda imagen diferente de la primera imagen; filtro de bucle 120 que aplica un filtro a los bloques incluidos en la primera imagen; transformador 106 que transforma, en coeficientes de transformación, errores de predicción entre una señal original y una señal de predicción generada por predictor intra 124 o predictor inter 126; cuantificador 108 que cuantifica los coeficientes de transformación para generar coeficientes de transformación cuantificados; y codificador de entropía 110 que genera un flujo de bits codificado mediante la codificación de longitud variable de los coeficientes cuantificados. Al codificar un bloque actual en un modo de predicción inter en el que el decodificador 200 realiza una estimación de movimiento (por ejemplo, modo de fusión o modo de FRUC en S201 en la FIG. 17), el predictor inter 126: deriva un primer vector de movimiento del bloque actual (S203 o S205); que almacena, en la memoria, el primer vector de movimiento derivado; derivar un segundo vector de movimiento del bloque actual (S204 o S207); y que genera una imagen de predicción del bloque actual al realizar una compensación de movimiento utilizando el segundo vector de movimiento (S208). Al derivar el primer vector de movimiento, el codificador 100 deriva el primer vector de movimiento del bloque actual utilizando un primer vector de movimiento de un bloque procesado.

El decodificador 200 de acuerdo con la presente realización incluye: un decodificador (decodificador de entropía 202) que decodifica un flujo de bits codificado y genera coeficientes cuantificados; un cuantificador inverso 204 que cuantifica de forma inversa los coeficientes cuantificados y envía coeficientes de transformación; un transformador inverso 206 que transforma de forma inversa los coeficientes de transformación y envía errores de predicción; un predictor intra 216 que predice un bloque incluido en una primera imagen utilizando un bloque

de referencia incluido en la primera imagen; predictor inter 218 que predice un bloque incluido en la primera imagen utilizando un bloque de referencia incluido en una segunda imagen diferente de la primera imagen; y filtro de bucle 212 que aplica un filtro a los bloques incluidos en la primera imagen. Al decodificar un bloque actual en un modo de predicción inter en el que el decodificador 200 realiza una estimación de movimiento (por ejemplo, modo de fusión o modo de FRUC en S201 en la FIG. 17), el predictor inter 218: deriva un primer vector de movimiento del bloque actual (S203 o S205); almacena, en la memoria, el primer vector de movimiento derivado; deriva un segundo vector de movimiento del bloque actual (S204 o S207); y genera una imagen de predicción del bloque actual al realizar una compensación de movimiento utilizando el segundo vector de movimiento (S208). Al derivar el primer vector de movimiento, el decodificador 200 deriva el primer vector de movimiento del bloque actual utilizando un primer vector de movimiento de un bloque procesado.

[Ejemplo de implementación del codificador]

La FIG. 23 es un diagrama de bloques que ilustra un ejemplo de implementación del codificador 100 de acuerdo con la Realización 1. El codificador 100 incluye circuitería 160 y memoria 162. Por ejemplo, la pluralidad de elementos constituyentes del codificador 100 ilustrados en la FIG. 1 se implementan mediante la circuitería 160 y la memoria 162 ilustrados en la FIG. 23.

La circuitería 160 es la circuitería que realiza el procesamiento de información y son accesibles a la memoria 162. Por ejemplo, la circuitería 160 es la circuitería electrónica exclusiva o general que codifica un vídeo. La circuitería 160 pueden ser un procesador, como una unidad central de procesamiento (CPU). La circuitería 160 puede ser un agregado de una pluralidad de circuitos electrónicos. Más aún, por ejemplo, la circuitería 160 puede realizar las funciones de dos o más elementos constituyentes entre la pluralidad de elementos constituyentes del codificador 100 ilustrado en la FIG. 1 etc., excluidos los elementos constitutivos que almacenan información.

La memoria 162 es una memoria exclusiva o general para almacenar información utilizada por la circuitería 160 para codificar un vídeo. La memoria 162 puede ser una circuitería electrónica y se puede conectar a la circuitería 160. La memoria 162 puede estar incluida en la circuitería 160. La memoria 162 puede ser un agregado de una pluralidad de circuitos electrónicos. La memoria 162 puede ser un disco magnético, un disco óptico o similar, o se puede expresar como almacenamiento, un medio de grabación o similar. La memoria 162 puede ser una memoria no volátil o puede ser una memoria volátil.

Por ejemplo, la memoria 162 puede almacenar un vídeo que se va a codificar, o puede almacenar un flujo de bits correspondiente a un vídeo codificado. La memoria 162 puede almacenar un programa para hacer que la circuitería 160 codifique un vídeo.

Por ejemplo, la memoria 162 puede realizar las funciones de, entre la pluralidad de elementos constituyentes del codificador 100 ilustrados en la FIG. 1 etc., los elementos constituyentes que almacenan información. Específicamente, la memoria 162 puede realizar las funciones de memoria de bloque 118 y memoria de fotograma 122 ilustradas en la FIG. 1. Más específicamente, la memoria 162 puede almacenar un bloque reconstruido y un cuadro reconstruido, por ejemplo.

Tenga en cuenta que no toda la pluralidad de elementos constituyentes ilustrados en la FIG. 1 etc. necesitan ser implementados por el codificador 100, y no todos los procedimientos descritos anteriormente necesitan ser realizados por el codificador 100. Algunos de los elementos constituyentes ilustrados en la FIG. 1 etc. se pueden incluir en otro dispositivo, y algunos de los procedimientos descritos anteriormente se pueden realizar por otro dispositivo. La compensación de movimiento se realiza de manera eficiente mediante el codificador 100 implementando algunos de los elementos constituyentes ilustrados en la FIG. 1 etc. y al realizar algunos de los procedimientos descritos anteriormente.

[Ejemplo de implementación del decodificador]

La FIG. 24 es un diagrama de bloques que ilustra un ejemplo de implementación del decodificador 200 de acuerdo con la Realización 1. El decodificador 200 incluye la circuitería 260 y la memoria 262. Por ejemplo, la pluralidad de elementos constituyentes del decodificador 200 ilustrados en la FIG. 10 se implementan mediante la circuitería 260 y la memoria 262 ilustrados en la FIG. 24.

La circuitería 260 es la circuitería que realiza el procesamiento de información y son accesibles a la memoria 262. Por ejemplo, la circuitería 260 es la circuitería electrónica exclusiva o general que decodifican un vídeo. La circuitería 260 pueden ser un procesador, tal como una CPU. La circuitería 260 puede ser un agregado de una pluralidad de circuitos electrónicos. Más aún, por ejemplo, la circuitería 260 puede realizar las funciones de dos o más elementos constituyentes entre la pluralidad de elementos constituyentes del decodificador 200 ilustrado en la FIG. 10, etc., excluyendo los elementos constituyentes que almacenan información.

La memoria 262 es una memoria exclusiva o general para almacenar información utilizada por la circuitería 260 para decodificar un vídeo. La memoria 262 puede ser una circuitería electrónica y se puede conectar a la circuitería 260. La memoria 262 puede estar incluida en la circuitería 260. La memoria 262 puede ser un

agregado de una pluralidad de circuitos electrónicos. La memoria 262 puede ser un disco magnético, un disco óptico o similar, o se puede expresar como almacenamiento, un medio de grabación o similar. La memoria 262 puede ser una memoria no volátil o puede ser una memoria volátil.

5 Por ejemplo, la memoria 262 puede almacenar un flujo de bits correspondiente a un vídeo codificado, o puede almacenar un vídeo correspondiente a un flujo de bits decodificado. La memoria 262 puede almacenar un programa para hacer que la circuitería 260 decodifique un vídeo.

10 Por ejemplo, la memoria 262 puede realizar las funciones de, entre la pluralidad de elementos constituyentes del decodificador 200 ilustrado en la FIG. 10, etc., los elementos constituyentes que almacenan información. Específicamente, la memoria 262 puede realizar las funciones de memoria de bloques 210 y memoria de fotogramas 214 ilustradas en la FIG. 10. Más específicamente, la memoria 262 puede almacenar un bloque reconstruido y un cuadro reconstruido, por ejemplo.

15 Tenga en cuenta que no toda la pluralidad de elementos constituyentes ilustrados en la FIG. 10, etc. necesitan ser implementados por el decodificador 200, y no todos los procedimientos descritos anteriormente necesitan ser realizados por el decodificador 200. Algunos de los elementos constituyentes ilustrados en la FIG. 10, etc. se pueden incluir en otro dispositivo, y algunos de los procedimientos descritos anteriormente se pueden realizar por otro dispositivo. La compensación de movimiento se realiza de manera eficiente mediante el decodificador 200 implementando algunos de los elementos constituyentes ilustrados en la FIG. 10, etc. y al realizar algunos de los procedimientos descritos anteriormente.

[Información adicional]

20 El codificador 100 y el decodificador 200 de acuerdo con la presente realización se pueden utilizar como un codificador de imágenes y un decodificador de imágenes, respectivamente, o se pueden utilizar como un codificador de vídeo y un decodificador de vídeo, respectivamente. Alternativamente, el codificador 100 y el decodificador 200 se pueden utilizar cada uno como un dispositivo de predicción inter (un dispositivo de predicción de fotograma inter).

25 Es decir, el codificador 100 y el decodificador 200 pueden corresponder únicamente al predictor inter (predictor de fotograma inter) 126 y al predictor inter (predictor de fotograma inter) 218, respectivamente. Los demás elementos constituyentes, tal como el transformador 106 y el transformador inverso 206, se pueden incluir en otro dispositivo.

30 En la presente realización, cada uno de los elementos constituyentes se puede configurar en forma de un producto de hardware exclusivo, o se puede implementar al ejecutar un programa de software adecuado para el elemento constituyente. Cada uno de los elementos constituyentes se puede implementar por medio de una unidad de ejecución de programa, tal como una CPU o un procesador, que lee y ejecuta un programa de software grabado en un medio de grabación tal como un disco duro o una memoria de semiconductores.

35 Específicamente, el codificador 100 y el decodificador 200 pueden incluir cada uno circuitería de procesamiento y almacenamiento conectados eléctricamente a la circuitería de procesamiento y accesibles desde la circuitería de procesamiento. Por ejemplo, la circuitería de procesamiento corresponden a la circuitería 160 o 260, y el almacenamiento corresponde a la memoria 162 o 262.

40 La circuitería de procesamiento incluyen al menos uno de un producto de hardware exclusivo o una unidad de ejecución de programa, y realizan el procesamiento utilizando el almacenamiento. Cuando la circuitería de procesamiento incluyen una unidad de ejecución de programa, el almacenamiento almacena un programa de software ejecutado por la unidad de ejecución de programa.

En la presente memoria, el software para implementar, por ejemplo, el codificador 100 o el decodificador 200 de acuerdo con la presente realización es un programa como el siguiente:

45 Los elementos constituyentes pueden ser circuitos como se describió anteriormente. Los circuitos pueden constituir un circuito en su totalidad, o pueden ser circuitos individuales. Cada elemento constituyente se puede implementar por un procesador general, o se puede implementar por un procesador exclusivo.

50 Más aún, el procesamiento ejecutado por un elemento constituyente particular puede ser ejecutado por otro elemento constituyente. El orden de ejecución del procesamiento puede ser modificado, o una pluralidad de procedimientos pueden ser ejecutados en paralelo. Además, un dispositivo de codificación y decodificación puede incluir el codificador 100 y el decodificador 200.

Aunque algunos aspectos del codificador 100 y del decodificador 200 se han descrito anteriormente en base a una realización, los aspectos del codificador 100 y del decodificador 200 no están limitados a esta realización. Varias modificaciones a esta realización que son concebibles para aquellos expertos en la técnica, así como realizaciones resultantes de combinaciones de elementos constituyentes en diferentes realizaciones, se

pueden incluir dentro del alcance de los aspectos del codificador 100 y del decodificador 200, siempre que no se aparten de la esencia de la presente divulgación.

Este aspecto se puede implementar en combinación con uno o más de los otros aspectos de acuerdo con la presente divulgación. Además, parte de los procedimientos en los diagramas de flujo, parte de los elementos constituyentes de los aparatos y parte de la sintaxis descrita en este aspecto se pueden implementar en combinación con otros aspectos.

Realización 2

Como se describe en la realización y las variaciones anteriores, cada bloque funcional se puede realizar típicamente como una MPU y una memoria, por ejemplo. Más aún, los procedimientos realizados por cada uno de los bloques funcionales se realizan típicamente por una unidad de ejecución de programa, tal como un procesador, que lee y ejecuta software (un programa) grabado en un medio de grabación tal como ROM. El software se puede distribuir, por ejemplo, mediante descarga, y se puede grabar en un medio de grabación, tal como una memoria de semiconductores, y distribuir. Tenga en cuenta que cada bloque funcional también puede, por supuesto, realizarse como hardware (circuito dedicado).

Más aún, el procesamiento descrito en la realización y las variaciones anteriores se puede realizar mediante procesamiento integrado utilizando un único aparato (sistema) y, alternativamente, se puede realizar a través del procesamiento descentralizado utilizando una pluralidad de aparatos. Más aún, el procesador que ejecuta el programa descrito anteriormente puede ser un único procesador o una pluralidad de procesadores. En otras palabras, se puede realizar un procesamiento integrado y, alternativamente, se puede realizar un procesamiento descentralizado.

Las realizaciones de la presente divulgación no se limitan a la realización ejemplar y las variaciones anteriores; se pueden realizar diversas modificaciones a la realización ejemplar y las variaciones, cuyos resultados también se incluyen dentro del alcance de la realización y las variaciones de la presente divulgación.

A continuación, se describirán ejemplos de aplicación del procedimiento de codificación de cuadros en movimiento (procedimiento de codificación de imágenes) y el procedimiento de decodificación de cuadros en movimiento (procedimiento de decodificación de imágenes) descritos en la realización y las variaciones anteriores y un sistema que emplea los mismos. El sistema se caracteriza por incluir un codificador de imágenes que emplea el procedimiento de codificación de imágenes, un decodificador de imágenes que emplea el procedimiento de decodificación de imágenes y un codificador/decodificador de imágenes que incluye tanto el codificador de imágenes como el decodificador de imágenes. Otras configuraciones incluidas en el sistema se pueden modificar en una base caso por caso.

[Ejemplos de uso]

La FIG. 25 ilustra una configuración general del sistema de provisión de contenido ex100 para implementar un servicio de distribución de contenido. El área en la que se proporciona el servicio de comunicación se divide en celdas de tamaños deseados, y las estaciones base ex106, ex107, ex108, ex109 y ex110, que son estaciones inalámbricas fijas, se ubican en celdas respectivas.

En el sistema de provisión de contenido ex100, los dispositivos que incluyen el ordenador ex111, el dispositivo de juego ex112, la cámara ex113, el electrodoméstico ex114 y el teléfono inteligente ex115 se conectan a Internet ex101 a través del proveedor de servicios de Internet ex102 o la red de comunicaciones ex104 y las estaciones base ex106 a ex110. El sistema de provisión de contenido ex100 puede combinar y conectar cualquier combinación de los elementos anteriores. Los dispositivos pueden estar conectados entre sí directa o indirectamente a través de una red telefónica o comunicación de campo cercano en lugar de a través de las estaciones base ex106 a ex110, que son estaciones inalámbricas fijas. Más aún, el servidor de retransmisión ex103 está conectado a dispositivos que incluyen un ordenador ex111, un dispositivo de juego ex112, una cámara ex113, un electrodoméstico ex114 y un teléfono inteligente ex115 a través de, por ejemplo, Internet ex101. El servidor de retransmisión ex103 también está conectado, por ejemplo, a una terminal en un punto de acceso en un avión ex117 a través del satélite ex116.

Tenga en cuenta que en lugar de las estaciones base ex106 a ex110, se pueden utilizar puntos de acceso inalámbricos o puntos de acceso. El servidor de retransmisión ex103 se puede conectar a la red de comunicaciones ex104 directamente en lugar de a través de Internet ex101 o del proveedor de servicios de Internet ex102, y se puede conectar al avión ex117 directamente en lugar de a través del satélite ex116.

La cámara ex113 es un dispositivo capaz de capturar imágenes fijas y vídeo, tal como una cámara digital. El teléfono inteligente ex115 es un dispositivo de teléfono inteligente, teléfono celular o teléfono de sistema de telefonía personal (PHS) que puede operar bajo los estándares de sistemas de comunicaciones móviles de los sistemas típicos 2G, 3G, 3.9G y 4G, así como del sistema 5G de próxima generación.

El electrodoméstico ex118 es, por ejemplo, un refrigerador o un dispositivo incluido en un sistema de cogeneración de celdas de combustible para el hogar.

En el sistema de provisión de contenido ex100, un terminal que incluye una función de captura de imágenes y/o vídeos es capaz de, por ejemplo, realizar retransmisiones en vivo mediante la conexión al servidor de transmisión ex103 a través de, por ejemplo, la estación base ex106. Cuando se realiza una retransmisión en vivo, un terminal (por ejemplo, ordenador ex111, dispositivo de juego ex112, cámara ex113, electrodoméstico ex114, teléfono inteligente ex115 o avión ex117) realiza el procesamiento de codificación descrito en la realización anterior y variaciones sobre el contenido de imagen fija o vídeo capturado por un usuario a través del terminal, multiplexa datos de vídeo obtenidos a través de la codificación y datos de audio obtenidos al codificar el audio correspondiente al vídeo, y transmite los datos obtenidos al servidor de retransmisión ex103. En otras palabras, el terminal funciona como el codificador de imágenes de acuerdo con un aspecto de la presente divulgación.

El servidor de retransmisión ex103 transmite datos de contenido transmitidos a clientes que solicitan la transmisión. Los ejemplos de clientes incluyen ordenador ex111, dispositivo de juego ex112, cámara ex113, electrodoméstico ex114, teléfono inteligente ex115 y terminales dentro del avión ex117, que son capaces de decodificar los datos codificados descritos anteriormente. Los dispositivos que reciben los datos transmitidos decodifican y reproducen los datos recibidos. En otras palabras, cada dispositivo funciona como decodificador de imágenes de acuerdo con un aspecto de la presente divulgación.

[Procesamiento descentralizado]

El servidor de transmisión ex103 se puede realizar como una pluralidad de servidores u ordenadores entre las cuales se dividen tareas tales como el procesamiento, la grabación y la retransmisión de datos. Por ejemplo, el servidor de retransmisión ex103 se puede realizar como una red de distribución de contenido (CDN) que transmite contenido a través de una red que conecta múltiples servidores de borde ubicados en todo el mundo. En una CDN, un servidor de borde físicamente cerca del cliente se asigna dinámicamente al cliente. El contenido se almacena en caché y se transmite al servidor de borde para reducir los tiempos de carga. En el caso, por ejemplo, de algún tipo de error o un cambio en la conectividad debido a, por ejemplo, un pico en el tráfico, es posible transmitir datos de manera estable a altas velocidades ya que es posible evitar partes afectadas de la red al, por ejemplo, dividir el procesamiento entre una pluralidad de servidores de borde o cambiar las tareas de retransmisión a un servidor de borde diferente y continuar con la retransmisión.

La descentralización no se limita solo a la división del procesamiento para la transmisión; La codificación de los datos capturados se puede dividir entre los terminales y realizarse por ellos, en el lado del servidor o en ambos. En un ejemplo, en la codificación típica, el procesamiento se realiza en dos bucles. El primer bucle sirve para detectar la complejidad de la imagen fotograma por fotograma o escena por escena, o para detectar la carga de codificación. El segundo bucle sirve para el procesamiento que mantiene la calidad de la imagen y mejora la eficiencia de codificación. Por ejemplo, es posible reducir la carga de procesamiento de los terminales y mejorar la calidad y la eficiencia de codificación del contenido al hacer que los terminales realicen el primer bucle de la codificación y que el lado del servidor que recibió el contenido realice el segundo bucle de la codificación. En dicho caso, tras la recepción de una solicitud de decodificación, es posible que los datos codificados resultantes del primer bucle realizado por un terminal se reciban y reproduzcan en otro terminal aproximadamente en tiempo real. Esto hace posible realizar una transmisión fluida en tiempo real.

En otro ejemplo, la cámara ex113 o similar extrae una cantidad de características de una imagen, comprime datos relacionados con la cantidad de características como metadatos y transmite los metadatos comprimidos a un servidor. Por ejemplo, el servidor determina la importancia de un objeto en base a la cantidad de características y cambia la precisión de cuantificación de acuerdo con lo anterior para realizar una compresión adecuada para el significado de la imagen. Los datos de cantidad de características son particularmente eficaces para mejorar la precisión y la eficiencia de la predicción del vector de movimiento durante la segunda pasada de compresión realizado por el servidor. Más aún, la codificación que tiene una carga de procesamiento relativamente baja, como la codificación de longitud variable (VLC), se puede manejar por el terminal, y la codificación que tiene una carga de procesamiento relativamente alta, como la codificación aritmética binaria adaptativa al contexto (CABAC), puede ser manejada por el servidor.

En todavía otro ejemplo, hay casos en los que una pluralidad de terminales capturan una pluralidad de vídeos de aproximadamente la misma escena en, por ejemplo, un estadio, un centro comercial o una fábrica. En dicho caso, por ejemplo, la codificación se puede descentralizar al dividir las tareas de procesamiento entre la pluralidad de terminales que capturaron los vídeos y, si es necesario, otros terminales que no capturaron los vídeos y el servidor, sobre una base unitaria. Las unidades pueden ser, por ejemplo, grupos de cuadros (GOP), cuadros o mosaicos resultantes de dividir un cuadro. Esto hace posible reducir los tiempos de carga y lograr una transmisión más cercana al tiempo real.

Más aún dado que los vídeos son aproximadamente de la misma escena, el servidor puede llevar a cabo la gestión y/o instrucción de modo que los vídeos capturados por los terminales se puedan referenciar de forma

cruzada. Más aún, el servidor puede recibir datos codificados de los terminales, cambiar la relación de referencia entre ítems de datos o corregir o reemplazar las propios cuadros, y luego realizar la codificación. Esto hace posible generar un flujo con mayor calidad y eficiencia para los ítems de datos individuales.

- 5 Más aún, el servidor puede transmitir datos de vídeo después de realizar la transcodificación para convertir el formato de codificación de los datos de vídeo. Por ejemplo, el servidor puede convertir el formato de codificación de MPEG a VP y puede convertir H.264 a H.265.

- 10 De esta manera, la codificación se puede realizar por un terminal o uno o más servidores. De acuerdo con lo anterior, aunque el dispositivo que realiza la codificación se denomina como "servidor" o "terminal" en la siguiente descripción, algunos o todos los procedimientos realizados por el servidor se pueden realizar por el terminal, y de la misma manera algunos o todos los procedimientos realizados por el terminal se pueden realizar por el servidor. Esto también se aplica a los procedimientos de decodificación.

[3D, Multiángulo]

- 15 En los últimos años, ha aumentado el uso de imágenes o vídeos combinados a partir de imágenes o vídeos de diferentes escenas capturadas simultáneamente o la misma escena capturada desde diferentes ángulos por una pluralidad de terminales como la cámara ex113 y/o el teléfono inteligente ex115. Los vídeos capturados por los terminales se combinan en base a, por ejemplo, la relación posicional relativa obtenida por separado entre los terminales, o regiones en un vídeo que tiene puntos de características coincidentes.

- 20 Además de la codificación de cuadros en movimiento bidimensionales, el servidor puede codificar una imagen fija en base al análisis de la escena de un cuadro en movimiento, ya sea automáticamente o en un punto en el tiempo especificado por el usuario, y transmitir la imagen fija codificada a un terminal de recepción. Además, cuando el servidor puede obtener la relación de posición relativa entre los terminales de captura de vídeo, además de cuadros en movimiento bidimensionales, el servidor puede generar la geometría tridimensional de una escena en base al vídeo de la misma escena capturado desde diferentes ángulos. Tenga en cuenta que el servidor puede codificar por separado datos tridimensionales generados a partir de, por ejemplo, una nube de puntos, y puede, en base a un resultado de reconocer o rastrear a una persona u objeto utilizando datos tridimensionales, seleccionar o reconstruir y generar un vídeo para ser transmitido a un terminal de recepción a partir de vídeos capturados por una pluralidad de terminales.

- 30 Esto permite al usuario disfrutar de una escena al seleccionar libremente los vídeos correspondientes a los terminales de captura de vídeo, y permite al usuario disfrutar del contenido obtenido al extraer, a partir de datos tridimensionales reconstruidos a partir de una pluralidad de imágenes o vídeos, un vídeo desde un punto de vista seleccionado. Además, de forma similar a lo que ocurre con el vídeo, el sonido puede grabarse desde ángulos relativamente diferentes, y el servidor puede multiplexar, con el vídeo, el audio desde un ángulo o espacio específico de acuerdo con el vídeo, y transmitir el resultado.

- 35 En los últimos años, el contenido que es una composición del mundo real y un mundo virtual, tal como el contenido de realidad virtual (VR) y realidad aumentada (AR), también se ha vuelto popular. En el caso de imágenes de VR, el servidor puede crear imágenes desde los puntos de vista de los ojos izquierdo y derecho y realizar una codificación que tolere la referencia entre las dos imágenes de punto de vista, tal como la codificación multivista (MVC), y, alternativamente, puede codificar las imágenes como flujos separados sin referencia. Cuando las imágenes se decodifican como flujos separados, los flujos se pueden sincronizar cuando se reproducen para recrear un espacio tridimensional virtual de acuerdo con el punto de vista del usuario.

- 40 En el caso de imágenes de AR, el servidor superpone información de objeto virtual existente en un espacio virtual sobre información de cámara que representa un espacio del mundo real, en base a una posición o movimiento tridimensional desde la perspectiva del usuario. El decodificador puede obtener o almacenar información de objeto virtual y datos tridimensionales, generar imágenes bidimensionales en base al movimiento desde la perspectiva del usuario y luego generar datos superpuestos conectando las imágenes sin problemas. Alternativamente, el decodificador puede transmitir, al servidor, movimiento desde la perspectiva del usuario además de una solicitud de información de objeto virtual, y el servidor puede generar datos superpuestos en base a datos tridimensionales almacenados en el servidor de acuerdo con el movimiento recibido, y codificar y transmitir los datos superpuestos generados al decodificador. Tenga en cuenta que los datos superpuestos incluyen, además de los valores RGB, un valor α que indica transparencia, y el servidor establece el valor α para las secciones distintas del objeto generado a partir de datos tridimensionales en, por ejemplo, 0, y puede realizar la codificación mientras esas secciones son transparentes. Alternativamente, el servidor puede establecer el fondo en un valor RGB predeterminado, como una clave de croma, y generar datos en los que se establezcan como fondo áreas distintas del objeto.

- 55 La decodificación de datos transmitidos de manera similar se puede realizar por el cliente (es decir, los terminales), en el lado del servidor, o dividida entre ellos. En un ejemplo, un terminal puede transmitir una solicitud de recepción a un servidor, el contenido solicitado se puede recibir y decodificar por otro terminal, y una señal decodificada se puede transmitir a un dispositivo que tiene una pantalla. Es posible reproducir datos

de alta calidad de imagen al descentralizar el procesamiento y al seleccionar apropiadamente el contenido independientemente de la capacidad de procesamiento del propio terminal de comunicaciones. En todavía otro ejemplo, mientras un televisor, por ejemplo, está recibiendo datos de imagen que son de gran tamaño, una región de un cuadro, tal como un mosaico obtenido al dividir el cuadro, puede ser decodificada y mostrada en un terminal o terminales personales de un espectador o espectadores del televisor. Esto hace posible que los espectadores compartan una vista de cuadro grande, así como que cada espectador verifique su área asignada o inspeccione una región con más detalle de cerca.

En el futuro, tanto en interiores como en exteriores, en situaciones en las que sea posible una pluralidad de conexiones inalámbricas a distancias cercanas, medias y lejanas, se espera poder recibir contenido sin problemas incluso cuando se cambia a datos apropiados para la conexión actual, utilizando un estándar de sistema de retransmisión tal como MPEG-DASH. Con esto, el usuario puede cambiar entre datos en tiempo real mientras selecciona libremente un decodificador o aparato de visualización que incluye no solo su propio terminal, sino también, por ejemplo, pantallas dispuestas en interiores o exteriores. Más aún, en base a, por ejemplo, la información sobre la posición del usuario, se puede realizar la decodificación mientras se intercambia qué terminal maneja la decodificación y qué terminal maneja la visualización del contenido. Esto hace posible, mientras se está en ruta hacia un destino, mostrar, en la pared de un edificio cercano en el que está incrustado un dispositivo capaz de mostrar contenido o en parte del suelo, información cartográfica mientras se está en movimiento. Más aún, también es posible cambiar la velocidad de bits de los datos recibidos en base a la accesibilidad a los datos codificados en una red, como cuando los datos codificados se almacenan en caché en un servidor al que se puede acceder rápidamente desde el terminal de recepción o cuando los datos codificados se copian en un servidor de borde en un servicio de suministro de contenido.

[Codificación escalable]

El intercambio de contenido se describirá con referencia a un flujo escalable, ilustrado en la FIG. 26, que se codifica por compresión a través de la implementación del procedimiento de codificación de cuadros en movimiento descrito en la realización y variaciones anteriores. El servidor puede tener una configuración en la que el contenido se cambia mientras se hace uso de la escalabilidad temporal y/o espacial de un flujo, lo que se logra mediante la división en capas y la codificación de capas, como se ilustra en la FIG. 26. Tenga en cuenta que puede haber una pluralidad de flujos individuales que son del mismo contenido pero de diferente calidad. En otras palabras, al determinar hasta qué capa se decodificará en base a factores internos, tales como la capacidad de procesamiento del lado del decodificador, y factores externos, tales como el ancho de banda de comunicación, el lado del decodificador se puede intercambiar libremente entre contenido de baja resolución y contenido de alta resolución durante la decodificación. Por ejemplo, en un caso en el que el usuario desea continuar viendo, en casa en un dispositivo tal como un TV conectado a Internet, un vídeo que había estado viendo previamente en el teléfono inteligente ex115 mientras estaba en movimiento, el dispositivo puede simplemente decodificar la misma transmisión hasta una capa diferente, lo que reduce la carga del lado del servidor.

Adicionalmente, además de la configuración descrita anteriormente en la que se logra la escalabilidad como resultado de que los cuadros se codifican por capa y la capa de mejora está por encima de la capa base, la capa de mejora puede incluir metadatos basados, por ejemplo, en información estadística sobre la imagen, y el lado del decodificador puede generar contenido de alta calidad de imagen al realizar formación de imágenes de súper resolución en un cuadro en la capa base en base a los metadatos. Las imágenes de súper resolución pueden mejorar la relación SN mientras se mantiene la resolución y/o se aumenta la resolución. Los metadatos incluyen información para identificar un coeficiente de filtro lineal o no lineal utilizado en el procesamiento de súper resolución, o información que identifica un valor de parámetro en el procesamiento de filtro, aprendizaje automático o procedimiento de mínimos cuadrados utilizado en el procesamiento de súper resolución.

Alternativamente, también es aceptable una configuración en la que un cuadro se divide en, por ejemplo, mosaicos de acuerdo con el significado de, por ejemplo, un objeto en la imagen, y en el lado del decodificador, solo se decodifica una región parcial al seleccionar un mosaico para decodificar. Más aún, al almacenar un atributo sobre el objeto (persona, automóvil, pelota, etc.) y una posición del objeto en el vídeo (coordenadas en imágenes idénticas) como metadatos, el lado del decodificador puede identificar la posición de un objeto deseado en base a los metadatos y determinar qué mosaico o mosaicos incluyen ese objeto. Por ejemplo, como se ilustra en la FIG. 27, los metadatos se almacenan utilizando una estructura de almacenamiento de datos diferente de los datos de píxeles, tal como un mensaje SEI en HEVC. Estos metadatos indican, por ejemplo, la posición, el tamaño o el color del objeto principal.

Más aún, los metadatos se pueden almacenar en unidades de una pluralidad de cuadros, tales como unidades de flujo, secuencia o acceso aleatorio. Con esto, el lado del decodificador puede obtener, por ejemplo, el momento en el que una persona específica aparece en el vídeo y, al ajustarlo con información de la unidad de cuadro, puede identificar un cuadro en la que está presente el objeto y la posición del objeto en el cuadro.

[Optimización de página web]

La FIG. 28 ilustra un ejemplo de una pantalla de visualización de una página web en, por ejemplo, el ordenador ex111. La FIG. 29 ilustra un ejemplo de una pantalla de visualización de una página web en, por ejemplo, el teléfono inteligente ex115. Como se ilustra en la FIG. 28 y la FIG. 29, una página web puede incluir una pluralidad de enlaces de imagen que son enlaces a contenido de imagen, y la apariencia de la página web difiere dependiendo del dispositivo utilizado para ver la página web. Cuando se pueden visualizar en la pantalla una pluralidad de enlaces de imagen, hasta que el usuario selecciona explícitamente un enlace de imagen, o hasta que el enlace de imagen se encuentra en el centro aproximado de la pantalla o hasta que el enlace de imagen completo cabe en la pantalla, el aparato de visualización (decodificador) muestra, como enlaces de imagen, imágenes fijas incluidas en el contenido o cuadros I, muestra vídeo como un gif animado que utiliza una pluralidad de imágenes fijas o cuadros I, por ejemplo, o recibe sólo la capa base y decodifica y muestra el vídeo.

Cuando el usuario selecciona un enlace de imagen, el aparato de visualización decodifica dando la máxima prioridad a la capa base. Tenga en cuenta que si hay información en el código HTML de la página web que indica que el contenido es escalable, el aparato de visualización puede decodificar hasta la capa de mejora. Además, para garantizar la reproducción en tiempo real, antes de realizar una selección o cuando el ancho de banda está severamente limitado, el aparato de visualización puede reducir el retardo entre el punto en el tiempo en el que se decodifica el cuadro principal y el punto en el tiempo en el que se muestra el cuadro decodificado (es decir, el retardo entre el inicio de la decodificación del contenido y la visualización del contenido) decodificando y mostrando solo cuadros de referencia hacia adelante (cuadro I, cuadro P, cuadro B de referencia hacia adelante). Más aún, el aparato de visualización puede ignorar deliberadamente la relación de referencia entre cuadros y decodificar de manera aproximada todos los cuadros B y P como cuadros de referencia hacia adelante, y luego realizar la decodificación normal a medida que aumenta el número de cuadros recibidos con el tiempo.

[Conducción autónoma]

Al transmitir y recibir datos de imagen fija o de vídeo, como información de mapa bidimensional o tridimensional para la conducción autónoma o la conducción asistida de un automóvil, el terminal de recepción puede recibir, además de datos de imagen pertenecientes a una o más capas, información sobre, por ejemplo, el clima o la construcción de carreteras como metadatos, y asociar los metadatos con los datos de imagen al decodificarse. Tenga en cuenta que los metadatos se pueden asignarse por capa y, alternativamente, se pueden simplemente multiplexar con los datos de imagen.

En dicho caso, dado que el automóvil, dron, avión, etc., incluido el terminal de recepción es móvil, el terminal de recepción puede recibir y decodificar sin problemas mientras cambia entre estaciones base entre las estaciones base ex106 a ex110 transmitiendo información que indica la posición del terminal de recepción al recibir una solicitud de recepción. Más aún, de acuerdo con la selección realizada por el usuario, la situación del usuario o el ancho de banda de la conexión, el terminal de recepción puede seleccionar dinámicamente en qué medida se reciben los metadatos o en qué medida se actualiza, por ejemplo, la información del mapa.

Con esto, en el sistema de provisión de contenido ex100, el cliente puede recibir, decodificar y reproducir, en tiempo real, información codificada transmitida por el usuario.

[Transmisión de contenido individual]

En el sistema de provisión de contenido ex100, además de contenido largo de alta calidad de imagen distribuido por una entidad de distribución de vídeo, también es posible la retransmisión de unidifusión o multidifusión de contenido corto de baja calidad de imagen de un individuo. Más aún, es probable que dicho contenido de individuos aumente aún más en popularidad. El servidor puede realizar primero un procesamiento de edición en el contenido antes del procesamiento de codificación para refinar el contenido individual. Esto se puede lograr, por ejemplo, con la siguiente configuración.

En tiempo real, mientras se captura contenido de vídeo o imagen o después de que el contenido se haya capturado y acumulado, el servidor realiza un procesamiento de reconocimiento en base a los datos sin procesar o codificados, tal como procesamiento de error de captura, procesamiento de búsqueda de escena, análisis de significado y/o procesamiento de detección de objetos. A continuación, en base al resultado del procesamiento de reconocimiento, el servidor, ya sea cuando se le solicite o de forma automática, edita el contenido, cuyos ejemplos incluyen: corrección como la corrección del enfoque y/o del desenfoque de movimiento; eliminación de escenas de baja prioridad, tales como escenas con poco brillo en comparación con otros cuadros o que están desenfocadas; ajuste de bordes de los objetos; y ajuste del tono de color. El servidor codifica los datos editados en base al resultado de la edición. Se sabe que los vídeos excesivamente largos tienden a recibir menos visualizaciones. De acuerdo con lo anterior, para mantener el contenido dentro de una duración específica que se ajuste a la duración del vídeo original, el servidor puede, además de las escenas de baja prioridad descritas anteriormente, recortar automáticamente las escenas con poco movimiento en base a un resultado de procesamiento de imágenes. Alternativamente, el servidor puede generar y codificar un resumen de vídeo en base a un resultado de un análisis del significado de una escena.

Tenga en cuenta que existen casos en los que el contenido individual puede incluir contenido que infringe un derecho de autor, un derecho moral, derechos de retrato, etc. Un caso de este tipo puede conducir a una situación desfavorable para el creador, tal como cuando el contenido se comparte más allá del alcance previsto por el creador. De acuerdo con lo anterior, antes de la codificación, el servidor puede, por ejemplo, editar imágenes para desenfocar las caras de las personas en la periferia de la pantalla o desenfocar el interior de una casa, por ejemplo. Más aún, el servidor se puede configurar para reconocer las caras de personas distintas de una persona registrada en imágenes que se van a codificar, y cuando dichas caras aparecen en una imagen, por ejemplo, aplicar un filtro de mosaico a la cara de la persona. Alternativamente, como preprocesamiento o posprocesamiento para la codificación, el usuario puede especificar, por razones de derechos de autor, que se procese una región de una imagen que incluya una persona o una región del fondo, y el servidor puede procesar la región especificada, por ejemplo, al reemplazar la región con una imagen diferente o desenfocando la región. Si la región incluye una persona, es posible seguirla en el cuadro en movimiento y la región de la cabeza se puede reemplazar por otra imagen a medida que la persona se mueve.

Más aún, dado que existe una demanda de visualización en tiempo real de contenido producido por individuos, que tiende a ser pequeño en tamaño de datos, el decodificador recibe primero la capa base como la prioridad más alta y realiza la decodificación y reproducción, aunque esto puede diferir dependiendo del ancho de banda. Cuando el contenido se reproduce dos o más veces, tal como cuando el decodificador recibe la capa de mejora durante la decodificación y reproducción de la capa base y repite la reproducción, el decodificador puede reproducir un vídeo de alta calidad de imagen que incluye la capa de mejora. Si la transmisión se codifica utilizando dicha codificación escalable, el vídeo puede ser de baja calidad cuando está en un estado no seleccionado o al comienzo del vídeo, pero puede ofrecer una experiencia en la que la calidad de imagen de la transmisión aumenta progresivamente de una manera inteligente. Esto no se limita solo a la codificación escalable; la misma experiencia se puede ofrecer al configurar un solo flujo a partir de un flujo de baja calidad reproducido por primera vez y un segundo flujo codificado utilizando el primer flujo como referencia.

[Otros ejemplos de uso]

La codificación y decodificación se pueden realizar por LSI ex500, que normalmente se incluye en cada terminal. LSI ex500 se puede configurar con un solo chip o una pluralidad de chips. El software para codificar y decodificar cuadros en movimiento puede estar integrado en algún tipo de medio de grabación (como un CD-ROM, un disco flexible o un disco duro) que sea legible por, por ejemplo, el ordenador ex111, y la codificación y decodificación se pueden realizar utilizando el software. Además, cuando el teléfono inteligente ex115 está equipado con una cámara, se pueden transmitir los datos de vídeo obtenidos por la cámara. En este caso, los datos de vídeo son codificados por LSI ex500 incluido en el teléfono inteligente ex115.

Tenga en cuenta que LSI ex500 se puede configurar para descargar y activar una aplicación. En dicho caso, el terminal primero determina si es compatible con el esquema utilizado para codificar el contenido o si es capaz de ejecutar un servicio específico. Cuando el terminal no es compatible con el esquema de codificación del contenido o cuando el terminal no es capaz de ejecutar un servicio específico, el terminal primero descarga un códec o software de aplicación y luego obtiene y reproduce el contenido.

Aparte del ejemplo del sistema de provisión de contenido ex100 que utiliza Internet ex101, al menos el codificador de cuadros en movimiento (codificador de imágenes) o el decodificador de cuadros en movimiento (decodificador de imágenes) descritos en la realización anterior y las variaciones se pueden implementar en un sistema de transmisión digital. El mismo procesamiento de codificación y decodificación se puede aplicar para transmitir y recibir ondas de radio de difusión superpuestas con datos de audio y vídeo multiplexados utilizando, por ejemplo, un satélite, aunque esto está orientado hacia la multidifusión mientras que la unidifusión es más fácil con el sistema de provisión de contenido ex100.

[Configuración de hardware]

La FIG. 30 ilustra el teléfono inteligente ex115. La FIG. 31 ilustra un ejemplo de configuración del teléfono inteligente ex115. El teléfono inteligente ex115 incluye una antena ex450 para transmitir y recibir ondas de radio hacia y desde la estación base ex110, una cámara ex465 capaz de capturar vídeo e imágenes fijas, y una pantalla ex458 que muestra datos decodificados, tales como el vídeo capturado por la cámara ex465 y el vídeo recibido por la antena ex450. El teléfono inteligente ex115 incluye además una interfaz de usuario ex466, tal como un panel táctil, una unidad de salida de audio ex457, tal como un altavoz para emitir voz u otro audio, una unidad de entrada de audio ex456, como un micrófono para la entrada de audio, una memoria ex467 capaz de almacenar datos decodificados, tal como vídeo capturado o imágenes fijas, audio grabado, vídeo recibido o imágenes fijas, y correo, así como datos decodificados, y un intervalo ex464, que es una interfaz para la tarjeta SIM ex468 para autorizar el acceso a una red y a varios datos. Téngase en cuenta que se puede utilizar una memoria externa en lugar de la memoria ex467.

Más aún, el controlador principal ex460 que controla de manera integral la pantalla ex458 y la interfaz de usuario ex466, el circuito de suministro de energía ex461, el controlador de entrada de la interfaz de usuario ex462, el procesador de señal de vídeo ex455, la interfaz de cámara ex463, el controlador de pantalla ex459, el

modulador/demodulador ex452, el multiplexor/demultiplexor ex453, el procesador de señal de audio ex454, el intervalo ex464 y la memoria ex467 están conectados a través del bus ex470.

5 Cuando el usuario enciende el botón de encendido del circuito de suministro de energía ex461, el teléfono inteligente ex115 se enciende y pasa a un estado operativo al recibir energía de un paquete de baterías para cada componente.

10 El teléfono inteligente ex115 realiza un procesamiento para, por ejemplo, realizar llamadas y transmitir datos, en base al control realizado por el controlador principal ex460, que incluye una CPU, ROM y RAM. Al realizar llamadas, una señal de audio grabada por la unidad de entrada de audio ex456 se convierte en una señal de audio digital por el procesador de señal de audio ex454, y esto se aplica con procesamiento de espectro disperso por el modulador/demodulador ex452 y procesamiento de conversión digital-analógica y conversión de frecuencia por el transmisor/receptor ex451, y luego se transmite a través de la antena ex450. Los datos recibidos se amplifican, se convierten en frecuencia y se convierten de analógico a digital, el modulador/demodulador ex452 procesa el espectro disperso inverso, el procesador de señal de audio ex454 los convierte en una señal de audio analógica, y luego se emiten desde la unidad de salida de audio ex457. En 15 el modo de transmisión de datos, los datos de texto, imagen fija o vídeo se transmiten por el controlador principal ex460 a través del controlador de entrada de interfaz de usuario ex462 como resultado de la operación de, por ejemplo, la interfaz de usuario ex466 del cuerpo principal, y se realiza un procesamiento de transmisión y recepción similar. En el modo de transmisión de datos, al enviar un vídeo, una imagen fija o un vídeo y audio, el procesador de señal de vídeo ex455 codifica por compresión, a través del procedimiento de codificación de cuadros en movimiento descrito en la Realización y variaciones anteriores, una señal de vídeo almacenada en la memoria ex467 o una señal de vídeo introducida desde la cámara ex465, y transmite los datos de vídeo codificados al multiplexor/demultiplexor ex453. Más aún, el procesador de señal de audio ex454 codifica una 20 señal de audio grabada por la unidad de entrada de audio ex456 mientras la cámara ex465 está capturando, por ejemplo, un vídeo o una imagen fija, y transmite los datos de audio codificados al multiplexor/demultiplexor ex453. El multiplexor/demultiplexor ex453 multiplexa los datos de vídeo codificados y los datos de audio codificados utilizando un esquema predeterminado, modula y convierte los datos utilizando el modulador/demodulador (circuito modulador/demodulador) ex452 y el transmisor/receptor ex451, y transmite el resultado a través de la antena ex450.

30 Cuando se recibe un vídeo adjunto en un correo electrónico o en un chat, o un vídeo vinculado desde una página web, por ejemplo, para decodificar los datos multiplexados recibidos a través de la antena ex450, el multiplexor/demultiplexor ex453 demultiplexa los datos multiplexados para dividirlos en un flujo de bits de datos de vídeo y un flujo de bits de datos de audio, suministra los datos de vídeo codificados al procesador de señales de vídeo ex455 a través del bus síncrono ex470, y suministra los datos de audio codificados al procesador de señales de audio ex454 a través del bus síncrono ex470. El procesador de señal de vídeo ex455 decodifica la señal de vídeo utilizando un procedimiento de decodificación de cuadros en movimiento correspondiente al procedimiento de codificación de cuadros en movimiento descrito en la realización y variaciones anteriores, y el vídeo o una imagen fija incluida en el archivo de cuadros en movimiento vinculado se muestra en la pantalla ex458 a través del controlador de pantalla ex459. Más aun, el procesador de señal de audio ex454 decodifica la señal de audio y envía audio desde la unidad de salida de audio ex457. Tenga en cuenta que, dado que la retransmisión en tiempo real se está volviendo cada vez más popular, existen casos en los que la reproducción del audio puede ser socialmente inapropiada de acuerdo con el entorno del usuario. De acuerdo con lo anterior, como valor inicial, es preferente una configuración en la que solo se reproducen datos de vídeo, es decir, no se reproduce la señal de audio. El audio se puede sincronizar y reproducir solo cuando se recibe una entrada, como cuando el usuario hace clic en los datos de vídeo.

45 Aunque se utilizó el teléfono inteligente ex115 en el ejemplo anterior, son concebibles tres implementaciones: un terminal transceptor que incluye tanto un codificador como un decodificador; un terminal transmisor que incluye únicamente un codificador; y un terminal receptor que incluye únicamente un decodificador. Además, en la descripción del sistema de transmisión digital, se proporciona un ejemplo en el que se reciben o transmiten datos multiplexados obtenidos como resultado de la multiplexación de datos de vídeo con, por ejemplo, datos de audio, pero los datos multiplexados pueden ser datos de vídeo multiplexados con datos distintos de los datos de audio, como datos de texto relacionados con el vídeo. Más aún, es posible que se reciban o transmitan los propios datos de vídeo en lugar de los datos multiplexados.

55 Aunque se describe que el controlador principal ex460 que incluye una CPU controla los procedimientos de codificación o decodificación, los terminales a menudo incluyen GPU. De acuerdo con lo anterior, es aceptable una configuración en la que se procesa un área grande a la vez haciendo uso de la capacidad de rendimiento de la GPU a través de la memoria compartida por la CPU y la GPU o la memoria que incluye una dirección que se administra de manera que permite el uso común por parte de la CPU y la GPU. Esto hace posible acortar el tiempo de codificación, mantener la naturaleza de tiempo real de la transmisión y reducir el retardo. En particular, el procesamiento relacionado con la estimación de movimiento, el filtrado de desbloqueo, el desplazamiento adaptativo de muestra (SAO) y la transformación/cuantificación se pueden llevar a cabo de 60 manera efectiva por la GPU en lugar de la CPU en unidades de, por ejemplo, cuadros, todo a la vez.

Este aspecto se puede implementar en combinación con uno o más de los otros aspectos de acuerdo con la presente divulgación. Además, parte de los procedimientos en los diagramas de flujo, parte de los elementos constituyentes de los aparatos y parte de la sintaxis descrita en este aspecto se pueden implementar en combinación con otros aspectos.

5 **Aplicabilidad industrial**

La presente divulgación es aplicable, por ejemplo, a receptores de televisión, grabadoras de vídeo digitales, sistemas de navegación para automóviles, teléfonos móviles, cámaras digitales, cámaras de vídeo digitales, sistemas de teleconferencia, espejos electrónicos, etc.

Marcas de referencia en los dibujos

10	100	codificador
	102	divisor
	104	sustractor
	106	transformador
	108	cuantificador
15	110	codificador de entropía
	112, 204	cuantificador inverso
	114, 206	transformador inverso
	116, 208	sumador
	118, 210	memoria de bloques
20	120, 212	filtro de bucle
	122, 214	memoria de fotograma
	124, 216	predictor intra
	126, 218	predictor inter
	128, 220	controlador de predicción
25	160, 260	circuitería
	162, 262	memoria
	200	decodificador
	202	decodificador de entropía

REIVINDICACIONES

1. Un decodificador (200), que comprende:

circuitería (260); y

memoria (262), en la que

5 la circuitería (260) se configuran para, utilizando la memoria (262), en el procesamiento de predicción inter:

derivar un primer vector de movimiento de un primer bloque actual que se va a procesar, utilizando un vector de movimiento de un bloque previo que se ha procesado previamente;

derivar un segundo vector de movimiento del primer bloque actual al realizar una estimación de movimiento en la vecindad de una posición especificada por el primer vector de movimiento; y

10 generar una imagen de predicción del primer bloque actual al realizar una compensación de movimiento utilizando el segundo vector de movimiento,

en el que un modo del procesamiento de predicción inter es un modo de fusión,

caracterizado porque

15 la circuitería (262), en el procesamiento de predicción inter, se configura para derivar un tercer vector de movimiento de un segundo bloque actual que se procesará después del primer bloque actual, utilizando el primer vector de movimiento del primer bloque actual, cuando se incluye el segundo bloque actual en un cuadro que incluye el primer bloque actual, y

20 derivar el tercer vector de movimiento del segundo bloque actual, utilizando el segundo vector de movimiento del primer bloque actual, cuando se incluye el segundo bloque actual en un cuadro diferente del cuadro que incluye el primer bloque actual,

derivar un cuarto vector de movimiento del segundo bloque actual al realizar una estimación de movimiento en la vecindad de una posición especificada por el tercer vector de movimiento; y

generar una imagen de predicción del segundo bloque actual al realizar una compensación de movimiento utilizando el cuarto vector de movimiento.

25 2. El decodificador (200) de acuerdo con la reivindicación 1, en el que

al derivar el tercer vector de movimiento, la circuitería (260) se configura para:

(i) generar una lista de predictores de vector de movimiento que indica una pluralidad de predictores de vector de movimiento utilizando el primer vector de movimiento o el segundo vector de movimiento del primer bloque actual, y

30 (ii) determinar el tercer vector de movimiento del segundo bloque actual en base a la pluralidad de predictores de vector de movimiento.

3. Un procedimiento de decodificación, que comprende:

derivar (S203) un primer vector de movimiento de un primer bloque actual que se va a procesar, utilizando un vector de movimiento de un bloque previo que se ha procesado previamente;

35 derivar (S204) un segundo vector de movimiento del primer bloque actual al realizar una estimación de movimiento en la vecindad de una posición especificada por el primer vector de movimiento; y

generar (S208) una imagen de predicción del primer bloque actual al realizar una compensación de movimiento utilizando el segundo vector de movimiento,

en el que un modo del procesamiento de predicción inter es un modo de fusión,

40 **caracterizado porque**

el procedimiento de decodificación comprende además:

derivar un tercer vector de movimiento de un segundo bloque actual que se procesará después del primer bloque actual, utilizando el primer vector de movimiento del primer bloque actual, cuando se incluye el segundo bloque actual en un cuadro que incluye el primer bloque actual, y

derivar el tercer vector de movimiento del segundo bloque actual, utilizando el segundo vector de movimiento del primer bloque actual, cuando se incluye el segundo bloque actual en un cuadro diferente del cuadro que incluye el primer bloque actual,

5

derivar un cuarto vector de movimiento del segundo bloque actual al realizar una estimación de movimiento en la vecindad de una posición especificada por el tercer vector de movimiento; y

generar una imagen de predicción del segundo bloque actual al realizar una compensación de movimiento utilizando el cuarto vector de movimiento.

FIG. 1

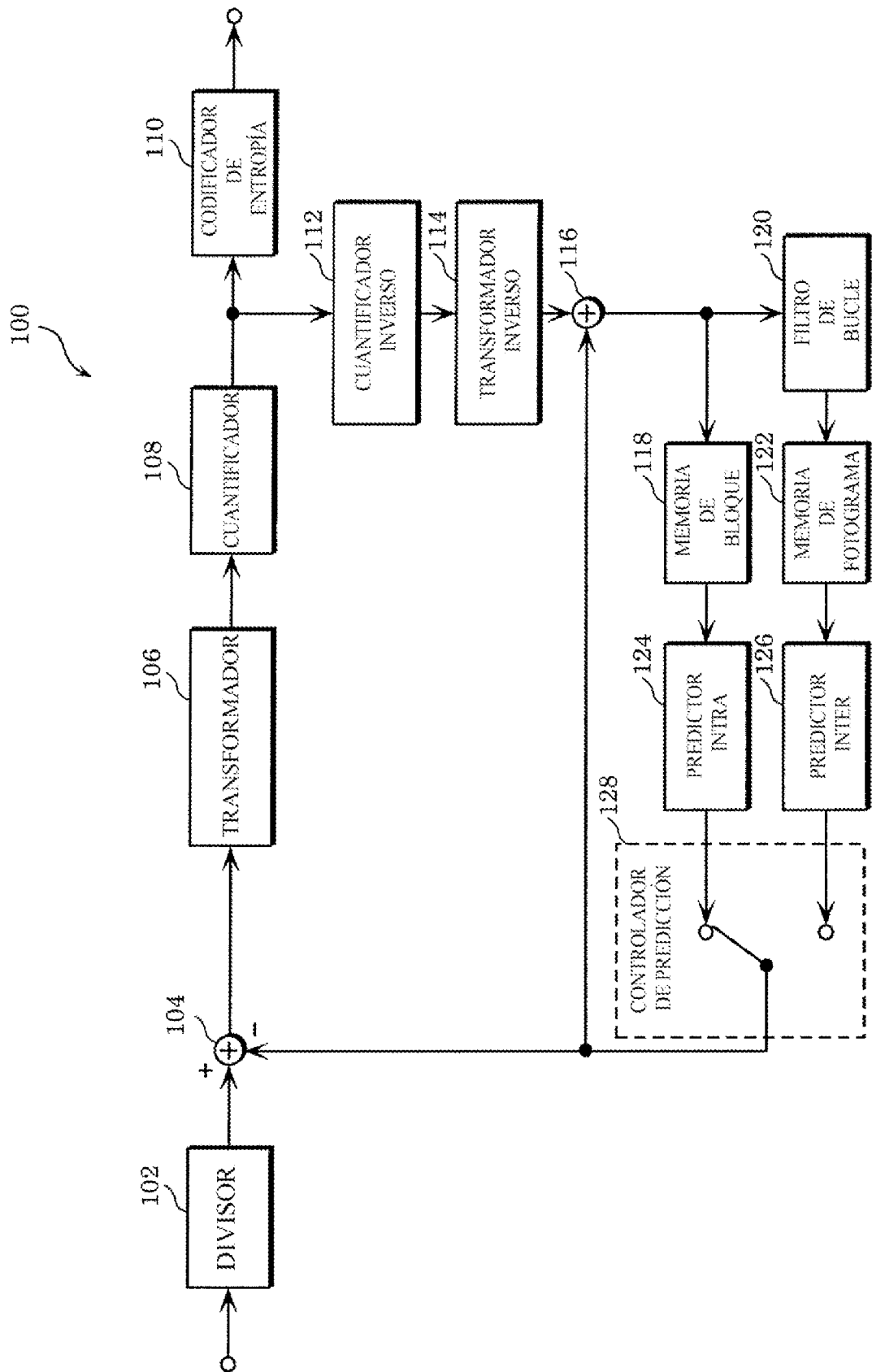


FIG. 2

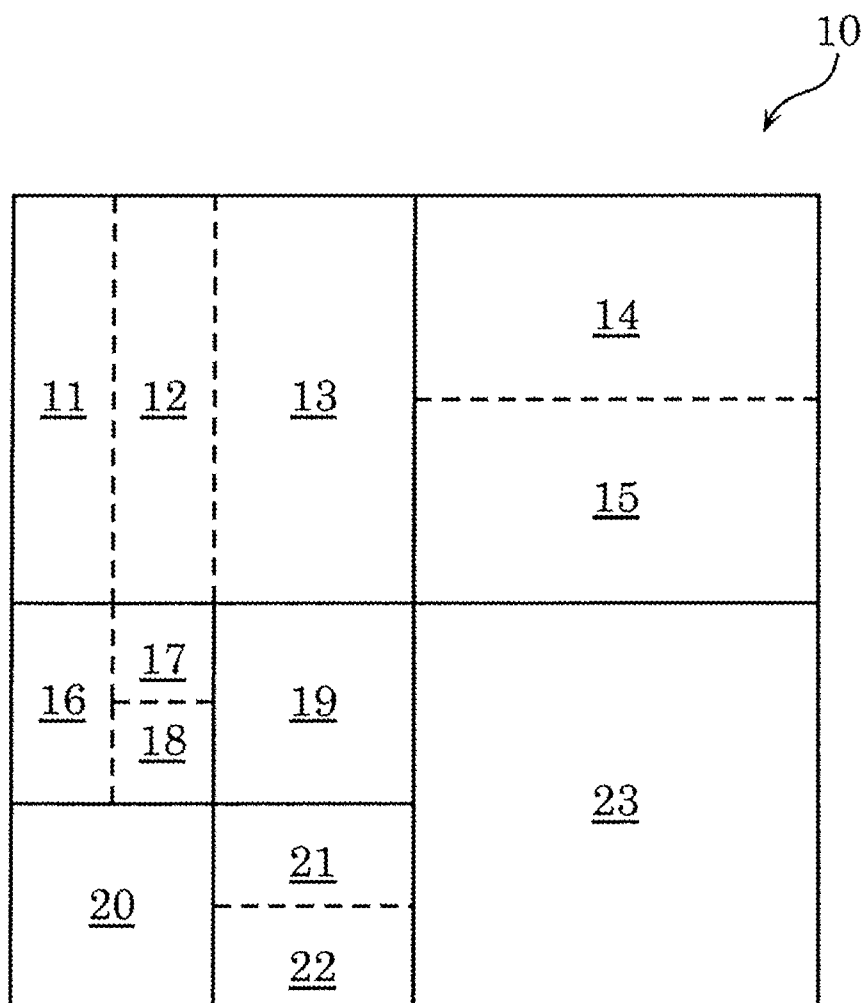


FIG. 3

TIPOS DE TRANSFORMACIÓN	FUNCIÓN BASE $T_i(j)$, $i, j = 0, 1, \dots, N - 1$
DCT - II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos \left(\frac{\pi \cdot i \cdot (2j + 1)}{2N} \right)$ $\text{DONDE } \omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT - V	$T_i(j) = \omega_0 \cdot \omega_1 \cdot \sqrt{\frac{2}{2N - 1}} \cdot \cos \left(\frac{2\pi \cdot i \cdot j}{2N - 1} \right)$ $\text{DONDE } \omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}, \omega_1 = \begin{cases} \sqrt{\frac{2}{N}} & j = 0 \\ 1 & j \neq 0 \end{cases}$
DCT - VIII	$T_i(j) = \sqrt{\frac{4}{2N + 1}} \cdot \cos \left(\frac{\pi \cdot (2i + 1) \cdot (2j + 1)}{4N + 2} \right)$
DST - I	$T_i(j) = \sqrt{\frac{2}{N + 1}} \cdot \text{sen} \left(\frac{\pi \cdot (i + 1) \cdot (j + 1)}{N + 1} \right)$
DST - VII	$T_i(j) = \sqrt{\frac{4}{2N + 1}} \cdot \text{sen} \left(\frac{\pi \cdot (2i + 1) \cdot (j + 1)}{2N + 1} \right)$

FIG. 4A

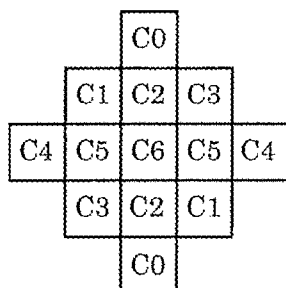


FIG. 4B

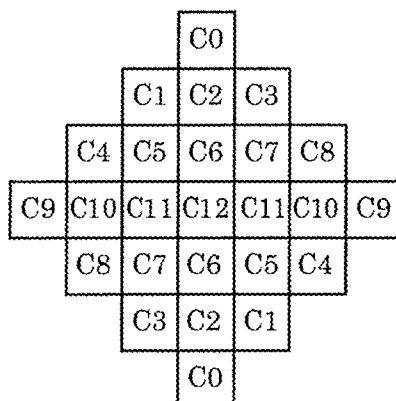


FIG. 4C

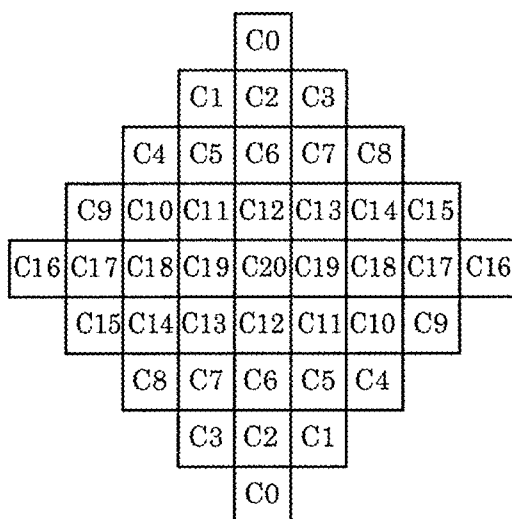


FIG. 5A

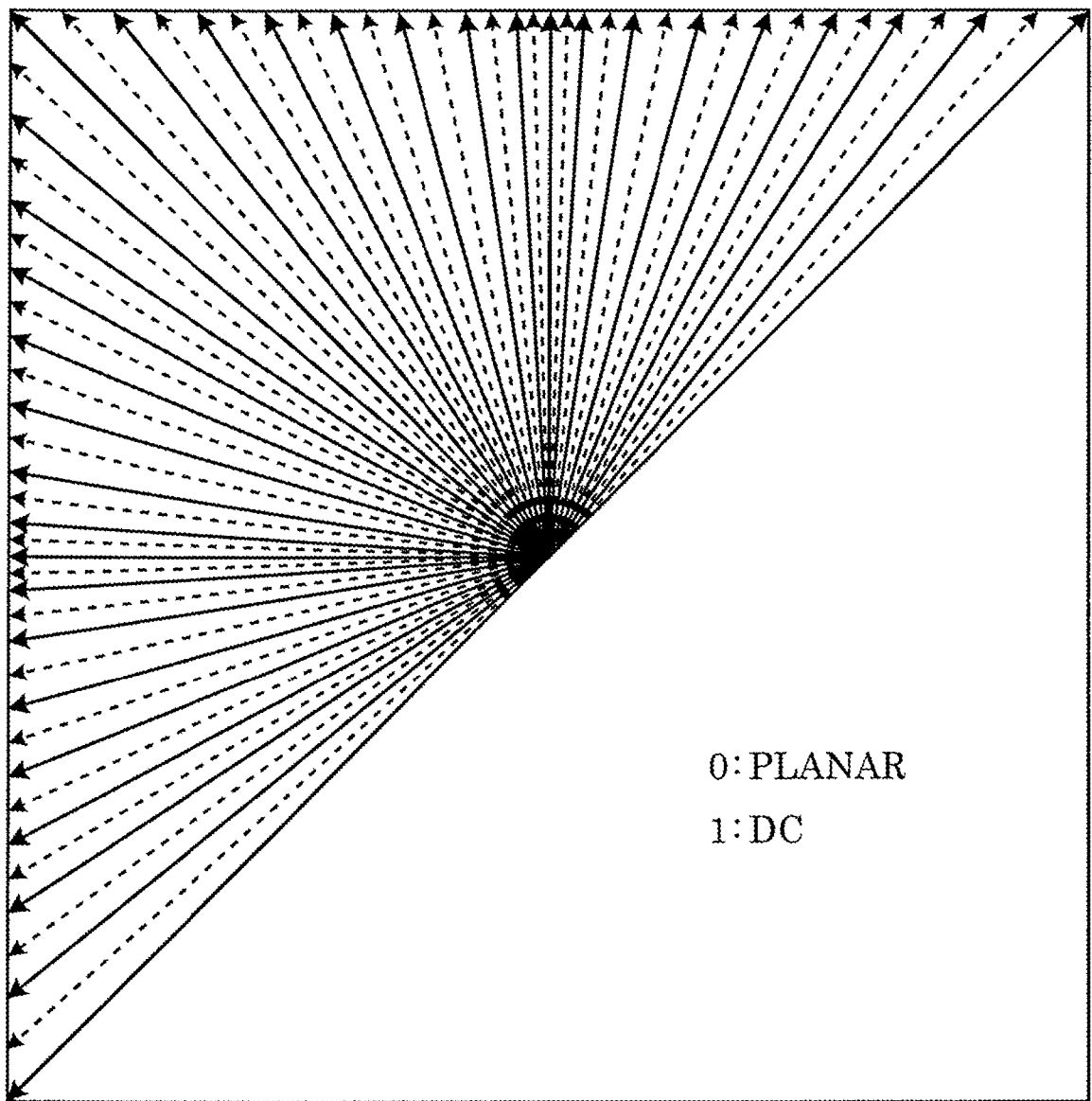


FIG. 5B

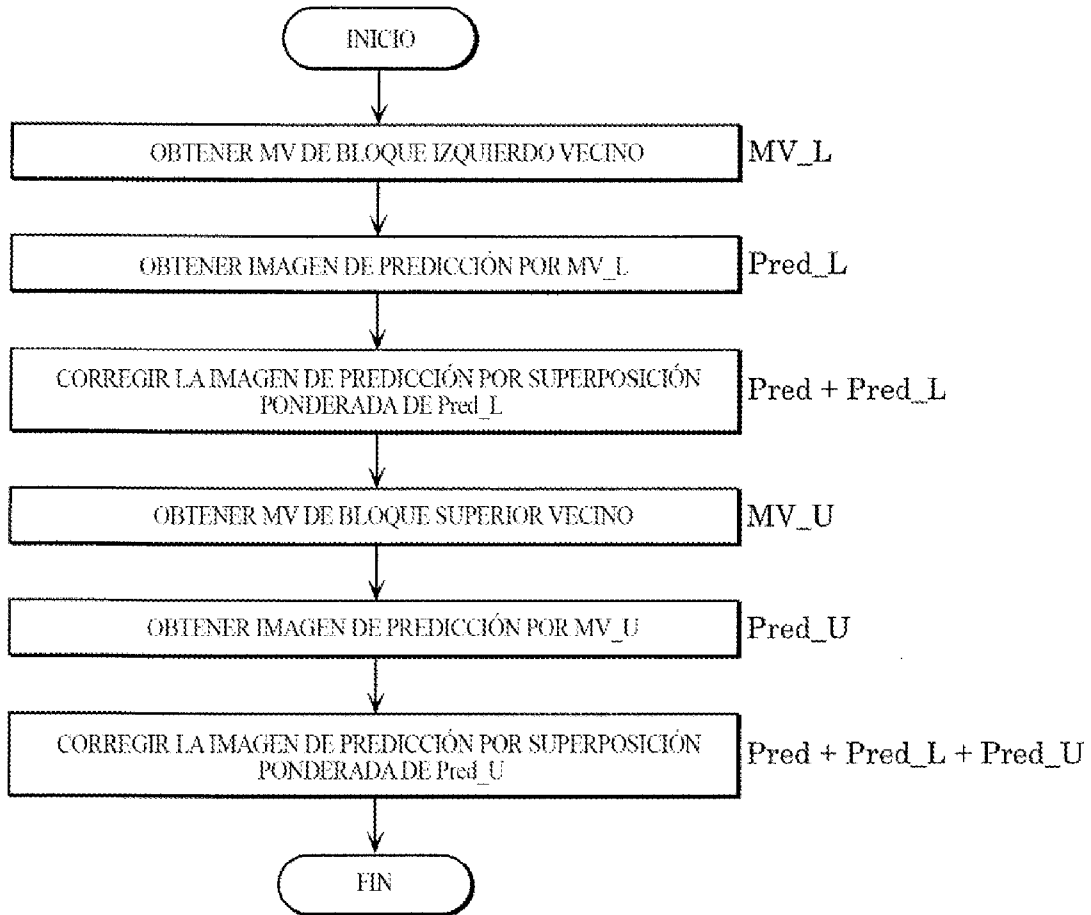


FIG. 5C

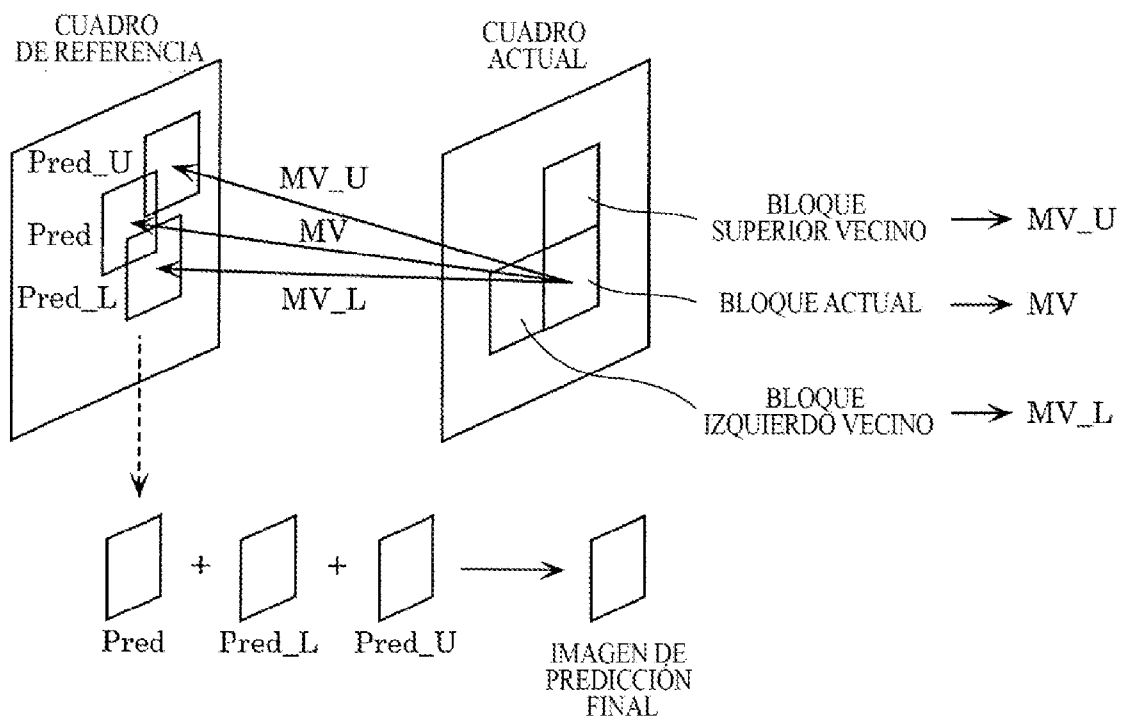


FIG. 5D

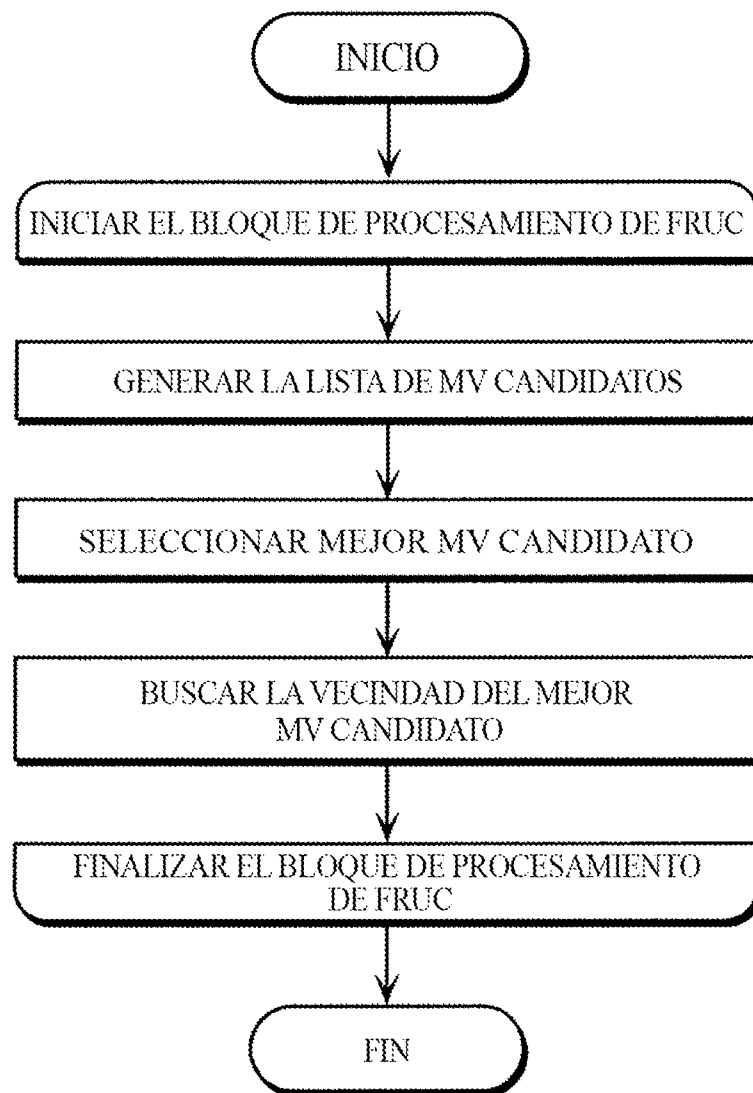


FIG. 6

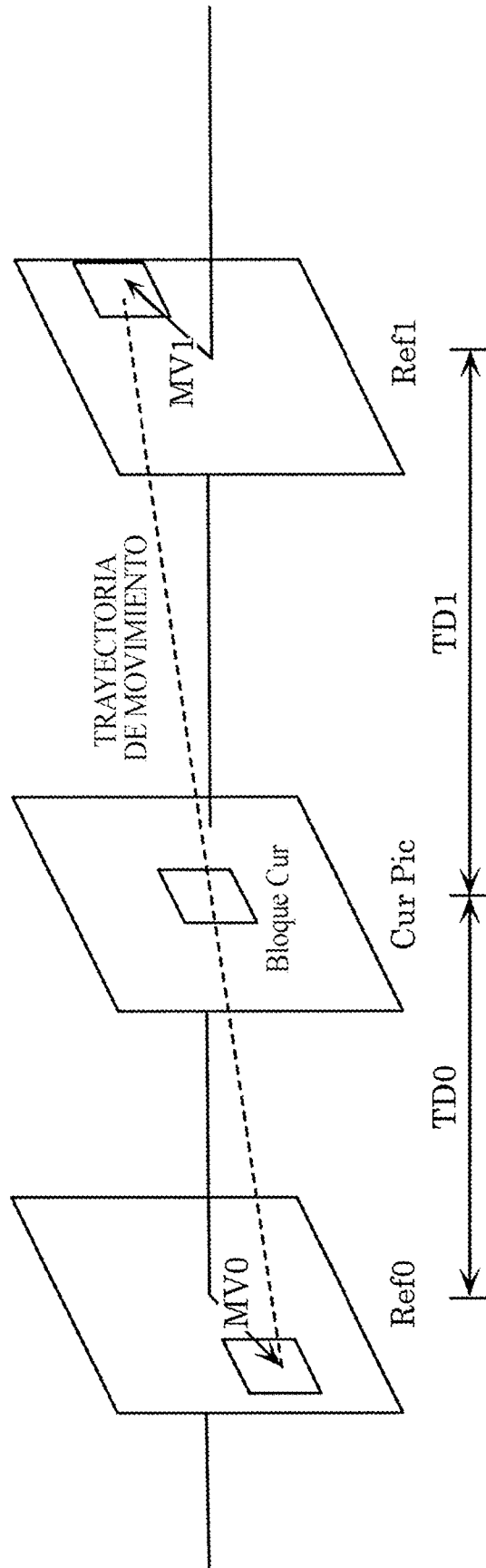


FIG. 7

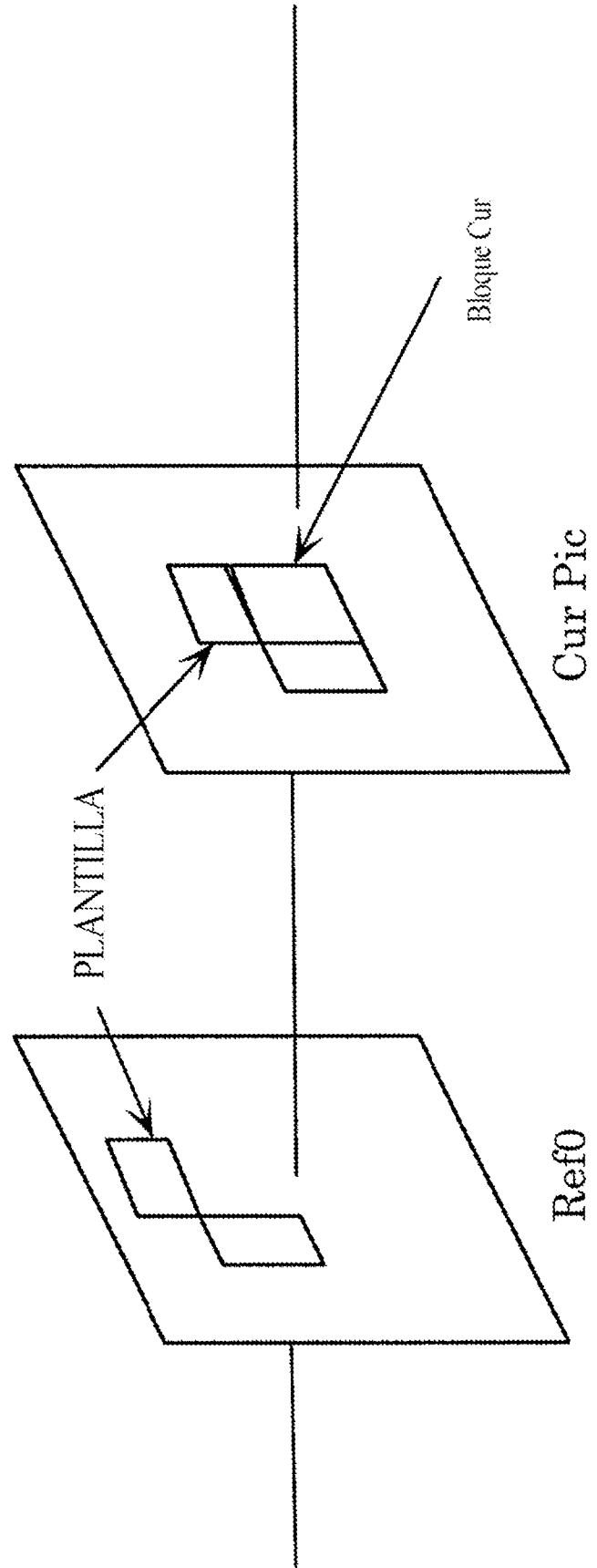


FIG. 8

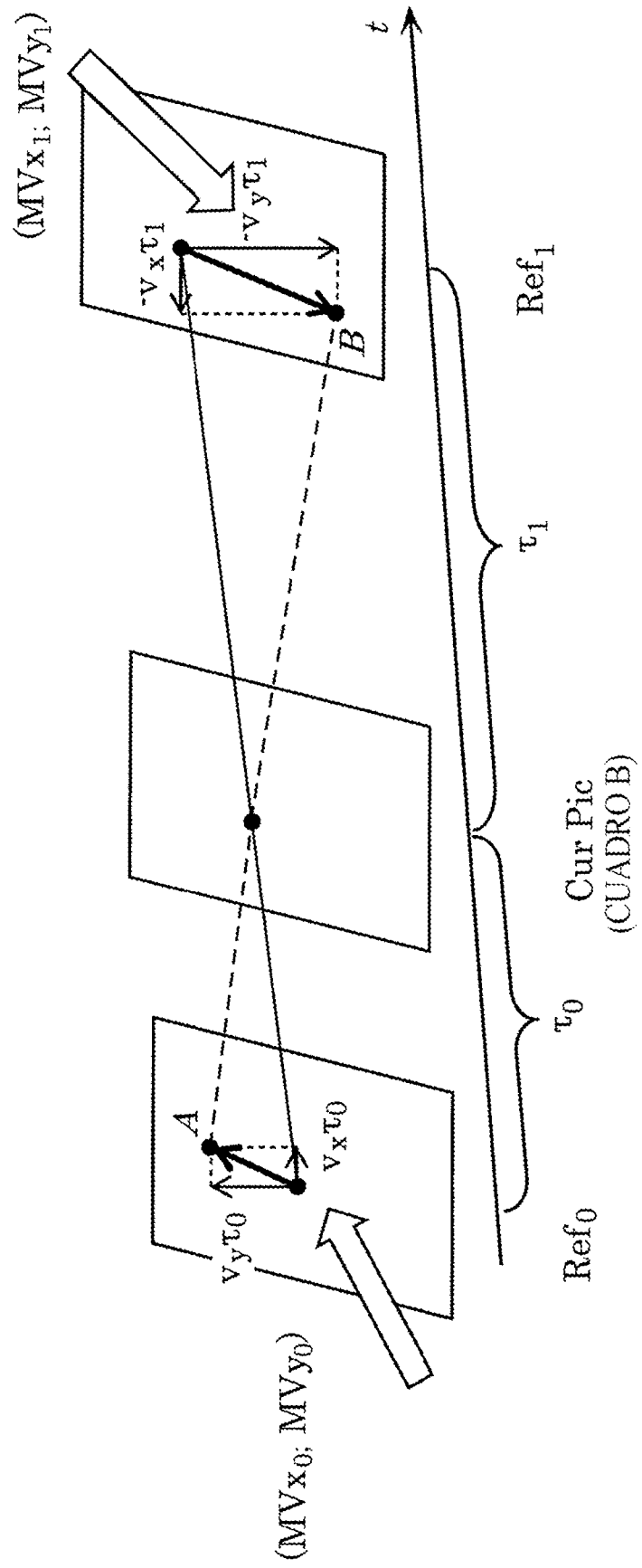


FIG. 9A

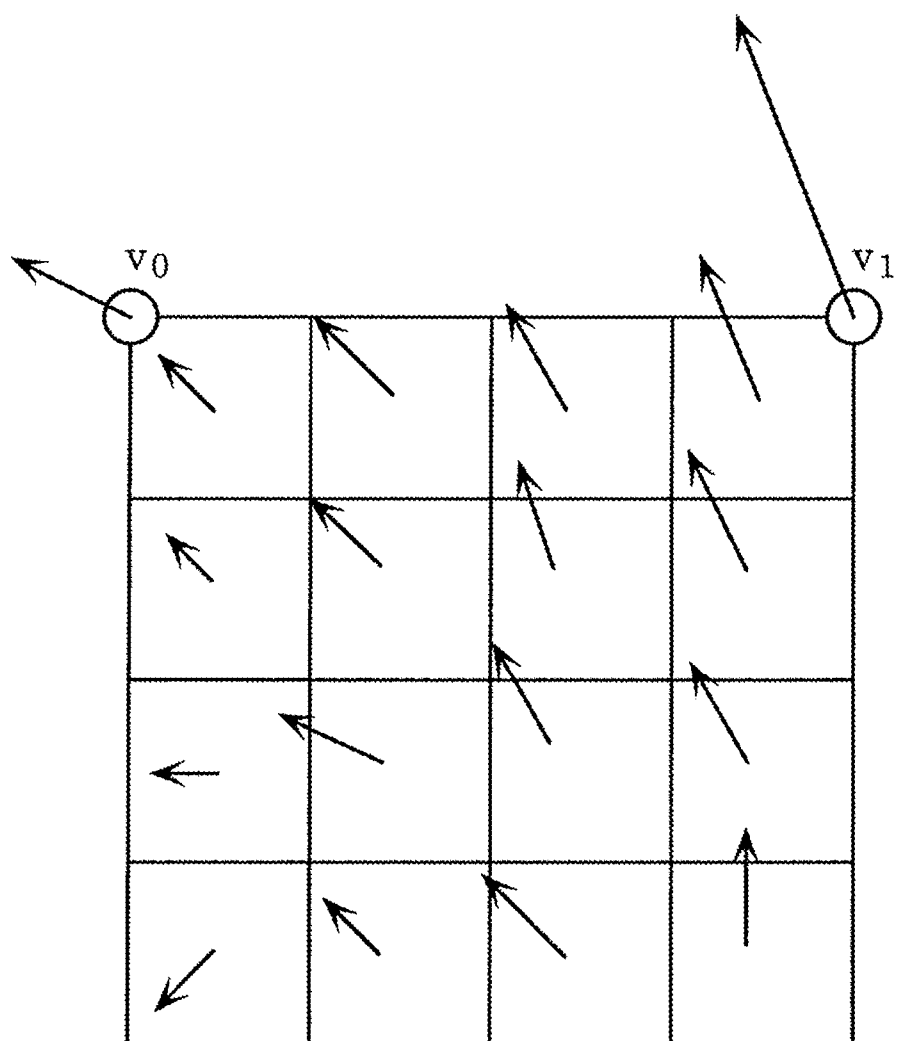


FIG. 9B

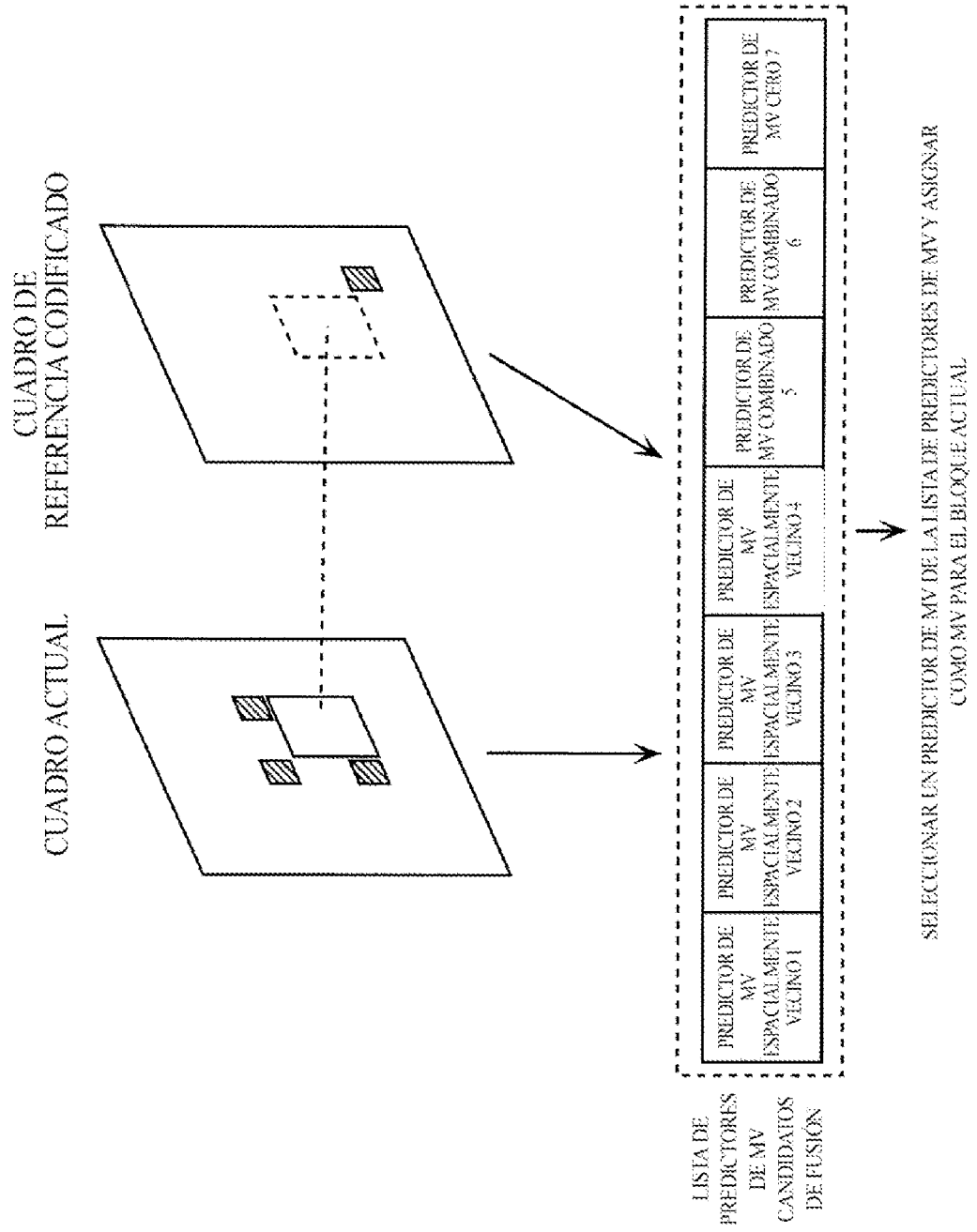


FIG. 9C

PLANTILLA GENERADA EN BASE A PIXEL DE REFERENCIA DE MV CANDIDATO (L0) Y PIXEL
DE REFERENCIA DE MV CANDIDATO (L1)

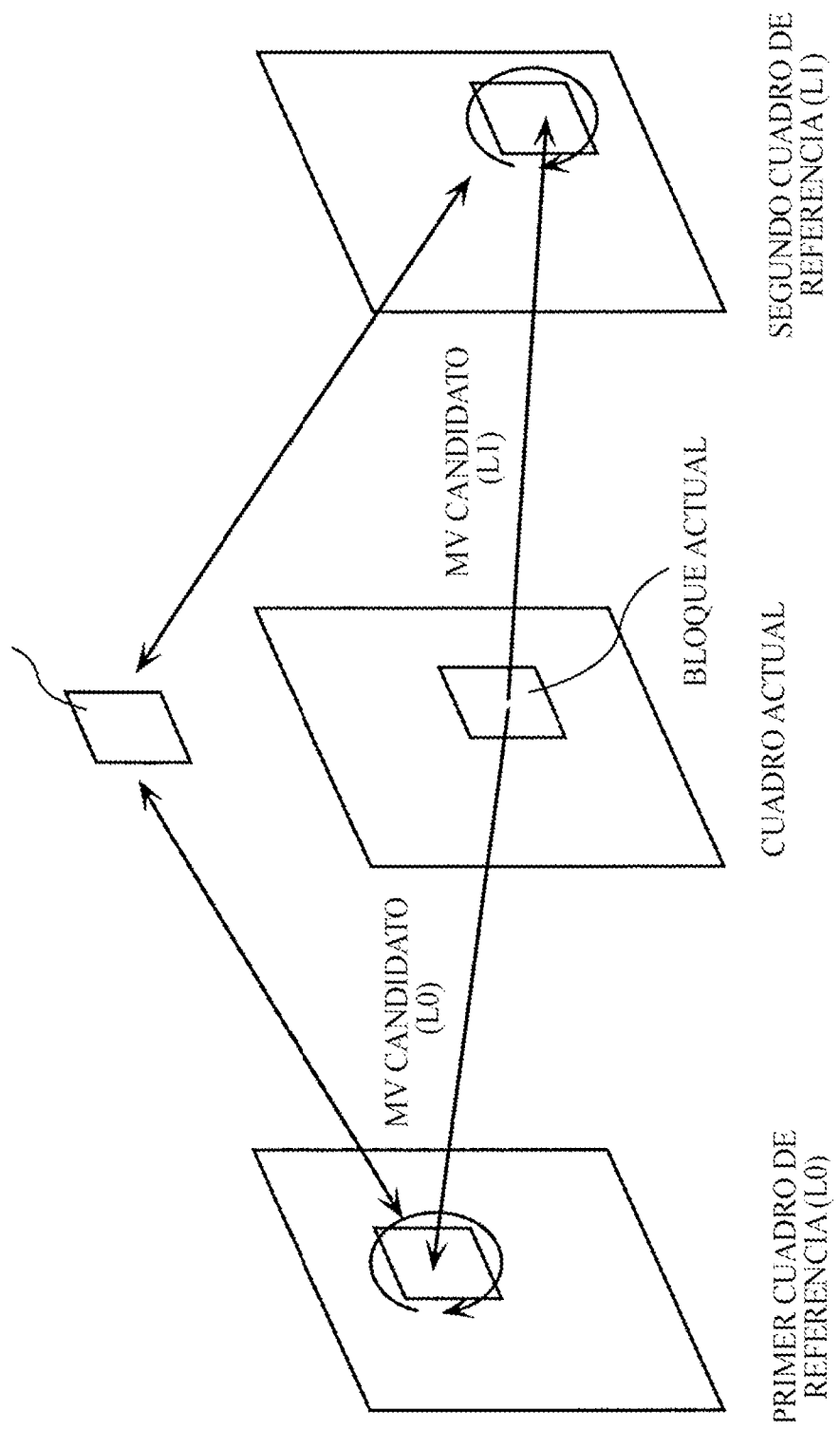


FIG. 9D

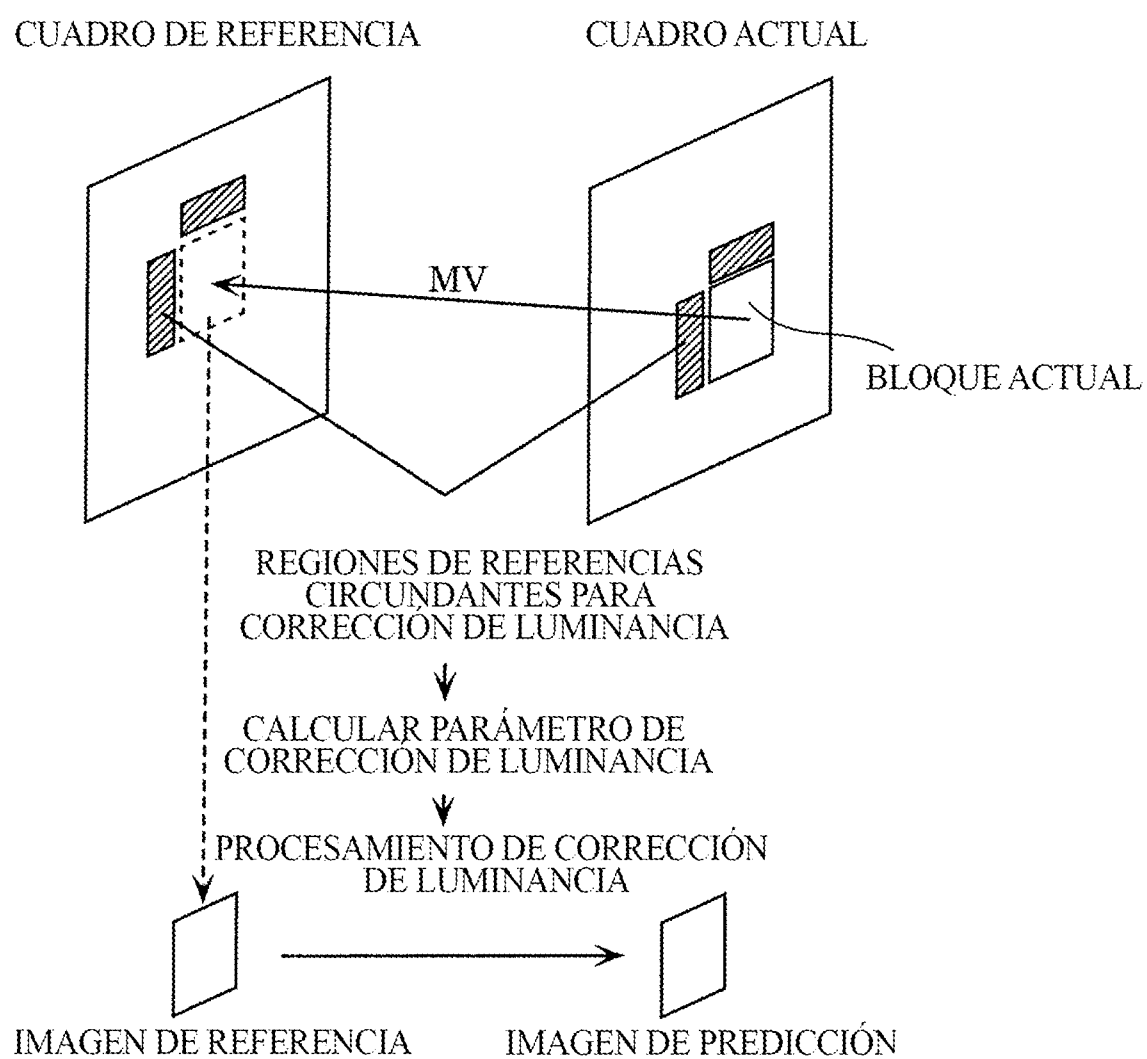


FIG. 10

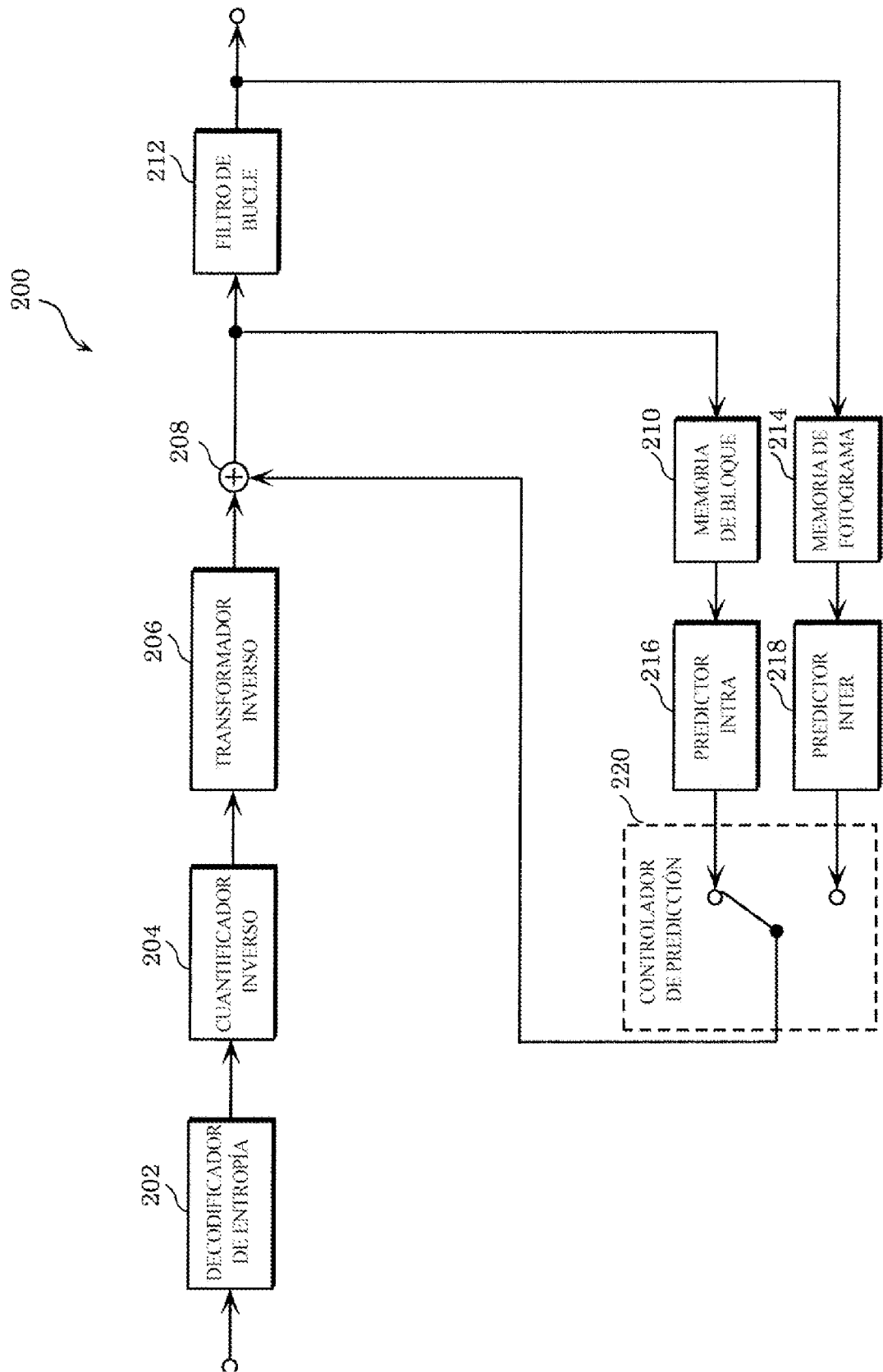


FIG. 11

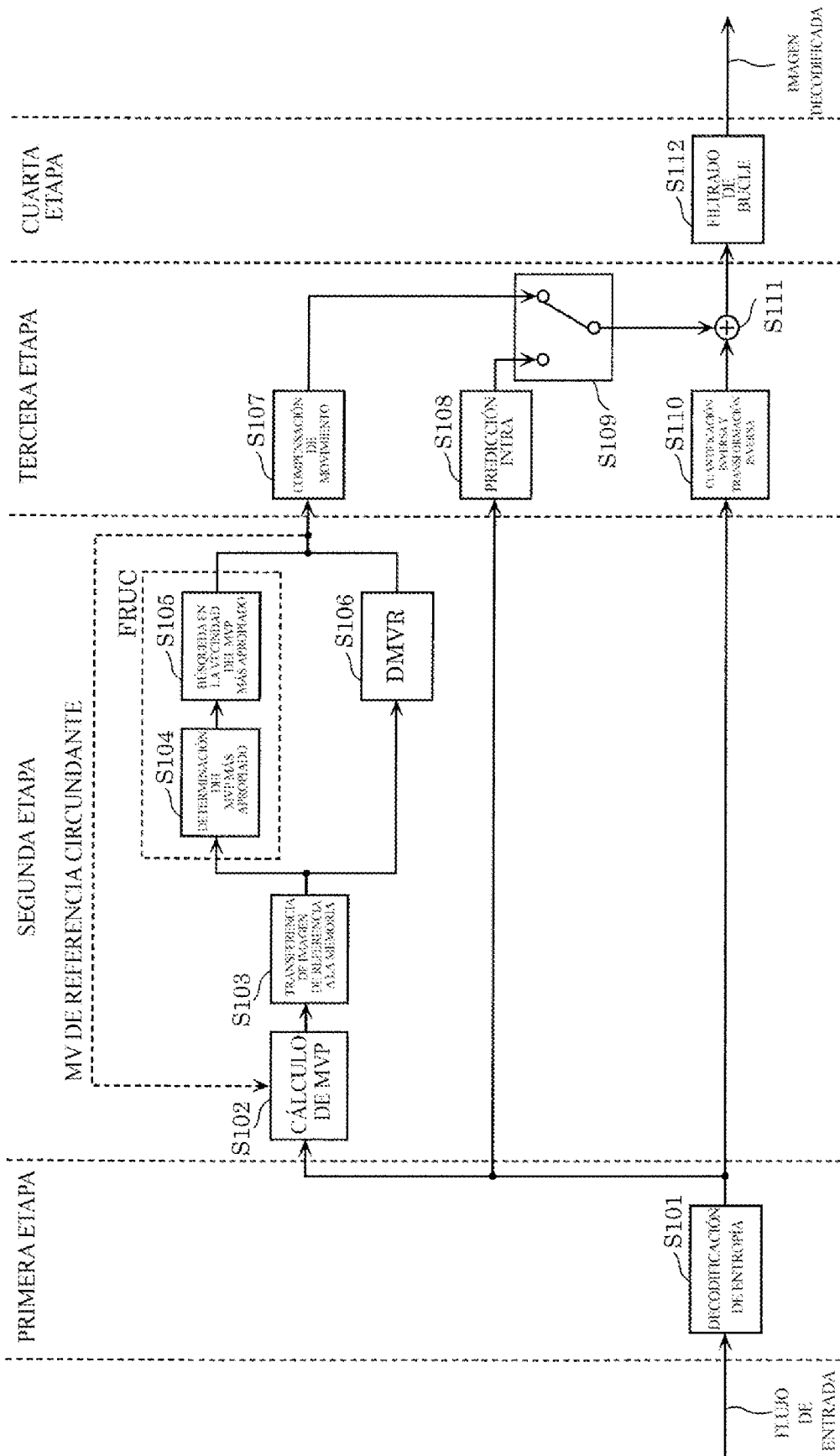


FIG. 12

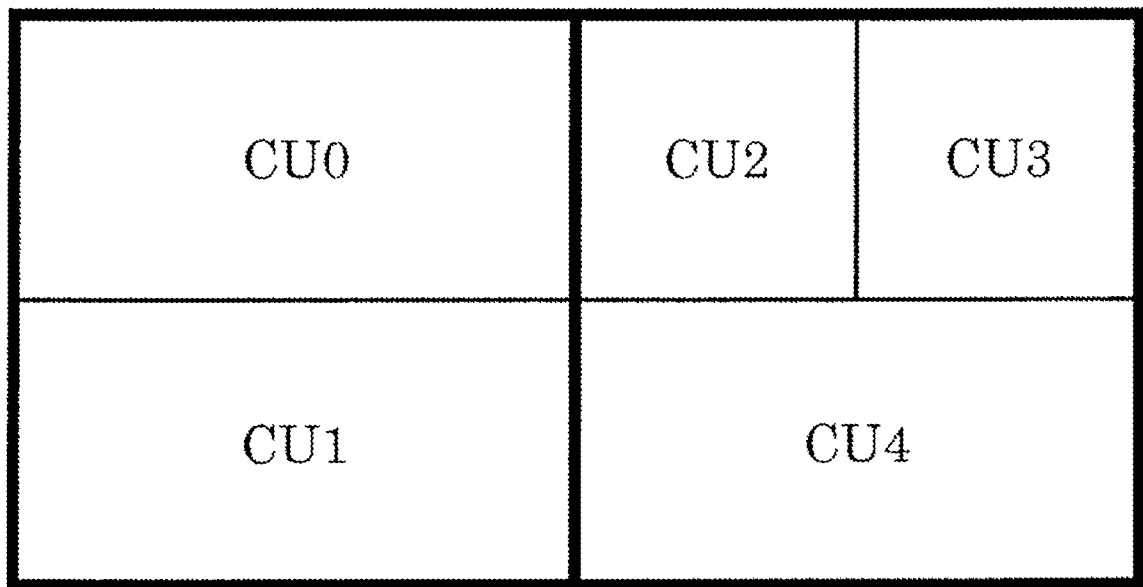


FIG. 13

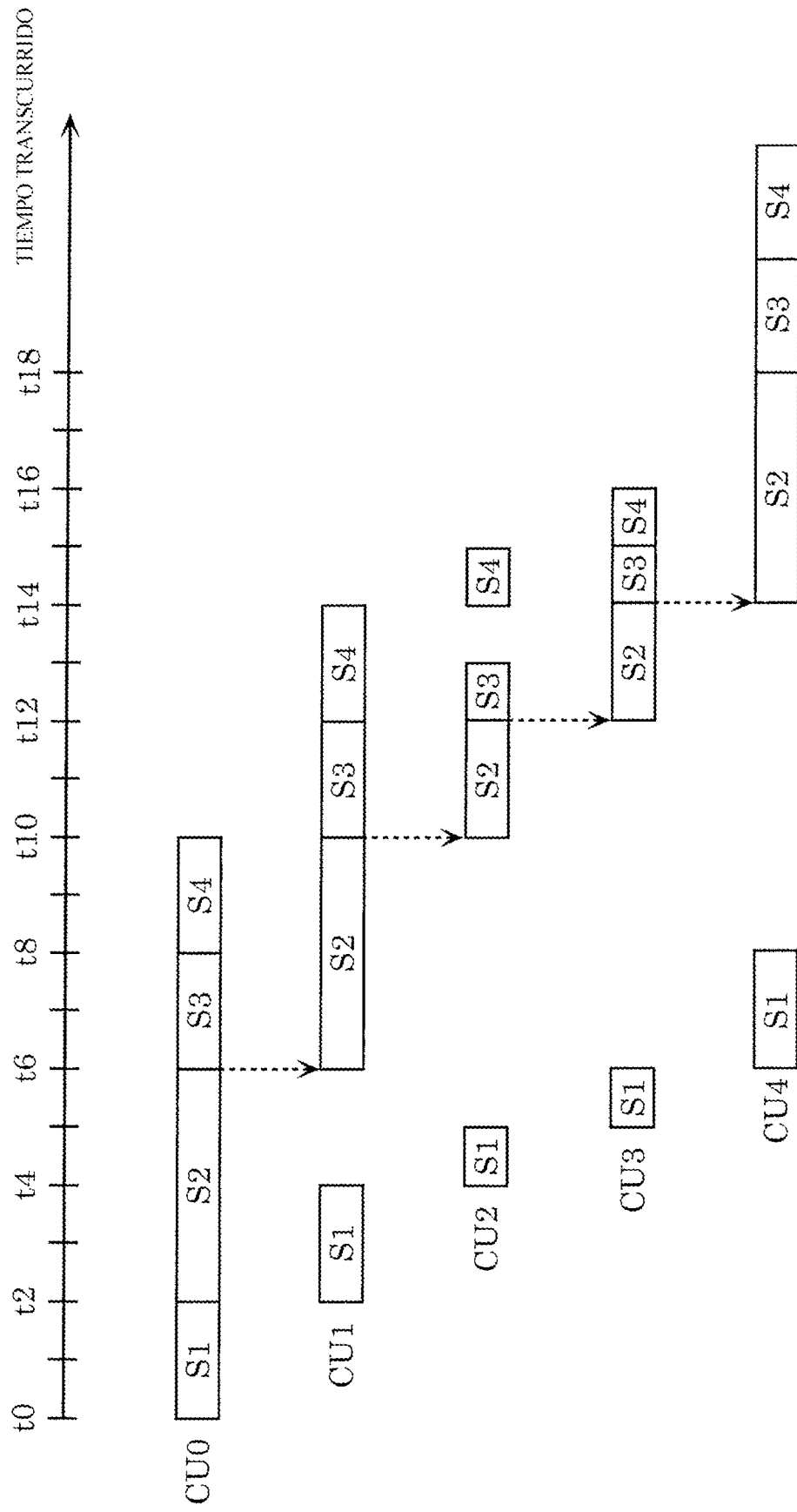


FIG. 14

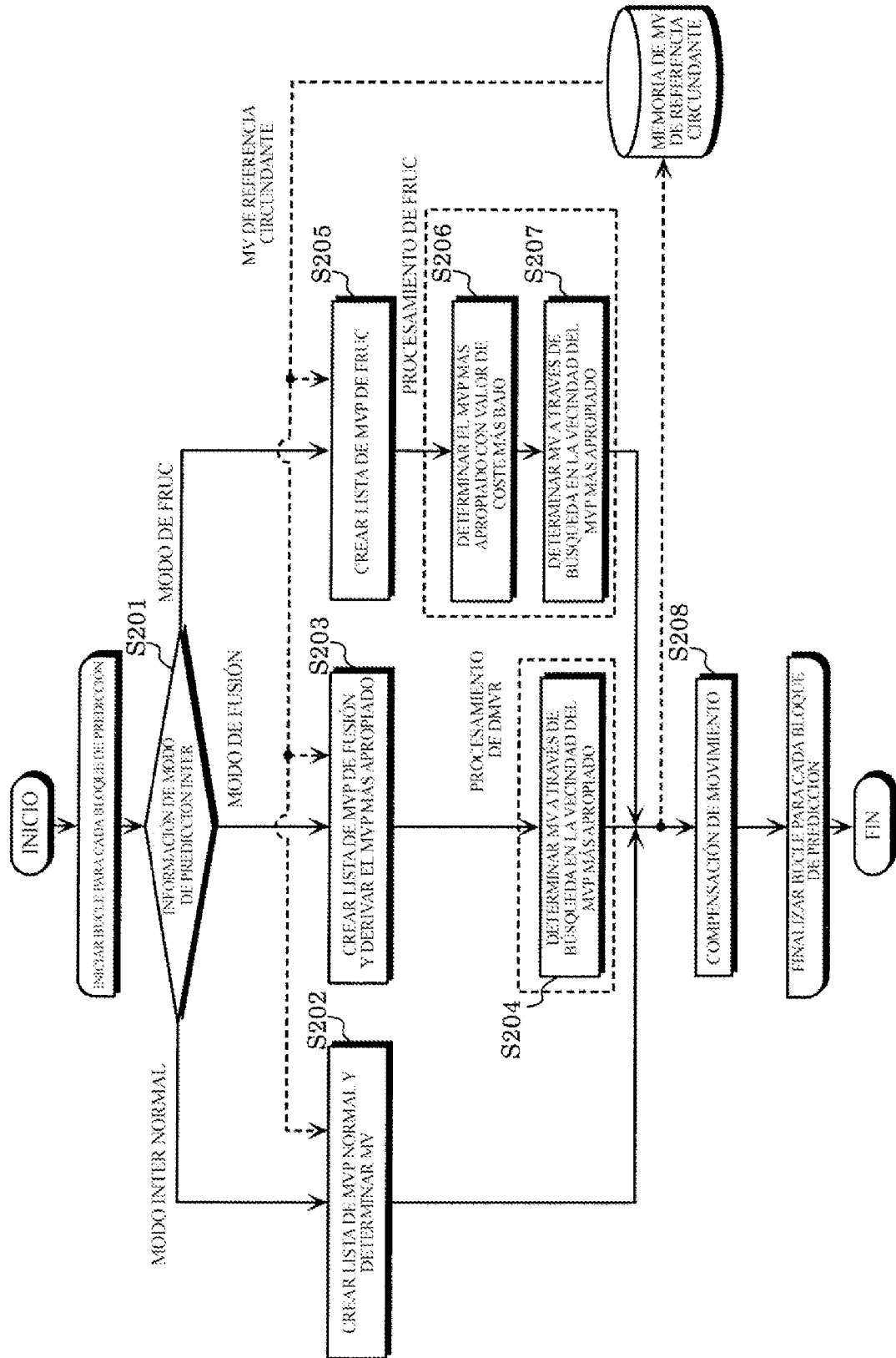


FIG. 15

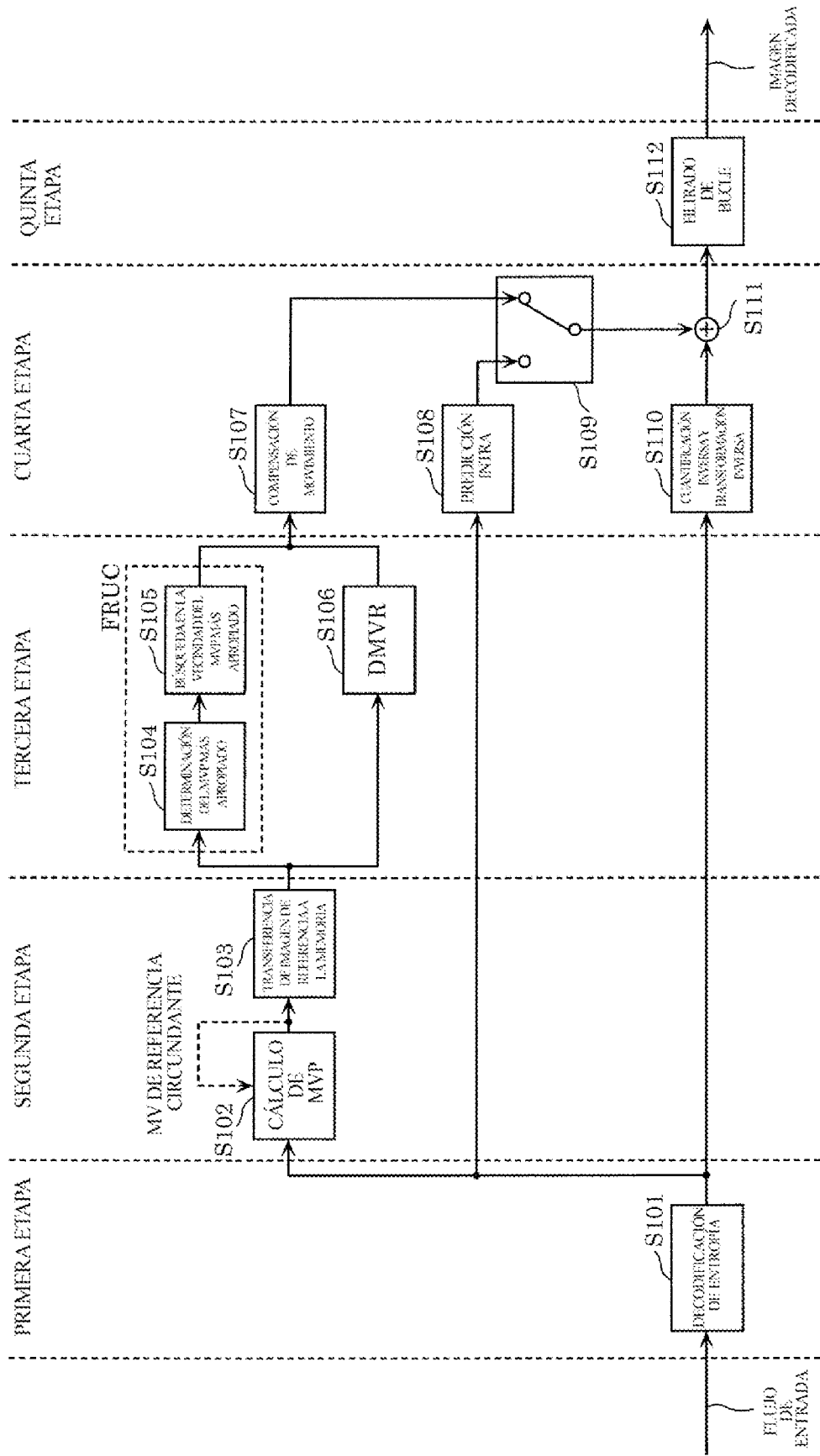


FIG. 16

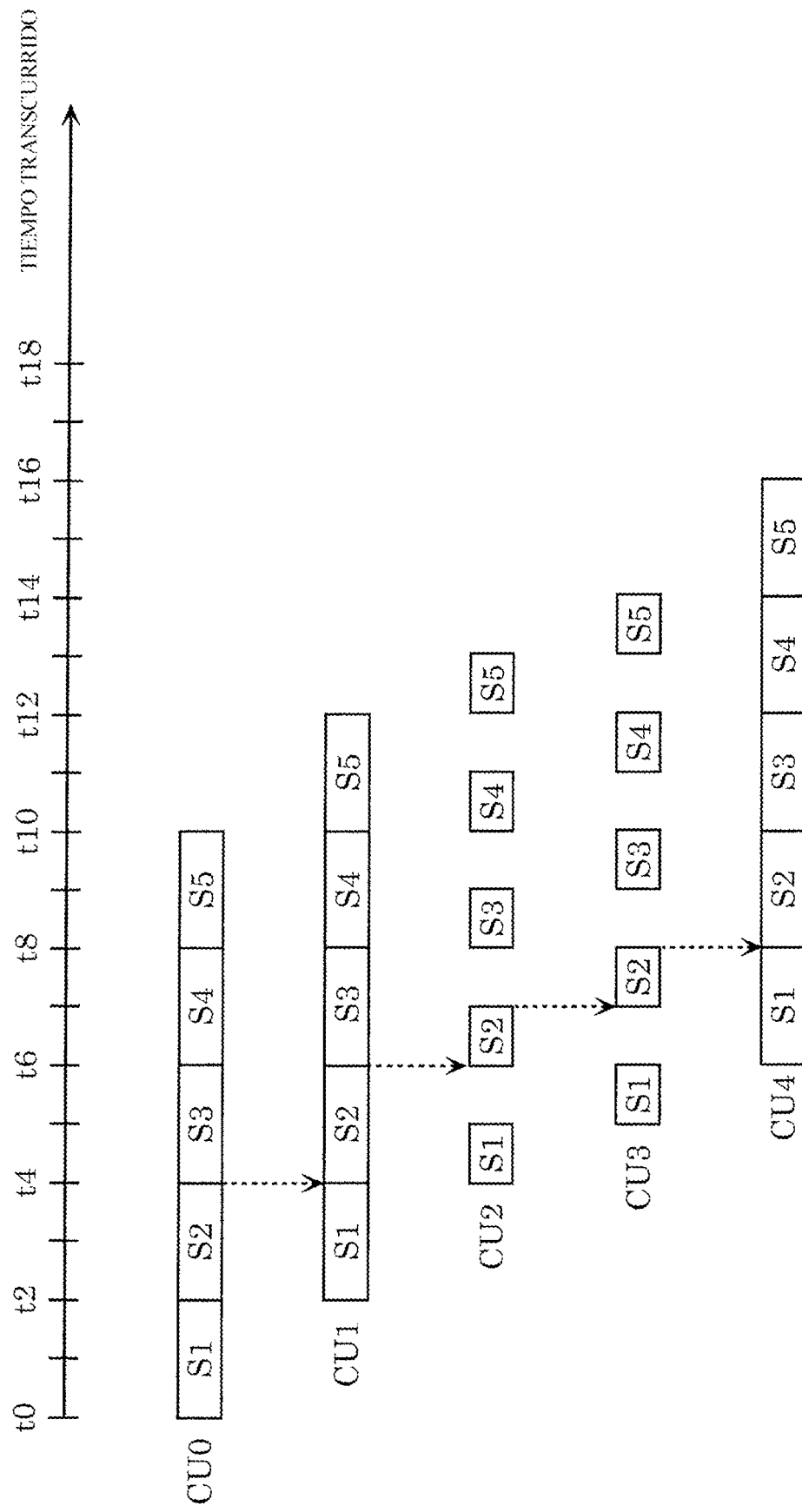


FIG. 17

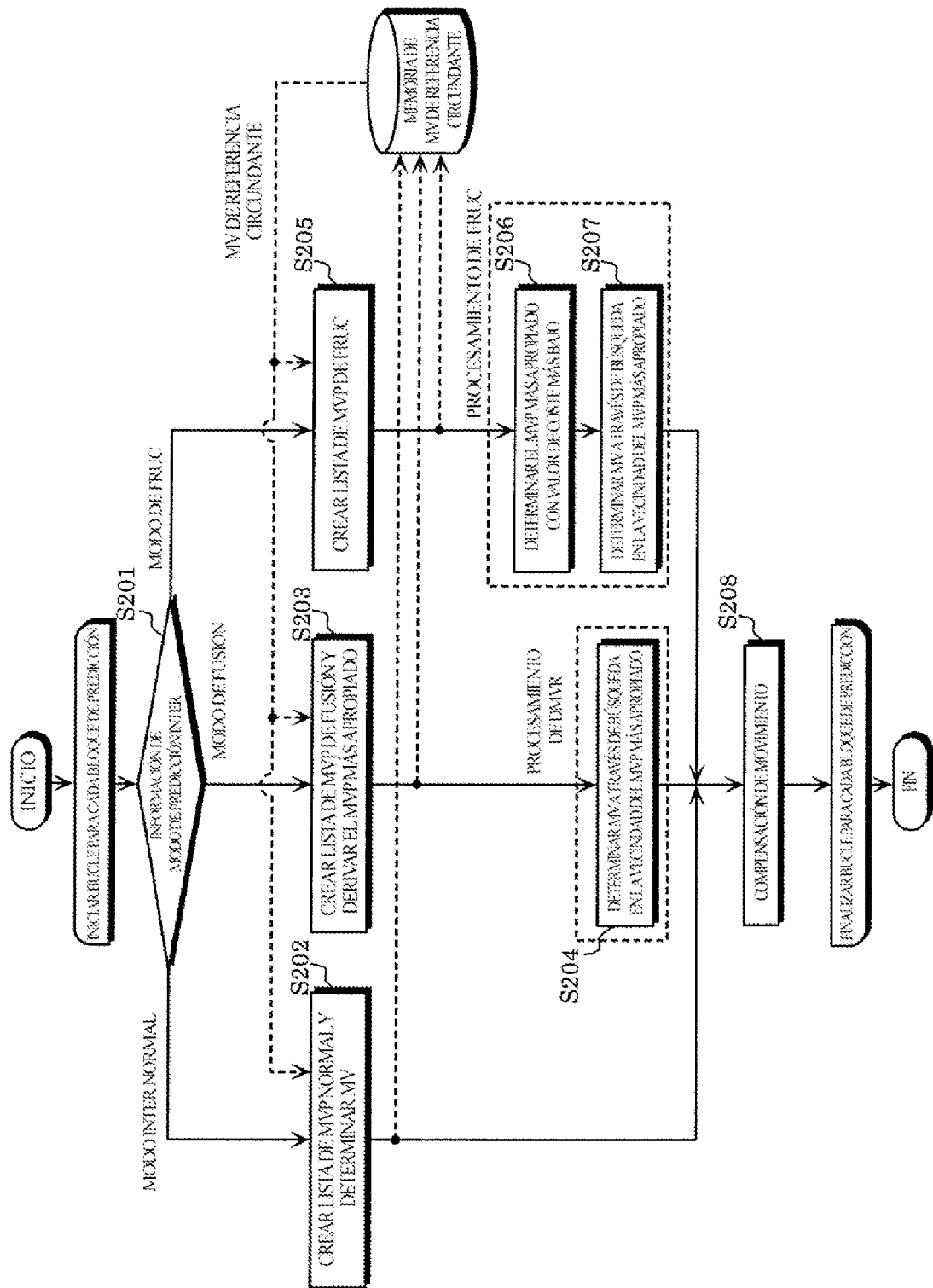


FIG. 18

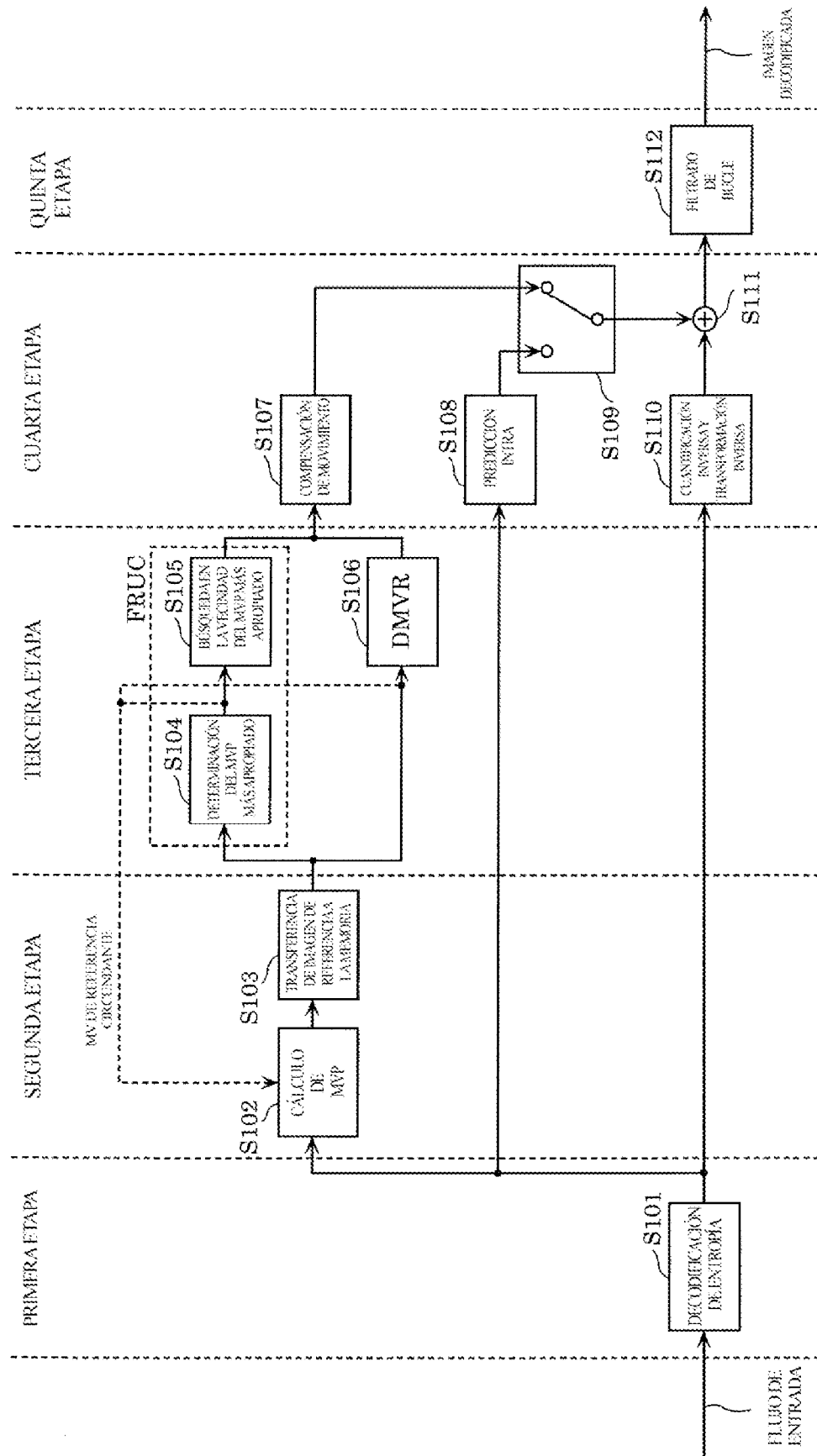


FIG. 19

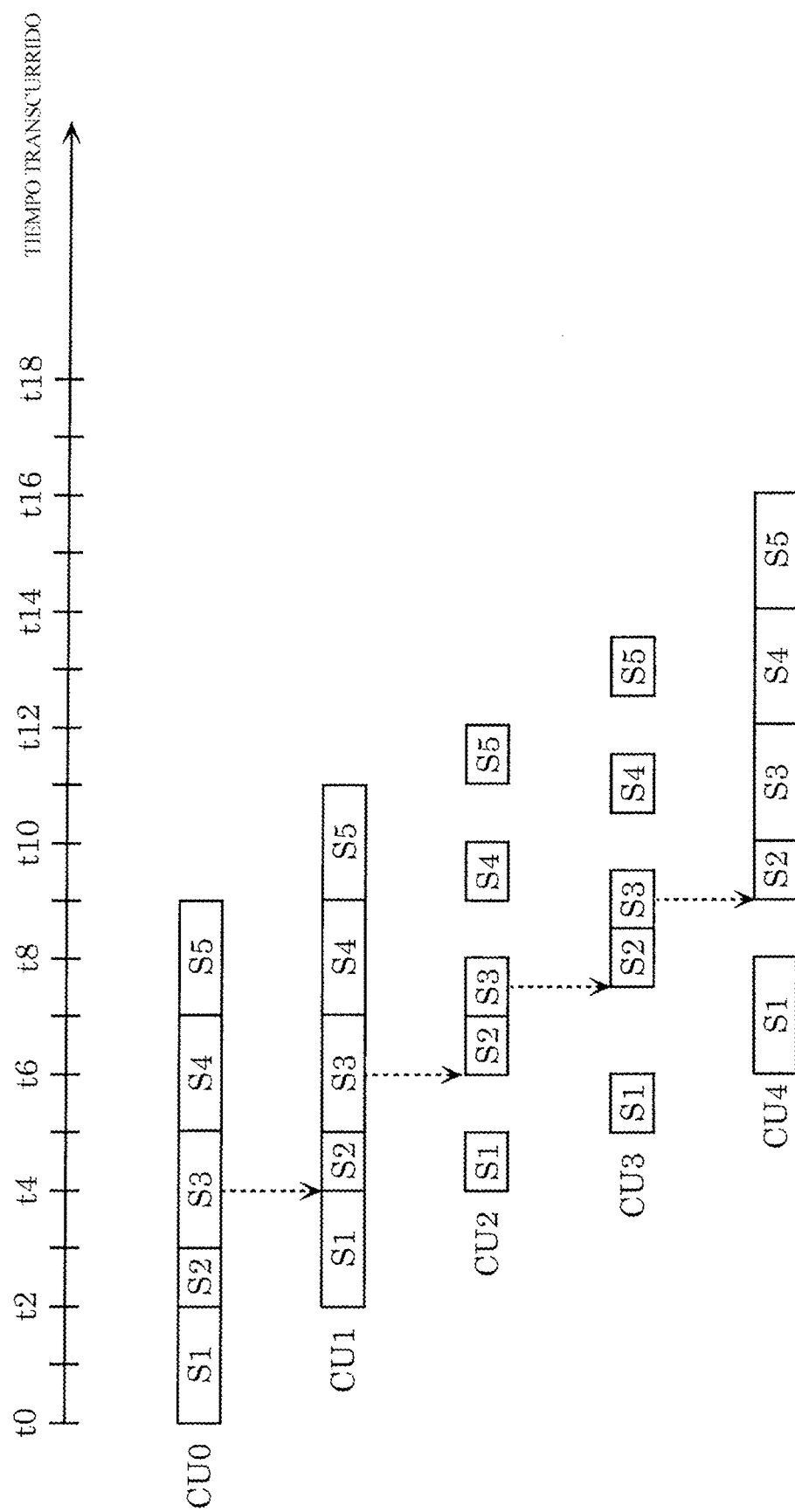


FIG. 20

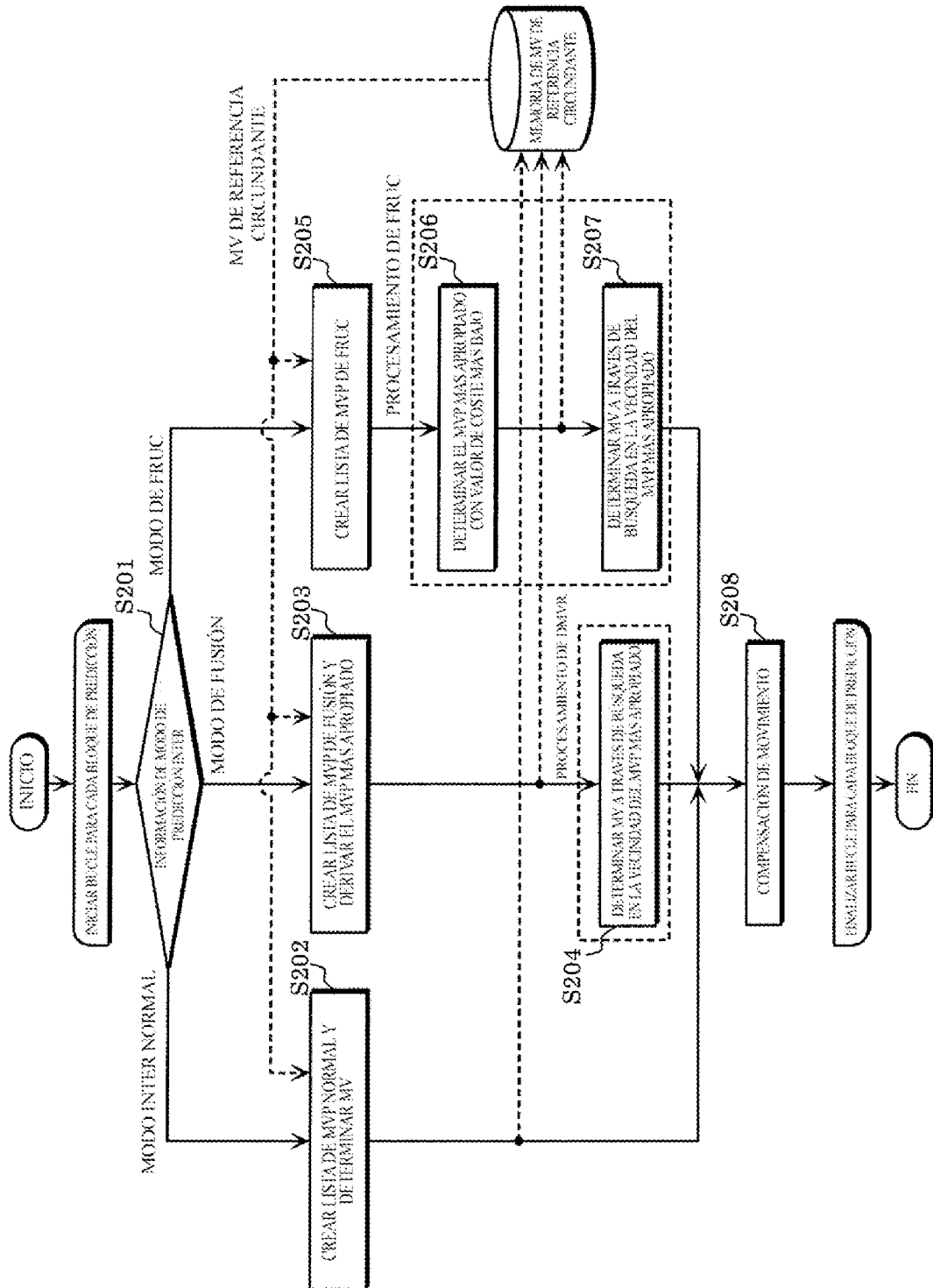


FIG. 21

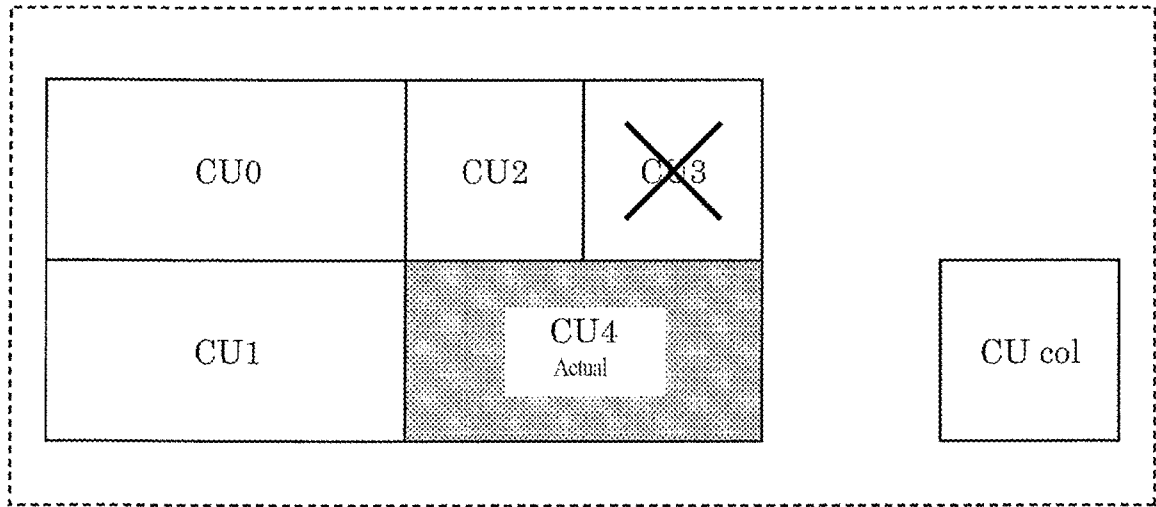


FIG. 22

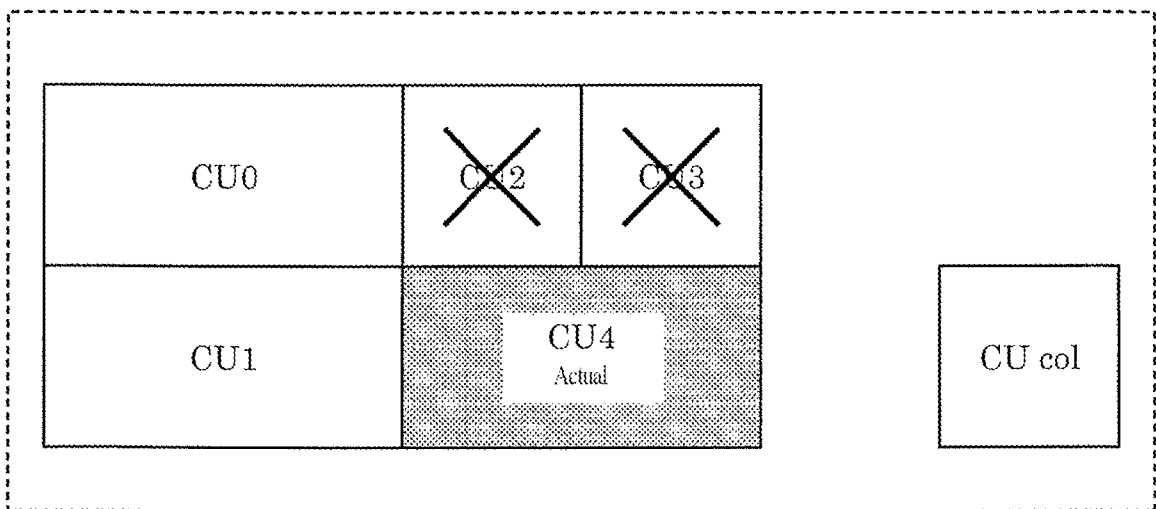


FIG. 23

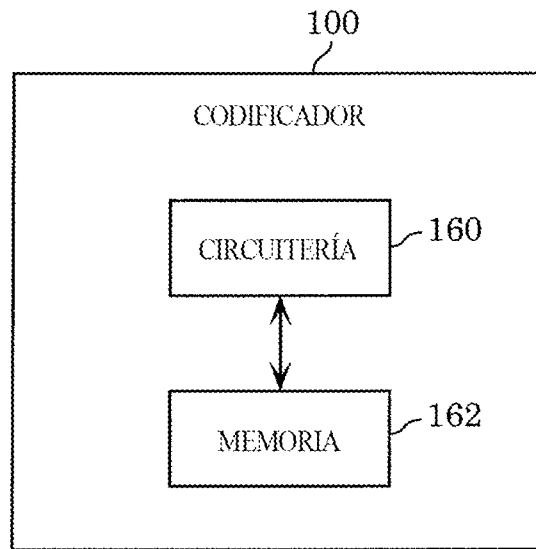


FIG. 24

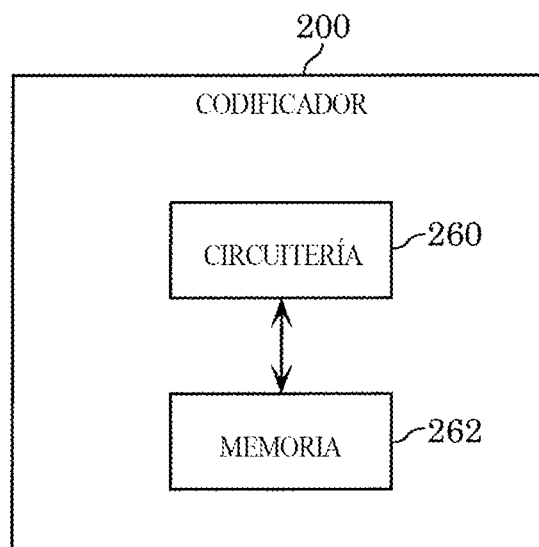


FIG. 25

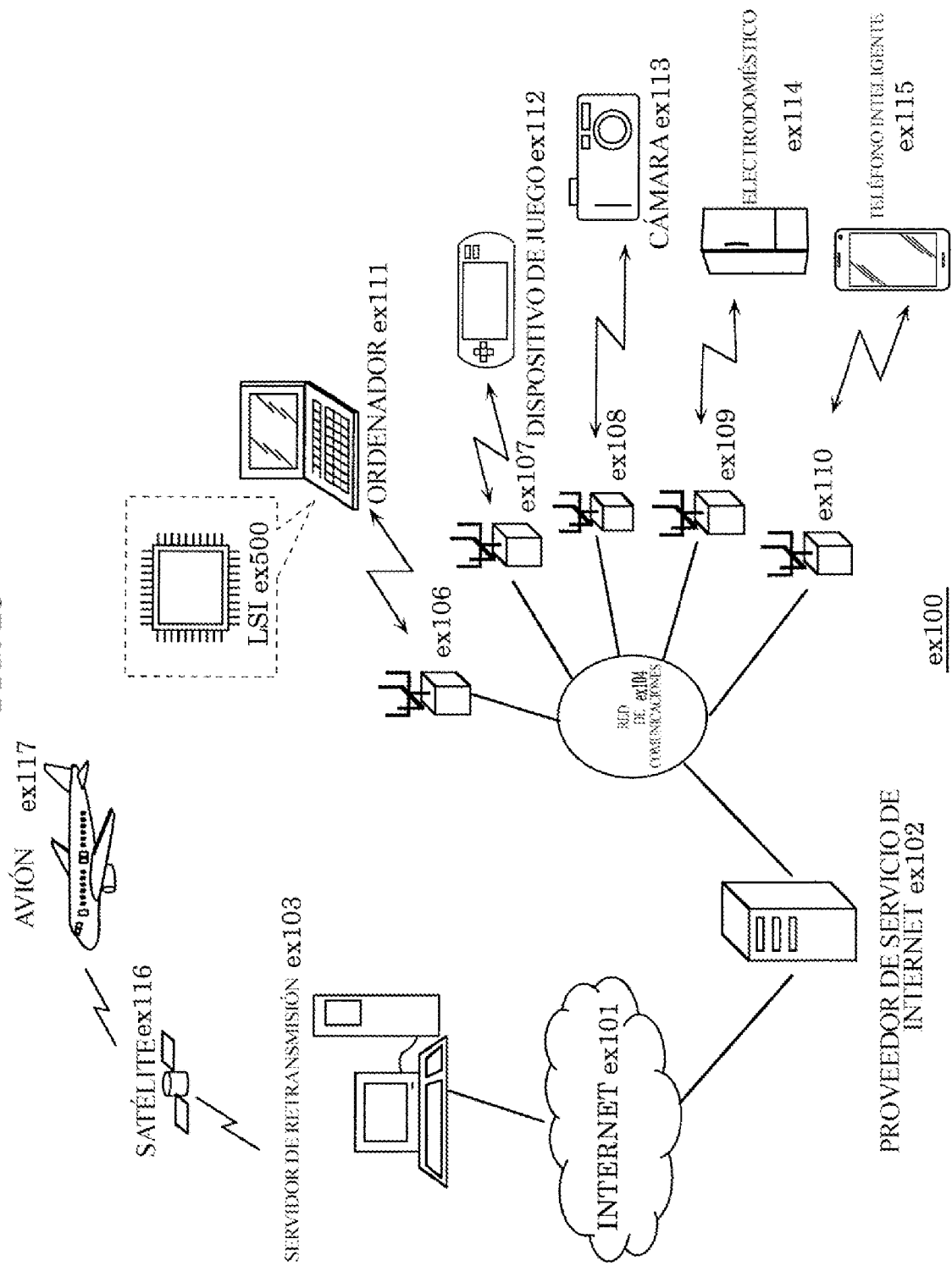


FIG. 26

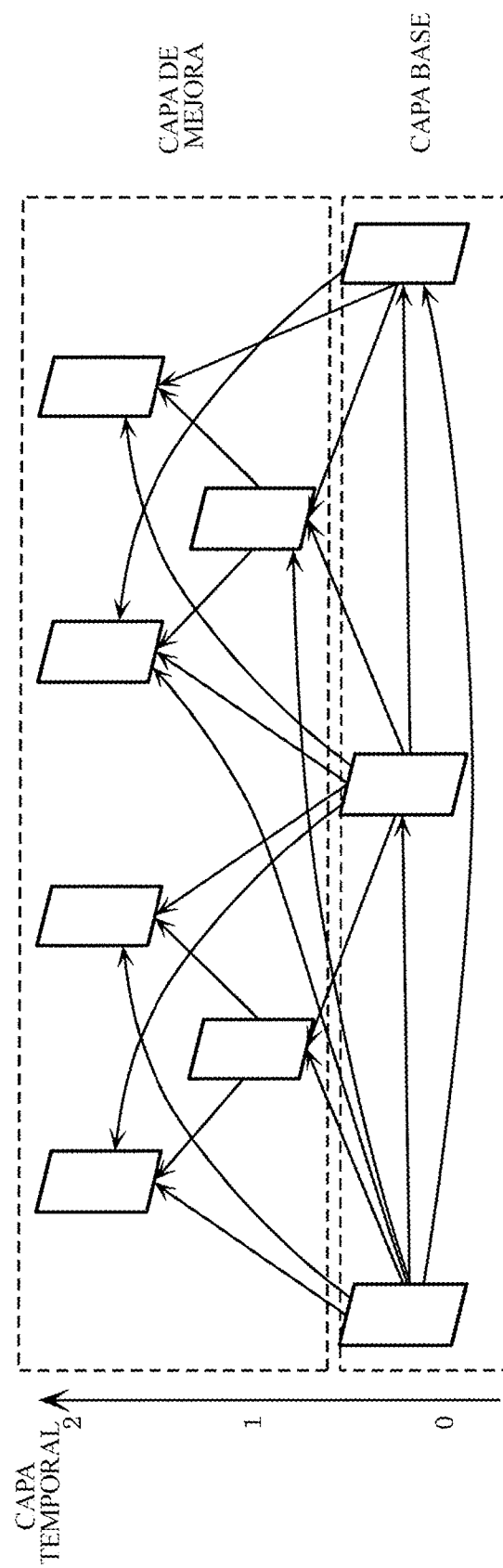


FIG. 27

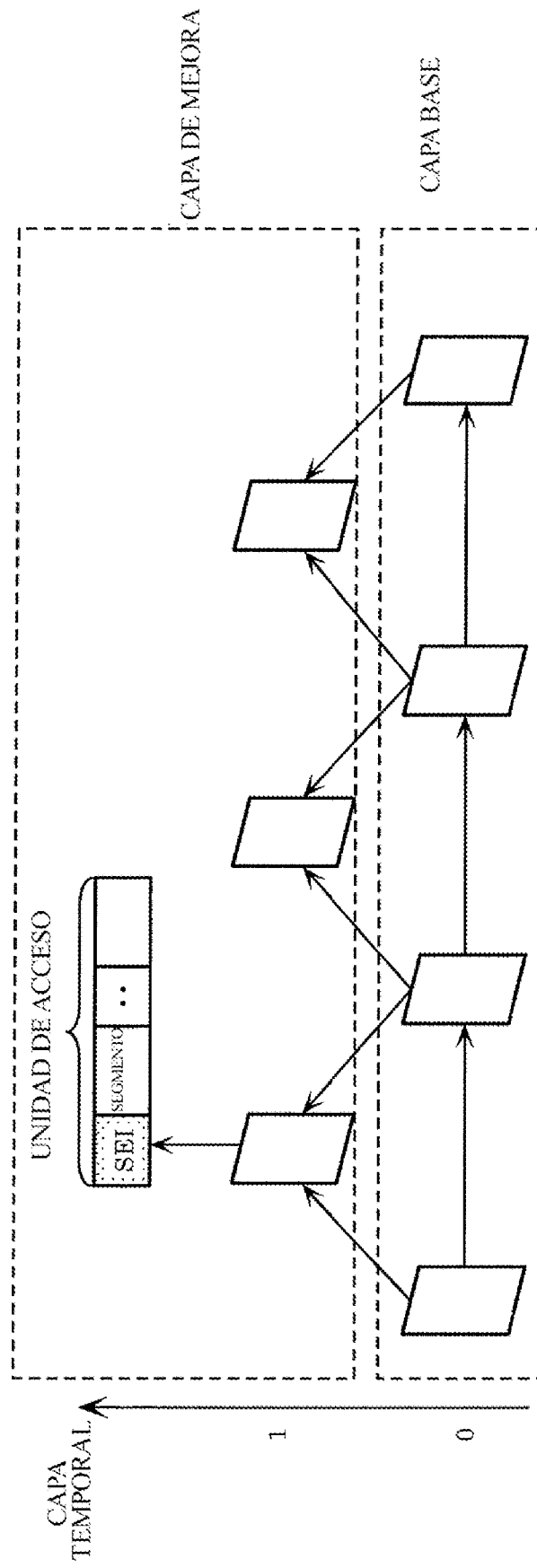


FIG. 28

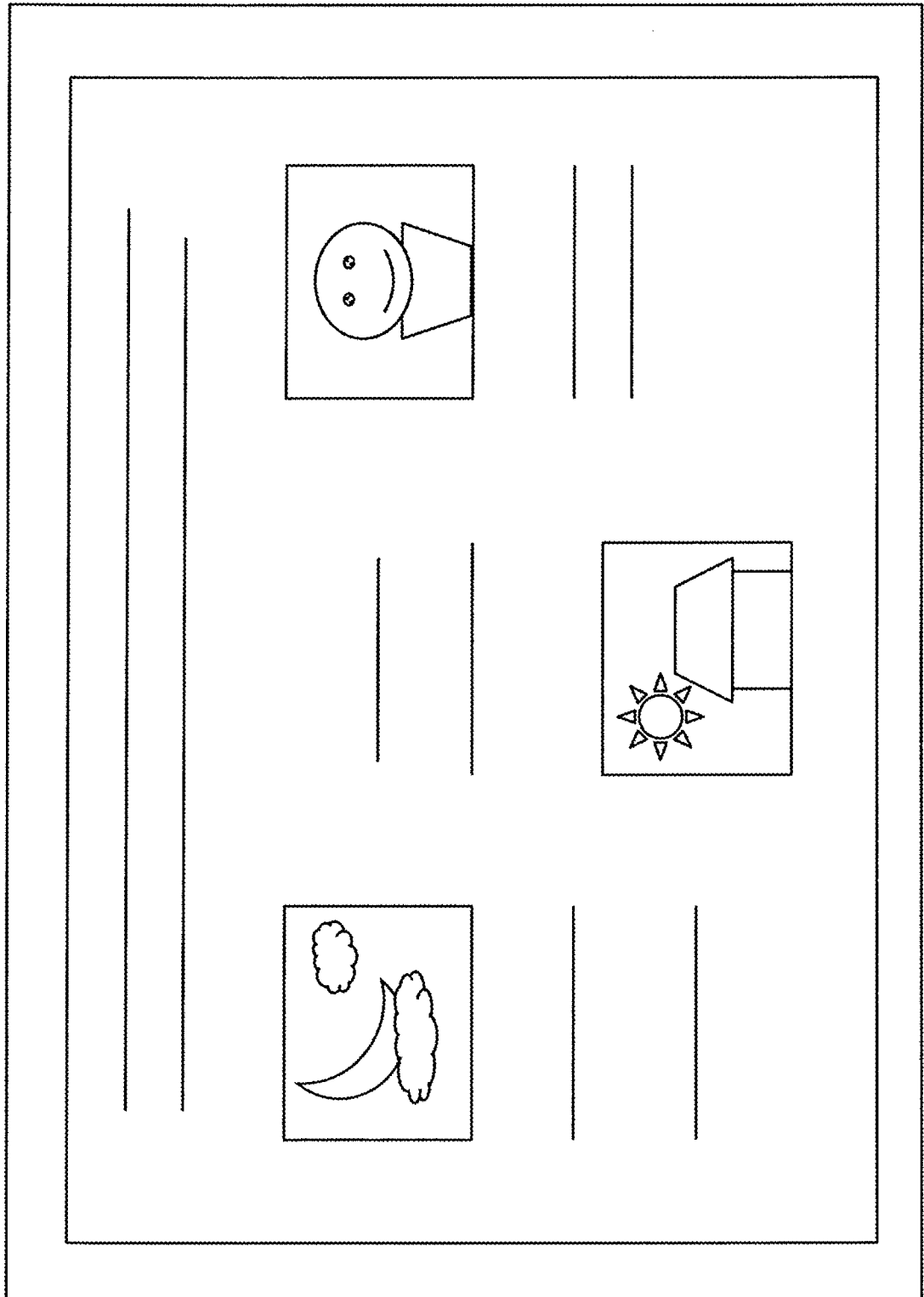


FIG. 29

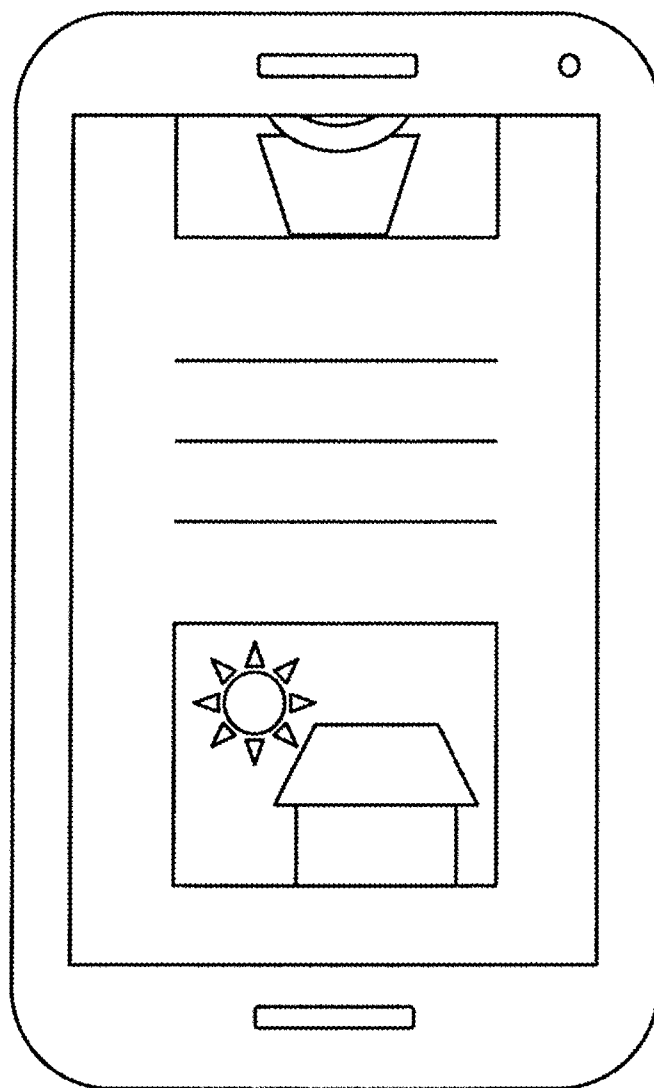


FIG. 30

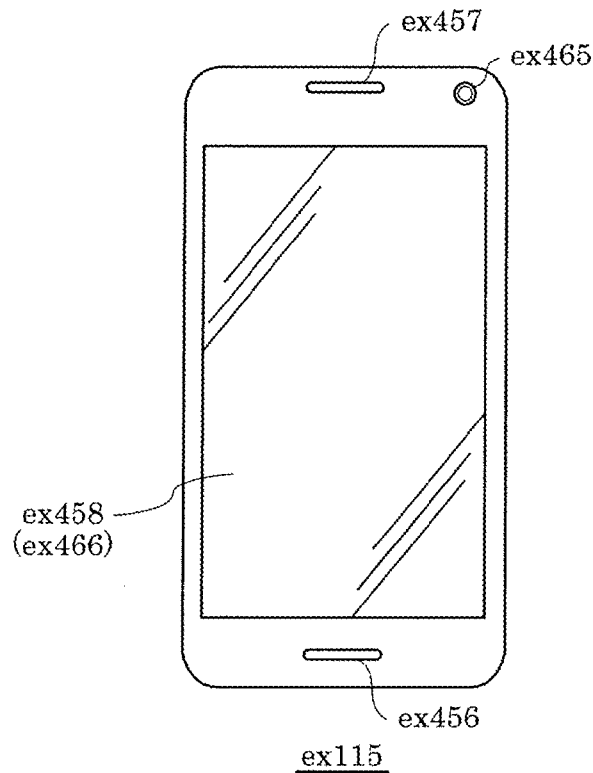


FIG. 31

