

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 993 476**

51 Int. Cl.:

**H04N 19/51** (2014.01)

**H04N 19/176** (2014.01)

**H04N 19/52** (2014.01)

**H04N 19/577** (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **24.04.2018 PCT/JP2018/016515**

87 Fecha y número de publicación internacional: **01.11.2018 WO18199050**

96 Fecha de presentación y número de la solicitud europea: **24.04.2018 E 18790447 (9)**

97 Fecha y número de publicación de la concesión europea: **16.10.2024 EP 3618444**

54 Título: **Dispositivo de codificación, dispositivo de decodificación, método de codificación y método de decodificación**

30 Prioridad:  
**27.04.2017 US 201762490872 P**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**30.12.2024**

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 2 993 476 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 codificación, dispositivo de decodificación, método de codificación y método de decodificación

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

- 5 Los estándares de codificación de video conocidos como codificación de video de alta eficiencia (HEVC) están estandarizados por el Equipo conjunto de colaboración sobre codificación de video (JCT-VC).

Lista de citas

Bibliografía no de patente

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

- 10 Hsiao-Chiang Chuang et al: "A block-based design for Bi-directional optical flow (BIO)", Joint Video Exploration Team (JVET) de ITU-T SG 16 WP 3 y ISO/IEC JTC 1/SC 29/WG 11, JVET-F0022, 6th meeting, March 2016, se refiere a la inter predicción de un bloque actual de una imagen, en la que se describe un diseño basado en bloques para el flujo óptico bidireccional en el que la estimación del movimiento se realiza por bloques.

- 15 Jianle Chen et al: "Algorithm Description of Joint Exploration Test Model 5 (JEM5)", Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, JVET-E1001-V2, 5th meeting, February 2017, se refiere al algoritmo de JEM 5, en el que se realiza la inter predicción utilizando dos imágenes de predicción, imágenes de gradiente respectivas y refinamiento de movimiento basado en píxeles.

En tales técnicas de codificación y decodificación, se desean una eficacia más mejorada de la compresión y una carga de proceso más reducida.

- 20 En vista de esto, la presente divulgación proporciona un codificador, un decodificador, un método de codificación y un método de decodificación capaces de realizar una eficiencia de compresión mejorada y una carga de procesamiento más reducida.

La invención se define en el conjunto de reivindicaciones adjuntas.

- 25 Aspectos generales o específicos de la presente divulgación se pueden realizar como un sistema, método, circuito integrado, programa de computadora, medio legible por computadora tal como un CD-ROM, o cualquier combinación dada de los mismos.

La presente divulgación puede proporcionar un codificador, un decodificador, un método de codificación y un método de decodificación capaces de realizar una eficiencia de compresión mejorada y una carga de procesamiento reducida.

### BREVE DESCRIPCIÓN DE LOS DIBUJOS

- 30 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 en bloques de acuerdo con la Realización 1.

La FIG. 3 es un gráfico que indica las funciones de base de transformación para cada tipo de transformación.

La FIG. 4A ilustra un ejemplo de una forma de filtro usada en ALF.

- 35 La FIG. 4B ilustra otro ejemplo de una forma de filtro usada en ALF.

La FIG. 4C ilustra otro ejemplo de una forma de filtro usada en ALF.

La FIG. 5A ilustra 67 modos de intra predicción usados en intra predicción.

La FIG. 5B es un diagrama de flujo para ilustrar un esquema de un proceso de corrección de imágenes de predicción realizado mediante procesamiento OBMC

- 40 La FIG. 5C es un diagrama conceptual para ilustrar un esquema de un proceso de corrección de la imagen de predicción mediante el procesamiento OBMC.

La FIG. 5D ilustra un ejemplo de FRUC.

La FIG. 6 es para ilustrar la concordancia de patrones (concordancia bilateral) entre dos bloques a lo largo de una trayectoria de movimiento.

La FIG. 7 ilustra la concordancia de patrones (concordancia de plantilla) entre una plantilla de la imagen actual y un bloque en una imagen de referencia.

La FIG. 8 ilustra un modelo que supone un movimiento lineal uniforme.

5 La FIG. 9A ilustra la derivación de un vector de movimiento de cada subbloque a partir de los vectores de movimiento de los bloques vecinos.

La FIG. 9B ilustra el esquema de un proceso para derivar un vector de movimiento mediante el modo de fusión.

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

La FIG. 9D ilustra un esquema de un método de generación de imágenes de predicción que utiliza un proceso de corrección de luminancia realizado mediante procesamiento LIC

10 La FIG. 10 es un diagrama de bloques que ilustra una configuración funcional de un decodificador de acuerdo con la Realización 1.

La FIG. 11 es un cuadro de flujo que ilustra la inter predicción de acuerdo con la Realización 2.

La FIG. 12 es una vista conceptual para explicar la inter predicción de acuerdo con la Realización 2.

15 La FIG. 13 es una vista conceptual para explicar ejemplos de rangos de referencia para un filtro de compensación de movimiento y un filtro de gradiente de acuerdo con la Realización 2.

La FIG. 14 es una vista conceptual para explicar ejemplos de rangos de referencia para un filtro de compensación de movimiento de acuerdo con la Variación 1 de la Realización 2.

La FIG. 15 es una vista conceptual para explicar ejemplos de rangos de referencia para un filtro de gradiente de acuerdo con la Variación 1 de la Realización 2.

20 La FIG. 16 ilustra un ejemplos de patrones de píxeles a los que hacer referencia en la derivación de un valor de estimación de movimiento local de acuerdo con la Variación 2 de la Realización 2.

La FIG. 17 ilustra una configuración general de un sistema de provisión de contenidos para implementar un servicio de distribución de contenidos.

La FIG. 18 ilustra un ejemplo de una estructura de codificación escalable.

25 La FIG. 19 ilustra un ejemplo de una estructura de codificación escalable.

La FIG. 20 ilustra un ejemplo de pantalla de visualización de una página web.

La FIG. 21 ilustra un ejemplo de pantalla de visualización de una página web.

La FIG. 22 ilustra un ejemplo de un teléfono inteligente.

La FIG. 23 es un diagrama de bloques que ilustra un ejemplo de configuración de un teléfono inteligente.

### 30 DESCRIPCIÓN DE EJEMPLOS DE REALIZACIONES

De aquí en adelante, se describirán realizaciones con referencia a los dibujos.

35 Cabe señalar que cada una de las realizaciones descritas a continuación muestra un ejemplo general o específico. Los valores numéricos, las formas, los materiales, los componentes, la disposición y conexión de los componentes, las etapas, el orden de las etapas, etc. que se indican en las siguientes realizaciones son meros ejemplos y, por lo tanto, no pretenden limitar el alcance de las reivindicaciones. Por lo tanto, entre los componentes de las siguientes realizaciones, los que no están mencionados en ninguna de las reivindicaciones independientes que definen los conceptos inventivos más amplios se describen como componentes opcionales.

#### REALIZACIÓN 1

40 En primer lugar, se presentará un esquema de la Realización 1. La Realización 1 es un ejemplo de un codificador y un decodificador a los que son aplicables los procesos y/o configuraciones presentados en la subsiguiente descripción de aspectos de la presente divulgación. Cabe señalar que la Realización 1 no es más que un ejemplo de codificador y decodificador a los que son aplicables los procesos y/o configuraciones presentados en la descripción de aspectos de la presente divulgación. Los procesos y/o configuraciones presentados en la descripción de aspectos de la presente divulgación también se pueden implementar en un codificador y un decodificador diferentes de aquellos de acuerdo  
45 con la Realización 1.

Cuando los procesos 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 las siguientes

- 5 (1) con respecto al codificador o al decodificador de acuerdo con la Realización 1, entre los componentes incluidos en el codificador o el decodificador de acuerdo con la Realización 1, se sustituye 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
- 10 (2) con respecto al codificador o al decodificador de acuerdo con la Realización 1, implementar cambios discrecionales a funciones o procesos implementados realizados por uno o más componentes incluidos en el codificador o el decodificador de acuerdo con la Realización 1, tales como adición, sustitución, o eliminación, etc., de tales funciones o procesos implementados, luego 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;
- 15 (3) con respecto al método implementado por el codificador o el decodificador de acuerdo con la Realización 1, implementar cambios discrecionales tales como adición de procesos y/o sustitución, eliminación de uno o más de los procesos incluidos en el método, y luego sustituir un proceso correspondiente a un proceso presentado en la descripción de aspectos de la presente divulgación por un proceso presentado en la descripción de aspectos de la presente divulgación;
- 20 (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 procesos implementados por un componente presentado en la descripción de aspectos de la presente divulgación;
- 25 (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 procesos 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 procesos implementados por un componente presentado en la descripción de aspectos de la presente divulgación;
- 30 (6) con respecto al método implementado por el codificador o el decodificador de acuerdo con la Realización 1, entre los procesos incluidos en el método, sustituir un proceso correspondiente a un proceso presentado en la descripción de aspectos de la presente divulgación por un proceso presentado en la descripción de aspectos de la presente divulgación; y
- 35 (7) combinar uno o más procesos incluidos en el método implementado por el codificador o el decodificador de acuerdo con la Realización 1 con un proceso presentado en la descripción de aspectos de la presente divulgación.

40 Cabe señalar que la implementación de los procesos y/o configuraciones presentados en la descripción de aspectos de la presente divulgación no se limita a los ejemplos anteriores. Por ejemplo, los procesos 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 imágenes en movimiento/imagen o decodificador de imágenes en movimiento/imágenes divulgado en la Realización 1. Además, los procesos y/o configuraciones presentados en la descripción de aspectos de la presente divulgación se pueden implementar independientemente. Además, los procesos y/o configuraciones descritos en diferentes aspectos se pueden combinar.

[Esquema del codificador]

45 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 imágenes en movimiento/imágenes que codifica una imagen en movimiento/imagen bloque a bloque.

50 Como se ilustra en la FIG. 1, el codificador 100 es un dispositivo que codifica una imagen bloque a 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 trama 122, intra predictor 124, inter predictor 126, y controlador de predicción 128.

55 El codificador 100 se realiza, por ejemplo, como un procesador genérico y una memoria. 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, intra predictor 124, inter predictor 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, intra predictor 124, inter predictor 126, y controlador de predicción 128.

De aquí en adelante, se describirá cada componente incluido en el codificador 100.

5 [Divisor]

El divisor 102 divide cada imagen incluida en una imagen en movimiento de entrada en bloques, y envía cada bloque al sustractor 104. Por ejemplo, el divisor 102 divide primero una imagen en bloques de tamaño fijo (por ejemplo, 128 X128). El bloque de tamaño fijo también se denomina unidad de árbol de codificación (CTU). A continuación, el divisor 102 divide cada bloque de tamaño fijo en bloques de tamaño variable (por ejemplo, 64 X 64 o más pequeños), sobre la base de la división recursiva de bloques en árbol cuádruple y/o árbol binario. El bloque de tamaño variable también se denomina unidad de codificación (CU), unidad de predicción (PU) o unidad de transformación (TU). Cabe señalar que en esta realización no es necesario diferenciar entre CU, PU y TU; todos o algunos de los bloques de una imagen se pueden procesar por CU, PU o TU.

10

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 bloque de bloques divididos por división del bloque de árbol cuádruple y las líneas discontinuas representan límites de bloque de bloques divididos por división de bloques de árbol binario.

15

Aquí, el bloque 10 es un bloque cuadrado de 128 X128 píxeles (bloque 128 X128). Este bloque 128 X128 10 se divide primero en cuatro bloques cuadrados de 64 X 64 (división de bloques en árbol cuádruple).

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

20

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

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

25

El bloque inferior derecho 64 X 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 basados en la división recursiva de bloques en árbol cuádruple y árbol binario. Este tipo de división también se denomina división de cuádruple árbol más árbol binario (QTBT).

35

Cabe señalar que en la FIG. 2, un bloque se divide en cuatro o dos bloques (división de bloques en árbol cuádruple o árbol binario), pero la división no se limita a este ejemplo. Por ejemplo, un bloque se puede dividir en tres bloques (división de bloque ternario). La división que incluye dicha división de bloque ternario también se denomina división de árbol multitypo (MBT).

40 [Sustractor]

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

La señal original es una señal de entrada en el codificador 100, y es una señal que representa una imagen para cada imagen cuadro incluido en una imagen en movimiento (por ejemplo, una señal luma y dos señales croma). De aquí en adelante, una señal que representa una imagen también se denomina muestra.

[Transformador]

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

50

Cabe señalar que el transformador 106 puede seleccionar adaptativamente un tipo de transformación de entre una pluralidad de tipos de transformada, y transformar los errores de predicción en coeficientes de transformada utilizando una función base de transformada correspondiente al tipo de transformada seleccionado. Este tipo de transformada también se denomina transformada de núcleo múltiple explícita (EMT) o transformada múltiple adaptativa (AMT).

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

10 La información que indica si se aplica dicha EMT o AMT (denominada, por ejemplo, un banderín AMT) y la información que indica el tipo de transformación seleccionado se señala a nivel de CU. Cabe señalar 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 imagen, nivel de corte, nivel de mosaico o nivel de CTU).

15 Además, 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 intra predicción. La información que indica si se aplica NSST y la información relacionada con la matriz de transformación utilizada en NSST se señalizan a nivel de  
20 CU. Cabe señalar 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 imagen, nivel de corte, nivel de mosaico o nivel de CTU).

25 Aquí, una transformación separable es un método en el que una transformación se realiza una pluralidad de veces mediante la realización por separado de una transformación para cada dirección de acuerdo con el número de dimensiones de entrada. Una transformación no separable es un método en el que se realiza una transformación colectiva en la que dos o más dimensiones en una entrada multidimensional se consideran colectivamente como una única dimensión.

30 En un ejemplo de transformación no separable, cuando la entrada es un bloque 4 X 4, el bloque 4 X 4 se considera como una única matriz que incluye 16 componentes, y la transformación aplica una matriz de transformación 16 X 16 a la matriz.

Además, similar a lo anterior, después de que un bloque de entrada 4 X 4 es considerado como una matriz única que incluye 16 componentes, una transformación que realiza una pluralidad de rotaciones Givens en la matriz (es decir, una Transformación de Hipercono-Givens) es también un ejemplo de una transformación no separable.

[Cuantificador]

35 El cuantificador 108 cuantifica los coeficientes de transformación de salida 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 sobre la base de los parámetros de cuantificación (QP) correspondientes a los coeficientes de transformación. A continuación, el  
40 cuantificador 108 emite los coeficientes de transformación cuantificados (de aquí en adelante denominados coeficientes cuantificados) del bloque actual al codificador de entropía 110 y al cuantificador inverso 112.

Un orden predeterminado es un orden para cuantificar/cuantificar inversamente coeficientes de transformación. Por ejemplo, un orden predeterminado de escaneado se define como orden ascendente de frecuencia (de baja a alta frecuencia) u orden descendente de frecuencia (de alta a baja frecuencia).

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

[Codificador de entropía]

50 El codificador de entropía 110 genera una señal codificada (corriente 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]

El cuantificador inverso 112 cuantifica inversamente los coeficientes cuantificados, que son entradas desde el cuantificador 108. Más específicamente, el cuantificador inverso 112 cuantifica inversamente, en un orden de escaneado predeterminado, los coeficientes cuantificados del bloque actual. El cuantificador inverso 112 entonces da emite los coeficientes de transformación cuantificados inversos del bloque actual al transformador inverso 114.

5 [Transformador inverso]

El transformador inverso 114 restaura los errores de predicción mediante la transformación inversa de 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 mediante la aplicación de 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.

10

Cabe señalar que como 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]

15 El sumador 116 reconstruye el bloque actual mediante la suma de los errores de predicción, que son entradas del transformador inverso 114, y las muestras de predicción, que son entradas del controlador de predicción 128. El sumador 116 envía entonces el bloque reconstruido a la memoria de bloque 118 y al filtro de bucle 120. El sumador 116 envía el bloque reconstruido a la memoria de bloque 118 y al filtro de bucle 120. Un bloque reconstruido también se denomina como un bloque local decodificado.

20 [Memoria de bloque]

La memoria de bloque 118 es un almacenamiento para almacenar bloques en una imagen para codificar (de aquí en adelante denominada como imagen actual) para referencia en la intra predicción. Más específicamente, la memoria de bloque 118 almacena la salida de bloques reconstruidos del sumador 116.

[Filtro de bucle]

25 El filtro de bucle 120 aplica un filtro de bucle a los bloques reconstruidos mediante el sumador 116, y envía los bloques reconstruidos filtrados a la memoria de trama 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).

30 En ALF, se aplica un filtro de error cuadrático mínimo para eliminar artefactos de compresión. Por ejemplo, un filtro de entre una pluralidad de filtros se selecciona para cada subbloque 2 X 2 en el bloque actual basado en la dirección y actividad de gradientes locales, y se aplica.

Más específicamente, en primer lugar, cada subbloque (por ejemplo, cada subbloque 2 X 2) se clasifica en una de una pluralidad de clases (por ejemplo, 15 o 25 clases). La clasificación del subbloque se basa en la direccionalidad y actividad del gradiente. Por ejemplo, el índice de clasificación C se obtiene a partir de la direccionalidad del gradiente D (por ejemplo, de 0 a 2 o de 0 a 4) y la actividad del gradiente A (por ejemplo, de 0 a 4) (por ejemplo,  $C = 5D + A$ ). A continuación, sobre la base del índice de clasificación C, cada subbloque se clasifica en una de una pluralidad de clases (por ejemplo, 15 o 25 clases).

35 Por ejemplo, la direccionalidad del gradiente D se calcula mediante la comparación de los gradientes de una pluralidad de direcciones (por ejemplo, las direcciones horizontal, vertical y dos diagonales). Además, por ejemplo, la actividad del gradiente A se calcula mediante la suma de gradientes de una pluralidad de direcciones y cuantificación de la suma.

40 El filtro para utilizar para cada subbloque se determina de entre la pluralidad de filtros sobre la base del resultado de dicha categorización.

45 La forma de filtro para utilizar en ALF es, por ejemplo, una forma de filtro simétrica circular. La FIG. 4A a la FIG. 4C ilustran ejemplos de formas de filtro utilizadas en ALF. La FIG. 4A ilustra un filtro con forma de diamante 5 X 5, La FIG. 4B ilustra un filtro con forma de diamante 7 X 7, y la FIG. 4C ilustra un filtro con forma de diamante 9 X 9. La información que indica la forma del filtro se señala a nivel de imagen. Cabe señalar que la señalización de la información que indica la forma del filtro no necesita realizarse a nivel de imagen, y se puede realizar a otro nivel (por ejemplo, a nivel de secuencia, nivel de corte, nivel de mosaico, nivel de CTU o nivel de CU).

50 La activación o desactivación del ALF se determina a nivel de imagen o de CU. Por ejemplo, para luma, la decisión de aplicar o no ALF se toma a nivel de CU, y para croma, la decisión de aplicar o no ALF se toma a nivel de imagen. La información que indica si el ALF está activado o desactivado se señala a nivel de imagen o de CU. Cabe señalar que la señalización de la información que indica si el ALF está activado o desactivado no necesita realizarse a nivel

de imagen o de CU, y se puede realizar a otro nivel (por ejemplo, a nivel de secuencia, nivel de corte, nivel de mosaico o nivel de CTU).

5 Los coeficientes establecidos para la pluralidad de filtros seleccionables (por ejemplo, 15 o 25 filtros) se señalizan en el nivel de imagen. Cabe señalar que la señalización del conjunto de coeficientes no necesita realizarse en el nivel de imagen, y se puede realizar en otro nivel (por ejemplo, en el nivel de secuencia, nivel de corte, nivel de mosaico, nivel de CTU, nivel de CU, o nivel de subbloque).

[Memoria de trama]

10 La memoria de trama 122 almacena imágenes de referencia utilizadas en la inter predicción, y también se denomina buffer de trama. Más específicamente, la memoria de trama 122 almacena bloques reconstruidos filtrados por el filtro de bucle 120.

[Intra Predictor]

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

20 Por ejemplo, el intra predictor 124 realiza la intra predicción mediante el uso de un modo de entre una pluralidad de modos de intra predicción predefinidos. Los modos de intra predicción incluyen uno o más modos de predicción no direccionales y una pluralidad de modos de predicción direccionales.

Los 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 de codificación de video H.265/de alta eficiencia (HEVC) (ver NPL 1).

25 La pluralidad de modos de predicción direccional incluye, por ejemplo, los 33 modos de predicción direccional definidos en el estándar H.265/HEVC. Cabe señalar 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 intra predicción utilizados en la intra predicción (dos modos de predicción no direccional y 65 modos de predicción direccional). Las flechas continuas representan las 33 direcciones definidas en el estándar H.265/HEVC, y las flechas discontinuas representan las 32 direcciones adicionales.

30 Cabe señalar que un bloque luma se puede hacer referencia en la intra predicción de bloque croma. En otras palabras, un componente de croma del bloque actual se puede predecir sobre la base de un componente de luma del bloque actual. Esta intra predicción de bloque también se denomina predicción de modelo lineal de componentes cruzados (CCLM). Este modo de intra predicción de bloque de croma que hace referencia a un bloque de luma (denominado, por ejemplo, modo CCLM) se puede añadir como uno de los modos de intra predicción de bloque de croma.

35 El intra predictor 124 puede corregir valores de píxel pos-intra predicción sobre la base de gradientes de píxel de referencia horizontales/verticales. La intra predicción acompañada de este tipo de corrección también se denomina combinación de intra predicción dependiente de la posición (PDPC). La información que indica si se aplica PDPC o no (denominada, por ejemplo, un banderín PDPC) se señala, por ejemplo, a nivel de CU. Cabe señalar la señalización de esta información no es necesario que se realice a nivel de CU, sino que se puede realizar a otro nivel (por ejemplo, a nivel de secuencia, nivel de imagen, nivel de corte, nivel de mosaico o nivel de CTU).

40 [Inter predictor]

45 El inter predictor 126 genera una señal de predicción (señal de inter predicción) mediante la inter predicción del bloque actual con referencia a un bloque o bloques de una imagen de referencia, que es diferente de la imagen actual y se almacena en la memoria de trama 122 (también denominada inter predicción de tramas). La inter predicción se realiza por bloque actual o por subbloque (por ejemplo, por bloque 4 X 4) en el bloque actual. Por ejemplo, el inter predictor 126 realiza una estimación de movimiento en una imagen de referencia para el bloque o subbloque actual. A continuación, el inter predictor 126 genera una señal de inter predicción del bloque o subbloque actual mediante compensación de movimiento utilizando la información de movimiento (por ejemplo, un vector de movimiento) obtenida de la estimación de movimiento. A continuación, el inter predictor 126 envía la señal de inter predicción generada al controlador de predicción 128.

50 La información de movimiento utilizada en la compensación de movimiento es señalizada. Un predictor de vector de movimiento se puede usar 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.

Cabe señalar que la señal de inter predicción se puede generar usando información de movimiento para un bloque vecino además de la información de movimiento para el bloque actual obtenida de la estimación de movimiento. Más

específicamente, la señal de inter predicción se puede generar por subbloque en el bloque actual mediante el cálculo de una suma ponderada de una señal de predicción basada en información de movimiento obtenida de la estimación de movimiento y una señal de predicción basada en información de movimiento para un bloque vecino. Dicha inter predicción (compensación de movimiento) también se denomina compensación de movimiento de bloque superpuesto (OBMC).

En tal modo OBMC, la información que indica el tamaño de subbloque para OBMC (denominado como por ejemplo, tamaño de bloque OBMC) se señala a nivel de secuencia. Además, la información que indica si se aplica o no el modo OBMC (denominada, por ejemplo, un banderín OBMC) se señala a nivel de CU. Cabe señalar que la señalización de dicha información no necesita realizarse a nivel de secuencia y a nivel de CU, sino que se puede realizar a otro nivel (por ejemplo, a nivel de imagen, nivel de corte, nivel de mosaico, nivel de CTU o nivel de subbloque).

El inter predictor 126 genera una señal de predicción (señal de inter predicción) mediante la inter predicción del bloque actual con referencia a un bloque o bloques de una imagen de referencia, que es diferente de la imagen actual y está almacenada en la memoria de trama 122 (también denominada inter predicción de tramas). La inter predicción se realiza por bloque actual o por subbloque (por ejemplo, por bloque 4 X 4) en el bloque actual. Por ejemplo, el inter predictor 126 realiza una estimación de movimiento en una imagen de referencia para el bloque o subbloque actual. A continuación, el inter predictor 126 genera una señal de inter predicción del bloque o subbloque actual mediante compensación de movimiento utilizando la información de movimiento (por ejemplo, un vector de movimiento) obtenida de la estimación de movimiento. A continuación, el inter predictor 126 envía la señal de inter predicción generada al controlador de predicción 128.

Se señala la información de movimiento utilizada en la compensación de movimiento. Un predictor de vector de movimiento se puede utilizar 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.

Cabe señalar que la señal de inter predicción se puede generar usando información de movimiento para un bloque vecino además de la información de movimiento para el bloque actual obtenida de la estimación de movimiento. Más específicamente, la señal de inter predicción se puede generar por subbloque en el bloque actual mediante el cálculo de una suma ponderada de una señal de predicción basada en información de movimiento obtenida de la estimación de movimiento y una señal de predicción basada en información de movimiento para un bloque vecino. Dicha inter predicción (compensación de movimiento) también se denomina compensación de movimiento de bloque superpuesto (OBMC).

En dicho modo OBMC, la información que indica el tamaño de subbloque para OBMC (denominado, por ejemplo, tamaño de bloque OBMC) se señala a nivel de secuencia. Además, la información que indica si se aplica o no el modo OBMC (denominada, por ejemplo, un banderín de OBMC) se señala a nivel de CU. Cabe señalar que la señalización de dicha información no necesita realizarse a nivel de secuencia y a nivel de CU, y se puede realizar a otro nivel (por ejemplo, a nivel de imagen, nivel de corte, nivel de mosaico, nivel de CTU o nivel de subbloque).

De aquí en adelante, el modo OBMC se describirá con más detalle. La FIG. 5B es un diagrama de flujo y La FIG. 5C es un diagrama conceptual para derivar un esquema de un proceso de corrección de imagen de predicción realizado mediante procesamiento OBMC.

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

A continuación, se obtiene una imagen de predicción (Pred\_L) mediante la aplicación de un vector de movimiento (MV\_L) del bloque vecino izquierdo codificado al bloque actual, y se realiza una primera pasada de la corrección de la imagen de predicción mediante la superposición de la imagen de predicción y Pred\_L.

Del mismo modo, se obtiene una imagen de predicción (Pred\_U) mediante la aplicación de un vector de movimiento (MV\_U) del bloque superior vecino codificado al bloque actual, y se realiza una segunda pasada de corrección de la imagen de predicción mediante la superposición de la imagen de predicción resultante de la primera pasada y Pred\_U. El resultado de la segunda pasada es la imagen de predicción final.

Cabe señalar que el ejemplo anterior es de un método de corrección de dos pasadas usando los bloques vecinos izquierdo y superior, pero el método puede ser un método de corrección de tres pasadas o superior que también use el bloque vecino derecho y/o inferior.

Cabe señalar que la región sujeta a superposición puede ser la región de píxeles entera del bloque y, alternativamente, puede ser una región parcial del límite del bloque.

Cabe señalar que aquí, el proceso de corrección de la imagen de predicción se describe como basado en una imagen de referencia única, pero lo mismo se aplica cuando una imagen de predicción se corrige sobre la base de una pluralidad de imágenes de referencia. En tal caso, después de obtener las imágenes de predicción corregidas

resultantes de realizar la corrección basada en cada una de las imágenes de referencia, las imágenes de predicción corregidas obtenidas también se superponen para obtener la imagen de predicción final.

Cabe señalar que la unidad del bloque actual puede ser un bloque de predicción y, alternativamente, puede ser un subbloque obtenido mediante la división adicional del bloque de predicción.

5 Un ejemplo de un método para determinar si se implementa el procesamiento OBMC es mediante el uso de un `obmc_flag`, que es una señal que indica si se implementa el procesamiento 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` a un valor de «1» cuando el bloque pertenece a una región que incluye movimiento complicado e implementa el procesamiento OBMC cuando se codifica, y establece el `obmc_flag` a un valor de «0» cuando el  
10 bloque no pertenece a una región que incluye movimiento complicado y codifica sin implementar el procesamiento OBMC. El decodificador cambia entre implementar o no el procesamiento OBMC mediante la decodificación del `obmc_flag` escrito en el flujo y realización de la decodificación de acuerdo con el valor del banderín.

Cabe señalar que la información de movimiento se puede derivar del lado del decodificador sin ser señalizada. Por ejemplo, se puede utilizar un modo de fusión definido en el estándar H.265/HEVC. Además, por ejemplo, la información de movimiento se puede derivar mediante la realización de una estimación de movimiento en el lado del decodificador.  
15 En este caso, la estimación de movimiento se realiza sin utilizar los valores de píxel del bloque actual.

Aquí, se describirá un modo para realizar la estimación del movimiento en el lado del decodificador. Un modo para realizar la estimación de movimiento en el lado del decodificador también se denomina como modo de derivación de vector de movimiento de patrón apareado (PMMVD) o modo de conversión ascendente de tasa de trama (FRUC).

20 Un ejemplo de procesamiento 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, cada uno de los cuales incluye 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 registradas en la lista de candidatas. Por ejemplo, los valores de la evaluación para los candidatos incluidos  
25 en la lista del candidato se calculan y se selecciona un candidato sobre la base de los valores calculados de la evaluación.

Después, un vector del movimiento para el bloque actual se deriva del vector del 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), como tal. Alternativamente, el vector de movimiento para el bloque actual se puede derivar de la concordancia de patrones realizada en las proximidades de una posición en una imagen de referencia correspondiente al vector de movimiento del candidato seleccionado. En otras palabras, cuando la proximidad del mejor MV candidato se busca a través del mismo método 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. Cabe señalar que también es aceptable una configuración en la que no se aplique este proceso.  
35

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

Cabe señalar que un valor de evaluación se calcula mediante el cálculo de la diferencia en la imagen reconstruida mediante la concordancia de patrones realizada entre una región de una imagen de referencia correspondiente a un vector de movimiento y una región predeterminada. Cabe señalar que el valor de evaluación se puede calcular utilizando alguna otra información además de la diferencia.  
40

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

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

La FIG. 6 ilustra un ejemplo de concordancia de patrones (concordancia bilateral) entre dos bloques a lo largo de una trayectoria de movimiento. Como se ilustra en La FIG. 6, en la primera concordancia de patrones, dos vectores de movimiento (MV0, MV1) se derivan mediante el hallazgo de la mejor concordancia entre dos bloques a lo largo de la trayectoria de movimiento del bloque actual (bloque actual) en dos imágenes de referencia diferentes (Ref0, Ref1). Más específicamente, se puede derivar una diferencia entre (i) una imagen reconstruida en una posición especificada en una primera imagen de referencia codificada (Ref0) especificada por un MV candidato y (ii) una imagen reconstruida en una posición especificada en una segunda imagen de referencia codificada (Ref1) especificada por un MV simétrico escalado a un rango de tiempo de visualización del MV candidato, y el valor de evaluación para el bloque actual se puede calcular utilizando la diferencia derivada. El MV candidato que tiene el mejor valor de evaluación entre la pluralidad de MV candidatos se puede seleccionar como MV final.  
50  
55

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 la imagen actual (Imag actual) y las dos imágenes de referencia (Ref0, Ref1). Por ejemplo, cuando la imagen actual se encuentra temporalmente entre las dos imágenes de referencia y la distancia temporal entre la imagen actual a las dos imágenes de referencia es la misma, la primera concordancia de patrón deriva un vector de movimiento bidireccional basado en el espejo.

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

La FIG. 7 ilustra un ejemplo de concordancia de patrones (concordancia de plantillas) entre una plantilla de la imagen actual y un bloque de una imagen de referencia. Como se ilustra en La FIG. 7, en la segunda concordancia de patrones, un vector de movimiento del bloque actual se deriva mediante la búsqueda de una imagen de referencia (Ref0) para encontrar el bloque que mejor concuerde con los bloques vecinos del bloque actual (bloque act) en la imagen actual (Imag act). Más específicamente, se puede derivar una diferencia entre (i) una imagen reconstruida de una región codificada que es ambas o una de las regiones vecinas izquierda y superior y (ii) una imagen reconstruida en la misma posición en una imagen de referencia codificada (Ref0) especificada por un MV, y el valor de evaluación para el bloque actual se puede calcular utilizando la diferencia derivada. El MV candidato que tiene el mejor valor de evaluación entre la pluralidad de MV candidato se podrá seleccionar como el mejor MV candidato.

La información que indica si se aplica o no el modo FRUC (denominado, por ejemplo, un banderín FRUC) se señala en el nivel de CU. Además, cuando se aplica el modo FRUC (por ejemplo, cuando el banderín FRUC se establece como verdadero), la información que indica el método de concordancia de patrones (primera concordancia de patrones o segunda concordancia de patrones) se señala a nivel de CU. Cabe señalar que la señalización de dicha información no se necesita realizar a nivel de CU, y se puede realizar a otro nivel (por ejemplo, nivel de secuencia, nivel de imagen, nivel de corte, nivel de mosaico, nivel de CTU o nivel de subbloque).

Aquí se describirá un modo para derivar un vector de movimiento basado en un modelo que asume un movimiento lineal uniforme. Este modo también se denomina modo de flujo óptico bidireccional (BIO).

La FIG. 8 ilustra un modelo que asume un movimiento lineal uniforme. En la FIG. 8,  $(v_x, v_y)$  indica un velocidad de vector, y  $t_0$  y  $t_1$  indica distancias temporales entre la imagen actual (Imag act) y dos imágenes de referencia (Ref0, Ref1).  $(MVx_0, MVy_0)$  indica un vector de movimiento correspondiente a la imagen de referencia Ref0, y  $(MVx_1, MVy_1)$  indica un vector de movimiento correspondiente a la imagen de referencia Ref1.

Aquí, bajo el supuesto de movimiento lineal uniforme exhibido por la velocidad de vector  $(v_x, v_y)$ ,  $(MVx_0, MVy_0)$  y  $(MVx_1, MVy_1)$  están representados como  $(v_x t_0, v_y t_0)$  y  $(-v_x t_1, -v_y t_1)$ , respectivamente, y se da la siguiente ecuación de flujo óptico.

MAT. 1

$$\frac{\partial I^{(k)}}{\partial t} + v_x \frac{\partial I^{(k)}}{\partial x} + v_y \frac{\partial I^{(k)}}{\partial y} = 0. \quad (1)$$

Aquí,  $I^{(k)}$  indica un valor luma de la imagen 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 de tiempo del valor luma, (ii) el producto de la velocidad horizontal y el componente horizontal del gradiente espacial de una imagen de referencia, y (iii) el producto de la velocidad vertical y el componente vertical del gradiente espacial de una imagen 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 pixel a pixel sobre la base de una combinación de la ecuación de flujo óptico y la interpolación de Hermite.

Cabe señalar que un vector de movimiento se puede derivar en el lado del decodificador usando un método diferente a derivar un vector de movimiento basado en un modelo asumiendo movimiento lineal uniforme. Por ejemplo, un vector de movimiento se puede derivar para cada subbloque basado en vectores de movimientos de bloques vecinos.

Aquí, se describirá un modo en el cual un vector de movimiento se deriva para cada subbloque basado en el vector de movimientos de bloques vecinos. Este modo también se denomina modo de predicción de compensación de movimiento afín.

La FIG. 9A ilustra la derivación de un vector de movimiento de cada subbloque basado en vectores de movimiento de bloques vecinos. En La FIG. 9A, el bloque actual incluye 16 subbloques  $4 \times 4$ . Aquí, el vector de movimiento  $v_0$  del punto de control de la esquina superior izquierda del bloque actual se deriva en base a los vectores de movimiento de los subbloques vecinos, y el vector de movimiento  $v_1$  del punto de control de la esquina superior derecha del bloque actual se deriva sobre la base de los vectores de movimiento de los bloques vecinos. A continuación, utilizando los

dos vectores de movimiento  $v_0$  y  $v_1$ , se obtiene el vector de movimiento  $(v_x, v_y)$  de cada subbloque del bloque actual mediante la ecuación 2 siguiente.

MAT. 2

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

5 Aquí,  $x$  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 afín puede incluir varios modos de diferentes métodos 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 afín (denominada, por ejemplo, banderín afín) se señala a nivel de CU. Cabe señalar que la señalización de la información que indica el modo de predicción de compensación de movimiento afín no se necesita realizar a nivel de CU, y se puede realizar a otro nivel (por ejemplo, a nivel de secuencia, nivel de imagen, nivel de corte, nivel de mosaico, nivel de CTU o nivel de subbloque).

[Controlador de predicción]

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

Aquí, se dará un ejemplo de derivación de un vector de movimiento por medio del modo de fusión en una imagen actual. La FIG. 9B ilustra un esquema de un proceso para derivar un vector de movimiento por medio del modo de fusión.

20 Primero, se genera una lista del predictor del MV en la cual se registran los predictores candidatos del MV. Los ejemplos de predictores de MV candidatos incluyen: predictores de MV vecinos espaciales, que son MV de bloques codificados situados en la proximidad espacial del bloque actual; un predictor de MV vecino temporalmente, que es un MV de un bloque en una imagen 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 mediante la combinación de los valores de 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 mediante la selección de un predictor de VM de entre la pluralidad de predictores de VM registrados en la lista de predictores de VM.

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

Cabe señalar que los predictores MV registrados en la lista de predictores 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.

Cabe señalar que el MV final se puede determinar mediante la realización de un procesamiento DMVR (que se describirá más adelante) utilizando el MV del bloque actual derivada a través del modo de fusión.

Aquí, se dará un ejemplo de determinación de un MV usando el procesamiento de DMVR.

40 La FIG. 9C es un diagrama conceptual para derivar un esquema del procesamiento 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 una primera imagen de referencia, que es una imagen procesada en la dirección L0 de acuerdo con el MV candidato, y una segunda imagen de referencia, que es una imagen procesada en la dirección L1 de acuerdo con el MV candidato, y se genera una plantilla mediante el cálculo de la media de los píxeles de referencia.

A continuación, utilizando la plantilla, se buscan las regiones circundantes de los MV candidatos de la primera y segunda imágenes de referencia, y se determina que el MV con el menor costo es el MV final. Cabe señalar que el valor del coste se calcula utilizando, por ejemplo, la diferencia entre cada valor de píxel de la plantilla y cada valor de píxel de las regiones buscadas, así como el valor del MV.

5 Cabe señalar que los esquemas de los procesos aquí descritos son fundamentalmente los mismos tanto en el codificador como en el decodificador.

Cabe señalar que se puede utilizar un procesamiento distinto del descrito anteriormente, siempre que el procesamiento sea capaz de derivar el MV final mediante la búsqueda en los alrededores del MV candidato.

Aquí, se dará un ejemplo de un modo que genera una imagen de predicción utilizando el procesamiento LIC.

10 La FIG. 9D ilustra un esquema de un método de generación de imagen de predicción que utiliza un proceso de corrección de luminancia realizado mediante procesamiento LIC.

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

15 A continuación, se extrae la información que indica cómo ha cambiado el valor de luminancia entre la imagen de referencia y la imagen actual y se calcula un parámetro de corrección de luminancia utilizando 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 la imagen de referencia especificada por el MV.

20 La imagen de predicción para el bloque actual se genera mediante la realización de un proceso de corrección de luminancia utilizando el parámetro de corrección de luminancia en la imagen de referencia en la imagen de referencia especificada por el MV.

Cabe señalar 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.

25 Además, aunque en este ejemplo se genera una imagen de predicción a partir de una única imagen de referencia, en los casos en los que también se genera una imagen de predicción a partir de una pluralidad de imágenes de referencia, la imagen de predicción se genera después de realizar un proceso de corrección de luminancia, mediante el mismo método, en las imágenes de referencia obtenidas a partir de las imágenes de referencia.

30 Un ejemplo de método para determinar si se implementa el procesamiento LIC es mediante el uso de un lic\_flag, que es una señal que indica si se implementa el procesamiento LIC. Como un ejemplo específico, el codificador determina si el bloque actual pertenece a una región de cambio de luminancia e implementa el procesamiento LIC al codificar, y establece el lic\_flag a un valor de "0" cuando el bloque no pertenece a una región de cambio de luminancia y codifica sin implementar el procesamiento LIC. El decodificador cambia entre implementar o no el procesamiento LIC mediante la decodificación de lic\_flag escrita en el flujo y la realización de la decodificación de acuerdo con el valor del banderín.

35 Un ejemplo de un método diferente para determinar si se implementa el procesamiento LIC es determinarlo en función de si se determinó que se implementará el procesamiento LIC 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 LIC en la codificación del bloque codificado circundante seleccionado después de derivar el MV en el procesamiento del modo de fusión, y se puede cambiar o no la implementación de procesamiento LIC sobre la base del resultado de la determinación. Cabe señalar que, en este ejemplo, lo mismo se aplica al procesamiento realizado en el lado del  
40 decodificador.

[Esquema de decodificador]

45 A continuación, se describirá un decodificador capaz de decodificar una señal codificada (flujo de bits codificado) emitida desde 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 imagen en movimiento/imagen que decodifica una imagen en movimiento bloque a bloque.

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

50 El decodificador 200 se realiza como, por ejemplo, un procesador y 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, intra predictor 216, inter predictor 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, intra predictor 216, inter predictor 218, y controlador de predicción 220.

De aquí en adelante, se describirá cada componente incluido en el decodificador 200.

[Decodificador de Entropía]

- 5 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. El decodificador de entropía 202 entonces debinariza la señal binaria. Con esto, el decodificador de entropía 202 envía coeficientes cuantificados de cada bloque al cuantificador inverso 204.

[Cuantificador Inverso]

- 10 El cuantificador inverso 204 cuantifica inversamente los coeficientes cuantificados de un bloque para decodificar (de aquí en adelante denominado como un bloque actual), que son entradas del decodificador de entropía 202. Más específicamente, el cuantificador inverso 204 cuantifica inversamente los coeficientes cuantificados del bloque actual sobre la base de los parámetros de cuantificación correspondientes a los coeficientes cuantificados. El cuantificador inverso 204 envía entonces los coeficientes cuantificados inversamente (es decir, los coeficientes de transformación)  
15 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 desde el cuantificador inverso 204.

- 20 Por ejemplo, cuando la información analizada de un flujo de bits codificado indica la aplicación de EMT o AMT (por ejemplo, cuando el banderín AMT se establece en verdadero), el transformador inverso 206 transforma inversamente los coeficientes de transformación del bloque actual sobre la base de la información que indica el tipo de transformación analizada.

Además, por ejemplo, cuando la información analizada 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.

- 25 [Sumador]

El sumador 208 reconstruye el bloque actual mediante la suma de los errores de predicción, que son entradas del transformador inverso 206, y las muestras de predicción, que es una entrada del controlador de predicción 220. El sumador 208 emite entonces el bloque reconstruido a la memoria de bloque 210 y al filtro de bucle 212.

[Memoria de bloque]

- 30 La memoria de bloque 210 es un almacenamiento para almacenar bloques en una imagen para decodificar (de aquí en adelante denominada como imagen actual) para referencia en intra predicción. Más específicamente, la memoria de bloque 210 almacena bloques reconstruidos enviados del sumador 208.

[Filtro de bucle]

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

Cuando la información que indica la activación o desactivación de ALF analizada desde un flujo de bits codificado indica activado, se selecciona un filtro de entre una pluralidad de filtros sobre la base de la dirección y actividad de gradientes locales, y el filtro seleccionado se aplica al bloque reconstruido.

[Memoria de trama]

- 40 La memoria de trama 214 es un almacenamiento para almacenar imágenes de referencia utilizadas en la inter predicción, y también se denomina buffer de trama. Más específicamente, la memoria de trama 214 almacena bloques reconstruidos filtrados por el filtro de bucle 212.

[Intra Predictor]

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

Cabe señalar que cuando se selecciona un modo de intra predicción en el que un bloque de croma se intra predice a partir de un bloque de luma, el intra predictor 216 puede predecir el componente de croma del bloque actual basado en el componente de luma del bloque actual.

5 Además, cuando la información que indica la aplicación de PDPC se analiza a partir de un flujo de bits codificado, el intra predictor 216 corrige los valores de píxel pos-intra predicción sobre la base de gradientes de píxel de referencia horizontales/verticales.

[Inter Predictor]

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

15 Cabe señalar que cuando la información analizada del flujo de bits codificado indica la aplicación del modo OBMC, el inter predictor 218 genera la señal de inter predicción usando información de movimiento para un bloque vecino además de la información de movimiento para el bloque actual obtenida de la estimación de movimiento.

20 Además, cuando la información analizada del flujo de bits codificado indica la aplicación del modo FRUC, el inter predictor 218 deriva la información de movimiento mediante la realización de la estimación de movimiento de acuerdo con el método de concordancia de patrón (concordancia bilateral o concordancia de plantilla) analizado del flujo de bits codificado. El inter predictor 218 entonces realiza compensación de movimiento usando la información de movimiento derivada.

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

25 [Controlador de predicción]

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

## REALIZACIÓN 2

30 A continuación, se describirá la Realización 2. La presente realización se refiere a la inter predicción en un modo denominado BIO. La presente realización difiere de la Realización 1 descrita anteriormente en que un vector de movimiento por bloque se corrige sobre una base por subbloque, no sobre una base por píxel. A continuación se describe la presente realización que se centra principalmente en los puntos diferentes de la Realización 1 descrita anteriormente.

35 Cabe señalar que las configuraciones del codificador y decodificador según la presente realización son sustancialmente las mismas que las ilustradas en la Realización 1. Por consiguiente, se omiten la representación y descripción duplicadas.

[Inter predicción]

40 La FIG. 11 es un cuadro de flujo que ilustra la inter predicción de acuerdo con la Realización 2. La FIG. 12 es una vista conceptual para explicar la inter predicción de acuerdo con la Realización 2. El siguiente proceso es realizado por el inter predictor 126 en el codificador 100 o el inter predictor 218 en el decodificador 200.

Como se ilustra en para codificar o decodificar (imagen actual 1000) (S101-S111). En la FIG. 12, un bloque actual para codificar o decodificar se selecciona como bloque actual 1001 de entre una pluralidad de bloques.

45 En el proceso de bucle realizado por bloque, el proceso de bucle se realiza por imagen de referencia para la primera imagen de referencia 1100 (L0) y la segunda imagen de referencia 1200 (L1) que son imágenes procesadas (S102-S106).

50 En el proceso de bucle realizado sobre una base por imagen de referencia, un vector de movimiento para obtener una imagen de predicción a partir de una imagen de referencia se deriva u obtiene en primer lugar sobre una base por bloque (S103). En la FIG. 12, el primer vector de movimiento 1110 (MV\_L0) se deriva u obtiene para la primera imagen de referencia 1100, y el segundo vector de movimiento 1210 (MV\_L1) se deriva u obtiene para la segunda imagen de referencia 1200. Los métodos para derivar un vector de movimiento incluyen un modo de inter predicción normal, un modo de fusión, un modo FRUC, etc. En el caso de un modo de inter predicción normal, por ejemplo, el codificador 100 obtiene un vector de movimiento a través de la estimación de movimiento, mientras que el decodificador 200 obtiene un vector de movimiento a partir de un flujo de bits.

A continuación, se realiza la compensación de movimiento utilizando el vector de movimiento derivado u obtenido, y se obtiene así una imagen de predicción a partir de una imagen de referencia. En la FIG. 12, la compensación de movimiento se realiza utilizando el primer vector de movimiento 1110, y la primera imagen de predicción 1140 se obtiene a partir de la primera imagen de referencia 1100. Además, la compensación de movimiento se realiza usando el segundo vector de movimiento 1210, y la segunda imagen de predicción 1240 se obtiene de la segunda imagen de referencia 1200.

En la compensación de movimiento, se aplica un filtro de compensación de movimiento a una imagen de referencia. El filtro de compensación de movimiento es un filtro de interpolación para obtener una imagen de predicción con exactitud de subpíxel. En la primera imagen de referencia 1100 de la FIG. 12, los píxeles en el primer rango de referencia de interpolación 1130, que incluyen píxeles en el primer bloque de predicción 1120 y los píxeles circundantes, son referidos por un filtro de compensación de movimiento para el primer bloque de predicción 1120 especificado por el primer vector de movimiento 1110. En la segunda imagen de referencia 1200, los píxeles de un segundo rango de referencia de interpolación 1230, que incluyen los píxeles del segundo bloque de predicción 1220 y los píxeles circundantes, son referidos por un filtro de compensación de movimiento para el segundo bloque de predicción 1220 especificado por el segundo vector de movimiento 1210.

Cabe señalar que el primer rango de referencia de interpolación 1130 y el segundo rango de referencia de interpolación 1230 están incluidos en un primer rango de referencia normal y un segundo rango de referencia normal, respectivamente, a los que se hace referencia para la compensación de movimiento del bloque actual 1001 en la inter predicción normal realizada sin utilizar valores de estimación de movimiento local. El primer rango de referencia normal se incluye en la primera imagen de referencia 1100, y el segundo rango de referencia normal se incluye en la segunda imagen de referencia 1200. En la inter predicción normal, se obtiene un vector de movimiento por bloque mediante la estimación del movimiento, se realiza una compensación del movimiento por bloque utilizando el vector de movimiento derivado y por ejemplo, se utiliza directamente una imagen con compensación del movimiento como imagen de predicción final. En otras palabras, los valores locales de estimación de movimiento no se utilizan en la inter predicción normal. Cabe señalar que el primer rango de referencia de interpolación 1130 y el segundo rango de referencia de interpolación 1230 pueden coincidir con el primer rango de referencia normal y el segundo rango de referencia normal, respectivamente.

Posteriormente, se obtiene una imagen de gradiente correspondiente a una imagen de predicción a partir de una imagen de referencia (S105). Cada píxel de una imagen de gradiente tiene un valor de gradiente que indica un gradiente espacial de luminancia o crominancia. Un valor de gradiente se obtiene mediante la aplicación de un filtro de gradiente a una imagen de referencia. En la primera imagen de referencia 1100 de la FIG. 12, un filtro de gradiente para el primer bloque de predicción 1120 se refiere a píxeles en el primer rango de referencia de gradiente 1135, que incluye píxeles en el primer bloque de predicción 1120 y los píxeles circundantes. Dicho primer rango de referencia de gradiente 1135 se incluye en el primer rango de referencia de interpolación 1130. En la segunda imagen de referencia 1200, un filtro de gradiente se refiere a píxeles en el segundo rango de referencia de gradiente 1235, que incluye píxeles en el segundo bloque de predicción 1220 y los píxeles circundantes. Dicho segundo rango de referencia de gradiente 1235 se incluye en el segundo rango de referencia de interpolación 1230.

Cuando finaliza la obtención de una imagen de predicción y una imagen de gradiente a partir de cada una de la primera y segunda imágenes de referencia, finaliza el proceso de bucle realizado sobre una base por imagen de referencia (S106). Posteriormente, se realiza un proceso de bucle por subbloque, donde los subbloques se obtienen mediante la partición adicional de un bloque (S107-S110). Cada uno de los subbloques tiene un tamaño (por ejemplo, 4 X 4 píxeles) menor que el del bloque actual.

En el proceso de bucle realizado por subbloque, el valor de estimación de movimiento local 1300 de un subbloque se deriva utilizando la primera imagen de predicción 1140 y la primera imagen de gradiente 1150 que se obtienen de la primera imagen de referencia 1100, y la segunda imagen de predicción 1240 y la segunda imagen de gradiente 1250 que se obtienen de la segunda imagen de referencia 1200 (S108). Por ejemplo, un valor de estimación de movimiento local 1300 se deriva para un subbloque haciendo referencia a píxeles incluidos en un subbloque de predicción en cada una de la primera imagen de predicción 1140, la segunda imagen de predicción 1240, la primera imagen de gradiente 1150 y la segunda imagen de gradiente 1250. Un subbloque de predicción es una región del primer bloque de predicción 1120 o segundo bloque de predicción 1220 y corresponde a subbloques del bloque actual 1001. El valor de estimación de movimiento local también se denomina vector de compensación de movimiento en algunos casos.

Posteriormente, se genera la imagen de predicción final 1400 dividido en subbloques usando valores de píxel de la primera imagen de predicción 1140, valores de píxel de la segunda imagen de predicción 1240, valores de gradiente de la primera imagen de gradiente 1150, valores de gradiente de la segunda imagen de gradiente 1250, y valores de estimación de movimiento local 1300 (S109). Cuando finaliza la generación de una imagen de predicción final para cada uno de los subbloques incluidos en el bloque actual, se genera una imagen de predicción final para el bloque actual y finaliza el proceso de bucle realizado por subbloque (S110).

Además, cuando finaliza el proceso de bucle realizado por bloque (S111), finaliza el proceso ilustrado en la FIG. 11.

Cabe señalar que mediante la asignación directa de un vector de movimiento por bloque de un bloque actual a cada uno de los subbloques, también es posible obtener una imagen de predicción o una imagen de gradiente sobre una base por subbloque.

[Rangos de Referencia para el filtro de compensación de movimiento y el filtro de gradiente].

5 Aquí, se describirán los rangos de referencia para un filtro de compensación de movimiento y un filtro de gradiente.

La FIG. 13 es una vista conceptual para explicar ejemplos de rangos de referencia para un filtro de compensación de movimiento y un filtro de gradiente de acuerdo con la Realización 2.

En la FIG. 13, una pluralidad de círculos indica cada uno un píxel. El tamaño de un bloque actual se define, por ejemplo, como 8 X 8 píxeles, y el tamaño de un subbloque se define, por ejemplo, como 4 X 4 píxeles en la FIG. 13.

10 El rango de referencia 1131 indica un rango de referencia (por ejemplo, un rango rectangular de 8 X 8 píxeles) para un filtro de compensación de movimiento que se aplica al píxel superior izquierdo 1122 en el primer bloque de predicción 1120. El rango de referencia 1231 indica un rango de referencia (por ejemplo, un rango rectangular de 8 X 8 píxeles) para un filtro de compensación de movimiento que se aplica al píxel superior izquierdo 1222 en el segundo bloque de predicción 1220.

15 El rango de referencia 1132 indica un rango de referencia (por ejemplo, un rango rectangular de 6 X 6 píxeles) para un filtro de gradiente que se aplica al píxel superior izquierdo 1122 en el primer bloque de predicción 1120. El rango de referencia 1231 indica un rango de referencia (por ejemplo, un rango rectangular de 6 X 6 píxeles) para un filtro de gradiente que se aplica al píxel superior izquierdo 1222 en el segundo bloque de predicción 1220.

20 Un filtro de compensación de movimiento y un filtro de gradiente se aplican también a otros píxeles en el primer bloque de predicción 1120 o en el segundo bloque de predicción 1220, mientras que se hace referencia a píxeles en un rango de referencia. El rango de referencia tiene un tamaño idéntico al rango de referencia para el filtro de gradiente aplicado al píxel superior izquierdo 1122, y está situado en una posición correspondiente a la posición de cada uno de los píxeles. Como resultado, los píxeles en el primer rango de referencia de interpolación 1130 y el segundo rango de referencia de interpolación 1230 son referidos de manera que se obtienen la primera imagen de predicción 1140 y la segunda imagen de predicción 1240. Además, se refieren a los píxeles del primer rango de referencia de gradiente 1135 y del segundo rango de referencia de gradiente 1235 para obtener la primera imagen de gradiente 1150 y la segunda imagen de gradiente 1250.

25 [Efectos ventajosos, etc.]

30 Como se describió anteriormente, con el codificador y decodificador según la presente realización, es posible derivar un valor de estimación de movimiento local sobre una base por subbloque. En consecuencia, es posible reducir una carga de procesamiento o un tiempo de procesamiento más que en el caso de derivar un valor de estimación de movimiento local sobre una base por píxel, mientras se intenta reducir los errores de predicción utilizando valores de estimación de movimiento local derivados sobre una base por subbloque.

35 Con el codificador y el decodificador según la presente realización, es posible incluir un rango de referencia de interpolación en un rango de referencia normal. En consecuencia, no hay necesidad de cargar nuevos datos de píxel desde una memoria de trama para la compensación de movimiento. Por lo tanto, es posible inhibir un aumento de la capacidad de memoria y un ancho de banda de memoria en la generación de una imagen de predicción final utilizando valores de estimación de movimiento local derivados sobre una base por subbloque.

40 Con el codificador y el decodificador según la presente realización, es posible incluir un rango de referencia de gradiente en un rango de referencia de interpolación. Por consiguiente, no hay necesidad de cargar nuevos datos de píxel desde una memoria de trama para obtener una imagen de gradiente. Por lo tanto, es posible inhibir un aumento de la capacidad de memoria y un ancho de banda de memoria.

45 Este aspecto se puede implementar en combinación con uno o más de los otros aspectos de acuerdo con la presente divulgación. Además, parte del proceso del diagrama de flujo, parte de los elementos constitutivos de los aparatos y parte de la sintaxis descrita en este aspecto se pueden implementare en combinación con otros aspectos.

#### VARIACIÓN 1 DE LA REALIZACIÓN 2

50 A continuación, se describirá en detalle una variación del filtro de compensación de movimiento y del filtro de gradiente con referencia a los dibujos. Cabe señalar que en la siguiente variación 1, un proceso relacionado con una segunda imagen de predicción es similar a un proceso relacionado con una primera imagen de predicción. En consecuencia, la descripción duplicada se omite o se simplifica cuando es necesario.

[Filtro de compensación de movimiento]

En primer lugar, se describirá el filtro de compensación de movimiento. La FIG. 14 es una vista conceptual para explicar ejemplos de rangos de referencia para un filtro de compensación de movimiento de acuerdo con la Variación 1 de la Realización 2.

5 A continuación se describe, a modo de ejemplo, el caso en el que se aplica un filtro de compensación de movimiento al primer bloque de predicción 1120 para una exactitud de un cuarto de píxel en dirección horizontal y de medio píxel en dirección vertical. Un filtro de compensación de movimiento es un denominado filtro de 8 tomas y se expresa mediante la siguiente ecuación (3).

MAT. 3

$$I^k[x, y] = \sum_{j=0}^7 \left( w_{0,5}[j] \sum_{i=0}^7 w_{0,25}[i] I_0^k[x - 3 + i, y - 3 + j] \right)$$

10  $W_{0,25} = (-1,4,-10,58,17,-5,1,0)$

$w_{0,5} = (-1,4,-11,40,40,-11,4,-1)$

Aquí,  $I^k[x, y]$  indica un valor de píxel de la primera imagen de predicción con exactitud de subpíxel cuando  $k=0$ , e indica un valor de píxel de la segunda imagen de predicción con precisión de subpíxel cuando  $k=1$ . Un valor de píxel es un valor mantenido por un píxel y es un valor de luminancia o crominancia en una imagen de predicción, por ejemplo.

15  $w_{0,25}$  y  $w_{0,5}$  indican un factor de ponderación para la exactitud de un cuarto de píxel y un factor de ponderación para la exactitud de medio píxel, respectivamente.  $I_0^k[x, y]$  indica un valor de píxel de la primera imagen de predicción con exactitud de píxel entero cuando  $k=0$ , e indica un valor de píxel de la segunda imagen de predicción con exactitud de píxel entero cuando  $k=1$ .

20 En el caso de aplicar el filtro de compensación de movimiento como se expresa en la ecuación (3) al píxel superior izquierdo 1122 en la FIG. 14, los valores de los píxeles alineados en cada fila en una dirección horizontal en el rango de referencia 1131A se suman usando factores de ponderación, y el resultado de sumar los valores de los píxeles en filas plurales también se suman usando factores de ponderación.

25 De esta manera, un filtro de compensación de movimiento para el píxel superior izquierdo 1122 se refiere a píxeles en el rango de referencia 1131A en la presente variación. El rango de referencia 1131A es un rango rectangular definido para incluir tres columnas de píxeles a la izquierda del píxel superior izquierdo 1122, cuatro columnas de píxeles a la derecha del píxel superior izquierdo 1122, tres filas de píxeles por encima del píxel superior izquierdo 1122, y cuatro filas de píxeles por debajo del píxel superior izquierdo 1122.

30 Dicho filtro de compensación de movimiento se aplica a todos los píxeles en el primer bloque de predicción 1120. En consecuencia, los píxeles en el primer rango de referencia de interpolación 1130A son referidos por el filtro de compensación de movimiento para el primer bloque de predicción 1120.

Un filtro de compensación de movimiento es igualmente aplicado al segundo bloque de predicción 1220 como es aplicado al primer bloque de predicción 1120. En otras palabras, los píxeles en el rango de referencia 1231A son referidos para el píxel superior izquierdo 1222, y los píxeles en el segundo rango de referencia de interpolación 1230A son referidos para todo el segundo bloque de predicción 1220.

35 [Filtro del gradiente]

A continuación, se describirá un filtro de gradiente. La FIG. 15 es una vista conceptual para explicar ejemplos de rangos de referencia para un filtro de gradiente de acuerdo con la Variación 1 de la Realización 2.

El filtro de gradiente de acuerdo con la presente variación es un filtro denominado de 5 tomas y se expresa mediante las siguientes ecuaciones (4) y (5).

40 MAT. 4

$$I_x^k[x, y] = \sum_{i=0}^4 w[i] I_0^k[x - 2 + i, y] \quad (4)$$

MAT. 5

$$I_x^k[x, y] = \sum_{f=-1}^1 w[f] I_0^k[x, y - 2 + f] \quad (5)$$

$$w = (2, -9, 0, 9, -2)$$

Aquí,  $I_x^k[x, y]$  indica un valor de gradiente horizontal de cada píxel en una primera imagen de gradiente cuando  $k=0$ ,

e indica un valor de gradiente vertical de cada píxel en una segunda imagen de gradiente cuando  $k=1$ .  $I_y^k[x, y]$  indica un valor de gradiente horizontal de cada píxel en la primera imagen de gradiente cuando  $k=0$ , e indica un valor de gradiente vertical de cada píxel en la segunda imagen de gradiente cuando  $k=1$ .  $w$  indica un factor de ponderación.

En el caso de aplicar el filtro de gradiente según se expresa por las ecuaciones (4) y (5) al píxel superior izquierdo 1122 de la FIG. 15, se calcula un valor de gradiente horizontal mediante la suma, usando factores de ponderación, los valores de píxel de una imagen de predicción con exactitud de píxel entero que son los valores de cinco píxeles que incluyen el píxel superior izquierdo 1122 y están alineados en dirección horizontal. Además, se calcula un valor de gradiente vertical mediante la suma, usando los factores de ponderación, los valores de píxel de una imagen de predicción con exactitud de píxel entero que son los valores de cinco píxeles que incluyen el píxel superior izquierdo 1122 y están alineados en dirección vertical. En tal caso, con el píxel superior izquierdo 1122 que sirve como punto simétrico, los factores de ponderación de un píxel situado por encima del píxel superior izquierdo 1122 y un píxel situado por debajo del píxel superior izquierdo 1122 tienen el mismo valor con signos más/menos opuestos. Lo mismo se aplica a los factores de ponderación de un píxel situado a la izquierda del píxel superior izquierdo 1122 y un píxel situado a la derecha del píxel superior izquierdo 1122.

Así, el filtro de gradiente para el píxel superior izquierdo 1122 se refiere a píxeles en el rango de referencia 1132A en la presente variación. El rango de referencia 1132A tiene la forma de una cruz que se extiende en dos píxeles desde el píxel superior izquierdo 1122 en ambas direcciones arriba-abajo e izquierda-derecha.

Tal filtro de gradiente se aplica a todos los píxeles en el primer bloque de predicción 1120. En consecuencia, con el filtro de compensación de movimiento para el primer bloque de predicción 1120, se hace referencia a los píxeles en el primer rango de referencia de gradiente 1135A.

Al segundo bloque de predicción 1220 se le aplica un filtro de gradiente igual que al primer bloque de predicción 1120. En otras palabras, los píxeles en el rango de referencia 1231A se refieren al píxel superior izquierdo 1222, y los píxeles en el segundo rango de referencia de gradiente 1235A se refieren al segundo bloque de predicción 1220 entero.

Cabe señalar que cuando un vector de movimiento que especifica un rango de referencia indica una posición sub-píxel, los valores de píxel en los rangos de referencia 1132A y 1231A para un filtro de gradiente se pueden transformar en valores de píxel con precisión sub-píxel, y un filtro de gradiente se puede aplicar a los valores de píxel resultantes de la transformación. Alternativamente, un filtro de gradiente, que tiene, como valores de coeficiente, valores obtenidos por convolución de valores de coeficiente para la transformación en valores de píxel con exactitud de sub-píxel y valores de coeficiente para derivar un valor de gradiente, se puede aplicar a valores de píxel con exactitud de píxel entero. En este caso, un filtro de gradiente difiere de una posición de subpíxel a una posición de subpíxel.

[Derivación del valor de estimación de movimiento local por subbloque].

A continuación, se describirá la derivación de un valor de estimación de movimiento local por subbloque. Específicamente, lo siguiente describe un ejemplo de la derivación del valor de estimación de movimiento local de un subbloque superior izquierdo entre una pluralidad de subbloques incluidos en un bloque actual.

En la presente variación, el valor de estimación de movimiento local horizontal  $u$  y el valor de estimación de movimiento local vertical  $v$  de un subbloque se derivan sobre la base de las siguientes ecuaciones (6).

MATH. 6

$$\left. \begin{aligned} u &= \frac{sG_x dI}{sG_x^2} \\ v &= \frac{sG_y dI - u * sG_x G_y}{sG_y^2} \end{aligned} \right\} (6)$$

40

Aquí,  $sG_xG_y$ ,  $sG_x^2$ ,  $sG_y^2$ ,  $sG_xdl$ , y  $sG_ydl$  son valores calculados por subbloque y se calculan respectivamente, en función de las siguientes ecuaciones (7).

MAT. 7

$$\left. \begin{aligned} sG_xG_y &= \sum_{[i,j] \in \Omega} w[i,j] * G_x[i,j] * G_y[i,j] \\ sG_x^2 &= \sum_{[i,j] \in \Omega} w[i,j] * G_x[i,j] * G_x[i,j] \\ sG_y^2 &= \sum_{[i,j] \in \Omega} w[i,j] * G_y[i,j] * G_y[i,j] \\ sG_xdl &= \sum_{[i,j] \in \Omega} w[i,j] * G_x[i,j] * \Delta I[i,j] \\ sG_ydl &= \sum_{[i,j] \in \Omega} w[i,j] * G_y[i,j] * \Delta I[i,j] \end{aligned} \right\} (7)$$

5 Aquí,  $\Omega$  es un conjunto de coordenadas de todos los píxeles incluidos en un subbloque de predicción que es una región correspondiente a subbloques en un bloque de predicción.  $G_x[i, j]$  indica una suma de un valor de gradiente horizontal de la primera imagen de gradiente y un valor de gradiente horizontal de la segunda imagen de gradiente, mientras que  $G_y[i, j]$  indica una suma de un valor de gradiente vertical de la primera imagen de gradiente y un valor de gradiente vertical de la segunda imagen de gradiente.  $\Delta I[i, j]$  indica una diferencia entre la primera imagen de predicción y la segunda imagen de predicción.  $w[i, j]$  indica un factor de ponderación que depende de la posición de un píxel en un subbloque de predicción. Por ejemplo, se pueden utilizar factores de ponderación con valores idénticos para todos los píxeles de un subbloque de predicción.

Específicamente,  $G_x[i, j]$ ,  $G_y[i, j]$ , y  $\Delta I[i, j]$  se expresan mediante las siguientes ecuaciones (8).

MAT 8

$$\left. \begin{aligned} G_x[x, y] &= I_x^0[x, y] + I_x^1[x, y] \\ G_y[x, y] &= I_y^0[x, y] + I_y^1[x, y] \\ \Delta I[i, j] &= I^0[i, j] - I^1[i, j] \end{aligned} \right\} (8)$$

15 Como se ha descrito anteriormente, se calcula un valor de estimación de movimiento local por subbloque.

[Generación de la imagen de predicción final]

A continuación, se describirá la generación de una imagen de predicción final. Cada valor de píxel  $p[x, y]$  de una imagen de predicción final se calcula sobre la base en la siguiente ecuación (9) utilizando el valor de píxel  $I^0[x, y]$  de la primera imagen de predicción y el valor de píxel  $I^1[x, y]$  de la segunda imagen de predicción.

MAT 9

$$p[x, y] = (I^0[x, y] + I^1[x, y] + b[x, y]) \gg 1 \quad (9)$$

25 Aquí,  $b[x, y]$  denota un valor de corrección de cada píxel. En la ecuación (9), cada valor de píxel  $p[x, y]$  de una la imagen de predicción final se calcula mediante el desplazamiento a la derecha, 1 bit, una suma de valores de píxeles  $I^0[x, y]$  de la primera imagen de predicción, el valor del píxel  $I^1[x, y]$  de la segunda imagen de predicción y el valor de corrección  $b[x, y]$ . Cabe señalar que el valor de corrección  $b[x, y]$  se expresa mediante la siguiente ecuación (10).

MAT. 10

$$b[x, y] = u(I_x^0[x, y] - I_x^1[x, y]) + v(I_y^0[x, y] - I_y^1[x, y]) \quad (10)$$

30 En la ecuación (10), el valor de corrección  $b[x, y]$  se calcula mediante la suma del resultado de multiplicar una diferencia entre un valor de gradiente horizontal de la primera imagen de gradiente y un valor de gradiente horizontal de la segunda imagen de gradiente  $I_x^0[x, y] - I_x^1[x, y]$  con el valor de estimación del movimiento local horizontal (u), y un resultado de multiplicar una diferencia entre un valor de gradiente vertical de la primera imagen de gradiente

y un valor de gradiente vertical de la segunda imagen de gradiente  $I_2^g[x,y] - I_1^g[x,y]$  con el valor de estimación del movimiento local vertical (v).

5 Cabe señalar que las expresiones aritméticas descritas utilizando las ecuaciones (6) a (10) son ejemplos, y que se pueden utilizare expresiones aritméticas diferentes siempre que las expresiones produzcan los mismos efectos que los producidos con las ecuaciones (6) a (10).

[Efectos ventajosos, etc.]

10 Como se ha descrito anteriormente, es posible derivar un valor de estimación de movimiento local por subbloque incluso con el uso del filtro de compensación y el filtro de gradiente de acuerdo con la presente variación. Con la generación de una imagen de predicción final para un bloque actual, utilizando los valores de estimación de movimiento local así derivados sobre una base por subbloque, es posible obtener los mismos efectos ventajosos que los obtenidos con la Realización 2 descrita anteriormente.

Este aspecto se puede implementar en combinación con uno o más de los otros aspectos de acuerdo con la presente divulgación. Además, parte del proceso en el diagrama de flujo, parte de los elementos constitutivos de los aparatos, y parte de la sintaxis descrita en este aspecto se pueden implementar en combinación con otros aspectos.

15 VARIACIÓN 2 DE LA REALIZACIÓN 2

En la Realización 2 y Variación 1 descritas anteriormente, se hace referencia a todos los píxeles incluidos en un subbloque de predicción en un bloque de predicción correspondiente a subbloques en un bloque actual en la derivación de un valor de estimación de movimiento local, pero los ejemplos de realizaciones de la presente divulgación no se limitan a ello. Por ejemplo, solo se puede hacer referencia a algunos de una pluralidad de píxeles incluidos en un subbloque de predicción.

25 En vista de esto, la presente variación describe el caso en el que solo algunos de una pluralidad de píxeles incluidos en un subbloque de predicción son referidos en la derivación de un valor de estimación de movimiento local sobre una base por subbloque. En la ecuación (7) de la Variación 1 descrita anteriormente, por ejemplo, se utiliza un conjunto de coordenadas de algunos píxeles de un subbloque de predicción en lugar de  $\Omega$ , que es un conjunto de coordenadas de todos los píxeles incluidos en un subbloque de predicción. Se pueden utilizar varios patrones como conjunto de coordenadas de algunos de los píxeles incluidos en un subbloque de predicción.

30 La FIG. 16 ilustra un ejemplos de patrones de píxeles a los que se hace referencia en la derivación de un valor de estimación de movimiento local de acuerdo con la Variación 2 de la Realización 2. En La FIG. 16, los círculos sombreados en el subbloque de predicción 1121 o 1221 indican píxeles a los que se hace referencia mientras que los círculos que no están sombreados indican píxeles a los que no se hace referencia.

Cada uno de los siete patrones de píxeles (a) a (g) ilustra algunos de una pluralidad de píxeles incluidos en el subbloque de predicción 1121 o 1221. Los siete patrones de píxeles son mutuamente diferentes.

35 En (a) hasta (c) en la FIG. 16, se hace referencia a solo ocho píxeles entre dieciséis píxeles incluidos en el subbloque de predicción 1121 o 1221. En (d) hasta (g) en La FIG. 16, se hace referencia a solo cuatro píxeles entre dieciséis píxeles incluidos en el subbloque de predicción 1121 o 1221. En otras palabras, ocho de dieciséis píxeles son diluidos en (a) hasta (c) mientras que doce de dieciséis píxeles están reducidos en (d) hasta (g) en la FIG. 16.

40 Más específicamente, en (a) en la FIG. 16, se hace referencia a ocho píxeles, que están en ubicaciones escalonadas por un píxel en las direcciones horizontal y vertical. En (b) en La FIG. 16, un par izquierdo y un par derecho, cada uno que incluye dos píxeles alineados en una dirección horizontal, se ha referencia alternadamente en una dirección vertical. En (c) de la FIG. 16, se hace referencia a cuatro píxeles en el centro y cuatro píxeles en las cuatro esquinas del subbloque de predicción 1121 o 1221.

En (d) y (e) en la FIG. 16, se hace referencia a cada dos píxeles, uno en la primera columna y el otro en la tercera columna, desde la izquierda. En (f) de la FIG. 16, se hace referencia a cuatro píxeles situados en las cuatro esquinas. En (g) en la FIG. 16, se hace referencia a cuatro píxeles situados en el centro.

45 Un patrón de píxel se puede seleccionar adaptativamente de entre tales patrones de píxel predeterminados basados en dos imágenes de predicción. Por ejemplo, se puede seleccionar un patrón de píxel que incluye píxeles tantos como el número correspondiente a un valor de gradiente representativo de dos imágenes de predicción. Específicamente, se puede seleccionar un patrón de píxel que incluye cuatro píxeles (por ejemplo, cualquiera de (d) a (g)) en el caso donde un valor de gradiente representativo es menor que un umbral, y se puede seleccionar un patrón de píxel que incluye ocho píxeles (por ejemplo, cualquiera de (a) a (c)) en el caso contrario.

50 En el caso donde un patrón del píxel se selecciona de entre una pluralidad de patrones del píxel, el valor local de la estimación del movimiento de un subbloque se deriva con referencia a los píxeles en un subbloque de la predicción que son indicados por el patrón de píxel seleccionado.

Cabe señalar que la información que indica el patrón de píxeles seleccionado se puede escribir en un flujo de bits. En este caso, un decodificador puede obtener la información del flujo de bits y seleccionar un patrón de píxeles sobre la base de la información obtenida. La información que indica el patrón de píxeles seleccionado se puede escribir en una cabecera por bloque, corte, imagen o flujo.

5 Como se ha descrito anteriormente, con el codificador y decodificador de acuerdo con la presente realización, es posible derivar un valor de estimación de movimiento local por subbloque haciendo referencia solo a algunos de una pluralidad de píxeles incluidos en un subbloque de predicción. En consecuencia, es posible reducir una carga de procesamiento o un tiempo de procesamiento más que en el caso en el que se hace referencia a todos los píxeles.

10 Con el codificador y decodificador de acuerdo con la presente realización, es posible derivar un valor de estimación de movimiento local en una base por subbloque haciendo referencia solo a píxeles incluidos en un patrón de píxel seleccionado de entre una pluralidad de patrones de píxel. En consecuencia, es posible hacer referencia a los píxeles apropiados para la derivación de un valor de estimación de movimiento local de un subbloque, mediante el cambio entre la pluralidad de patrones de píxeles. Esto permite la reducción de errores de predicción.

15 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 procesos en los diagramas de flujo, parte de los elementos constitutivos de los aparatos, y parte de la sintaxis descrita en este aspecto se puede implementar en combinación con otros aspectos.

#### OTRA VARIACIÓN DE LA REALIZACIÓN 2

El codificador y el decodificador de acuerdo con uno o los aspectos plurales de la actual divulgación se han descrito sobre la base de las realizaciones y las variaciones.

20 Por ejemplo, el número de tomas para el filtro de compensación de movimiento de acuerdo con la realización 2 y la variación 1 de la misma descritas anteriormente es de ocho píxeles, pero no se limita a ello. El número de tomas para el filtro de compensación de movimiento puede ser cualquier número siempre que se incluya un rango de referencia de interpolación en un rango de referencia normal.

25 Cabe señalar que el número de tomas para el filtro de gradiente de acuerdo con la Realización 2 y la Variación 1 de la misma descritas anteriormente es de seis o cinco píxeles, pero no se limita a tales. El número de tomas para el filtro de gradiente puede ser cualquier número siempre que un rango de referencia de gradiente esté incluido en un rango de referencia de interpolación.

30 Cabe señalar también que el primer rango de referencia de interpolación y el segundo rango de referencia de interpolación están incluidos en el primer rango de referencia de gradiente y el segundo rango de referencia de gradiente, respectivamente, en la Realización 2 y Variación 1 anteriormente descritas, pero no se limitan a ello. Por ejemplo, el primer rango de referencia de gradiente puede coincidir con el primer rango de referencia de interpolación, y el segundo rango de referencia de gradiente puede coincidir con el segundo rango de referencia de interpolación.

35 Cuando un valor de estimación de movimiento local es derivado por subbloque, los valores de píxel en un subbloque de predicción se pueden ponderar para que los valores de píxeles más cercanos al centro del subbloque de predicción tengan más prioridad que otros píxeles. En otras palabras, en la derivación de un valor de estimación de movimiento local, los valores de una pluralidad de píxeles incluidos en un subbloque de predicción se pueden ponderar y usar en cada uno de los primer y segundo bloques de predicción. En ese caso, un factor de ponderación para un píxel, entre la pluralidad de píxeles, aumenta a medida que el píxel se sitúa más cerca del centro de un subbloque de predicción. Más específicamente, en la Variación 1 de la Realización 2, por ejemplo, el factor de ponderación  $w[i,j]$  en la ecuación (7) puede tener un valor mayor a medida que el valor de coordenadas de un píxel se sitúa más cerca del centro de un subbloque de predicción

45 Cuando se obtiene un valor de estimación de movimiento local por subbloque, también se puede hacer referencia a los píxeles de otro subbloque de predicción vecino que pertenece al mismo bloque de predicción. En otras palabras, en cada uno de los primer y segundo bloques de predicción, un valor de estimación de movimiento local se puede derivar en una base por subbloque refiriéndose no solo a píxeles incluidos en un subbloque de predicción sino también a píxeles incluidos en otro subbloque de predicción vecino al subbloque de predicción en el bloque de predicción.

Cabe señalar que los rangos de referencia para el filtro de compensación de movimiento y el filtro de gradiente de acuerdo con la Realización 2 y la Variación 1 de la misma descritas anteriormente son meras ilustraciones, y los rangos de referencia no necesitan limitarse a tales.

50 Además, se ilustran siete patrones de píxeles en la Variación 2 de la Realización 2 descrita anteriormente, pero el número de patrones de píxeles no se limita a tales. Por ejemplo, se pueden utilizar patrones de píxeles obtenidos rotando cada uno de los siete patrones de píxeles.

55 Cabe señalar también que los valores de los factores de ponderación según la Variación 1 de la Realización 2 descrita anteriormente son meros ejemplos y no se limitan a tales. Además, los tamaños de un bloque y un subbloque según la Realización 2 descrita anteriormente y cada una de las variaciones de la misma son meros ejemplos y no se limitan

a tamaños de 8 X 8 y 4 X 4 píxeles. Incluso con diferentes tamaños, es posible realizar la inter predicción de la misma manera que se describe en la Realización 2 descrita anteriormente y cada una de las variaciones de la misma.

5 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 procesos en los diagramas de flujo, parte de los elementos constituyentes de los aparatos, y parte de la sintaxis descrita en este aspecto se puede implementar en combinación con otros aspectos.

### REALIZACIÓN 3

10 Como se describe en cada una de las realizaciones anteriores, cada bloque funcional se puede realizar típicamente como por ejemplo, una MPU y una memoria. Además, los procesos realizados por cada uno de los bloques funcionales son típicamente realizados por una unidad de ejecución de programa, tal como un procesador, que lee y ejecuta el software (un programa) registrado en un medio de registro tal como ROM. El software se puede distribuir mediante, por ejemplo, descarga, y se puede registrar en un medio de registro como una memoria semiconductora y distribuir. Cabe señalar que cada bloque funcional, obviamente, se puede realizar también como hardware (circuito dedicado).

15 Además, el procesamiento descrito en cada una de las realizaciones se puede realizar mediante procesamiento integrado utilizando un único aparato (sistema) y, alternativamente, se puede realizar mediante procesamiento descentralizado usando una pluralidad de aparatos. Además, el procesador que ejecuta el programa descrito anteriormente puede ser un procesador único o una pluralidad de procesadores. En otras palabras, se puede realizar un procesamiento integrado y, alternativamente, se puede realizar un procesamiento descentralizado.

20 Las realizaciones de la presente divulgación no se limitan a los ejemplos de realizaciones anteriores; se pueden hacer varias modificaciones a los ejemplos de realizaciones, cuyos resultados también se incluyen dentro del alcance de las realizaciones de la presente divulgación.

25 A continuación, se describirán ejemplos de aplicación del método de codificación de imágenes en movimiento (método de codificación de imágenes) y del método de decodificación de imágenes en movimiento (método de decodificación de imágenes) descritos en cada una de las realizaciones anteriores y un sistema que emplea los mismos. El sistema se caracteriza por incluir un codificador de imágenes que emplea el método de codificación de imágenes, un decodificador de imágenes que emplea el método 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 caso por caso.

[Ejemplos de uso]

30 La FIG. 17 ilustra una configuración general del sistema de provisión de contenidos ex100 para implementar un servicio de distribución de contenidos. 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.

35 En el sistema de provisión de contenidos ex100, los dispositivos, que incluye computadora ex111, dispositivo de juego ex112, cámara ex113, electrodoméstico ex114 y teléfono inteligente ex115 están conectados a Internet ex101 a través del proveedor de servicios de Internet ex102 o red de comunicaciones ex104 y estaciones base ex106 a ex110. El sistema de provisión de contenidos 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 estaciones base ex106 a ex110, que son estaciones inalámbricas fijas. Además, el servidor de transmisión ex103 está conectado a dispositivos que incluyen el  
40 computadora ex111, el dispositivo de juego ex112, la cámara ex113, el electrodoméstico ex114 y el teléfono inteligente ex115 a través de, por ejemplo, Internet ex101. El servidor de transmisión ex103 también está conectado a, por ejemplo, un terminal en un punto de acceso en un avión ex117 vía satélite ex116.

45 Cabe señalar que, en lugar de estaciones base ex106 a ex110, se pueden utilizar puntos de acceso inalámbricos o puntos de acceso. El servidor de transmisión ex103 puede estar conectado a la red de comunicaciones ex104 directamente en lugar de a través de Internet ex101 o del proveedor de servicios de Internet ex102, y puede estar conectado al avión ex117 directamente en lugar de a través del satélite ex116.

50 La cámara ex113 es un dispositivo capaz de capturar imágenes fijas y video, tal como una cámara digital. El teléfono inteligente ex115 es un dispositivo de teléfono inteligente, un teléfono celular o un teléfono con sistema de teléfono personal (PHS) que puede funcionar con los estándares de sistemas de comunicaciones móviles de los sistemas típicos 2G, 3G, 3,9G y 4G, así como con el sistema 5G de próxima generación.

El electrodoméstico ex118 es, por ejemplo, un refrigerador o un dispositivo incluido en un sistema de cogeneración doméstico de pila de combustible.

55 En el sistema de provisión de contenido ex100, un terminal que incluye una función de captura de imagen y/o video es capaz de, por ejemplo, transmisión en directo mediante la conexión al servidor de transmisión ex103 a través de, por ejemplo, la estación base ex106. En la transmisión en directo, un terminal (por ejemplo, computadora ex111,

dispositivo de juego ex112, cámara ex113, electrodoméstico ex114, teléfono inteligente ex115, o avión ex117) realiza el procesamiento de codificación descrito en las realizaciones anteriores sobre el contenido de imagen fija o video capturado por un usuario a través del terminal, multiplexa los datos de video obtenidos mediante la codificación y los datos de audio obtenidos mediante la codificación del audio correspondiente al video, y transmite los datos obtenidos al servidor de transmisión ex103. En otras palabras, el terminal funciona como el codificador de imagen de acuerdo con un aspecto de la presente divulgación.

El servidor de transmisión ex103 transmite los datos de contenido transmitidos a los clientes que solicitan el flujo. Los ejemplos de cliente incluyen computadora ex111, dispositivo de juego ex112, cámara ex113, electrodoméstico ex114, teléfono inteligente ex115 y terminales en el interior 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 uno de los dispositivos funciona como decodificador de imágenes según un aspecto de la presente divulgación.

[Procesamiento descentralizado]

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

La descentralización no se limita solo a la división del procesamiento para la transmisión; la codificación de los datos capturados se puede dividir y realizar en los terminales, 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 detecta la complejidad de la imagen cuadro a cuadro o escena a escena, o detecta la carga de codificación. El segundo bucle es 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 la codificación del contenido haciendo que los terminales realicen el primer bucle de la codificación y que el servidor que recibió el contenido realice el segundo bucle de la codificación. En tal caso, después de recibir una solicitud de descodificación, 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 permite realizar una transmisión en tiempo real fluida.

En otro ejemplo, la cámara ex113 o similar extrae una cantidad de características de una imagen, comprime los 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 sobre la base de la cantidad de características y cambia la exactitud de cuantificación en consecuencia para realizar una compresión adecuada para el significado de la imagen. Los datos de cantidad de características son particularmente efectivos para mejorar la precisión y eficiencia de la predicción del vector de movimiento durante el segundo pase de compresión realizado por el servidor. Además, la codificación que tiene una carga de procesamiento relativamente baja, tal como la codificación de longitud variable (VLC), puede ser gestionada por el terminal, y la codificación que tiene una carga de procesamiento relativamente alta, tal como la codificación aritmética binaria adaptativa al contexto (CABAC), puede ser gestionada por el servidor.

En otro ejemplo, hay casos en los que una pluralidad de videos de aproximadamente la misma escena son capturados por una pluralidad de terminales en, por ejemplo, un estadio, un centro comercial o una fábrica. En tal caso, por ejemplo, la codificación se puede descentralizar mediante la división de las tareas de procesamiento entre la pluralidad de terminales que capturaron los videos y, si es necesario, otros terminales que no capturaron los videos y el servidor, en función de cada unidad. Las unidades pueden ser, por ejemplo, grupos de imágenes (GOP), imágenes o mosaicos resultantes de dividir una imagen. Esto permite reducir los tiempos de carga y lograr una transmisión más cercana al tiempo real.

Además, dado que los videos son aproximadamente de la misma escena, el servidor puede llevar a cabo la gestión y/o instrucción para que los videos capturados por los terminales puedan tener referencias cruzadas. Además, el servidor puede recibir datos codificados de los terminales, cambiar la relación de referencia entre los elementos de datos o corregir o reemplazar las imágenes mismas y, a continuación, realizar la codificación. Esto permite generar un flujo con mayor calidad y eficiencia para los elementos individuales de datos.

Además, el servidor puede transmitir datos de video después de realizar la transcodificación para convertir el formato de codificación de los datos de video. Por ejemplo, el servidor puede convertir el formato de codificación de MPEG a VP, y puede convertir H.264 a H.265.

De este modo, la codificación puede ser realizada por un terminal o uno o más servidores. En consecuencia, aunque el dispositivo que realiza la codificación se denomina "servidor" o "terminal" en la siguiente descripción, algunos o

todos los procesos realizados por el servidor pueden ser realizados por el terminal, y del mismo modo algunos o todos los procesos realizados por el terminal pueden ser realizados por el servidor. Esto también se aplica a los procesos de decodificación.

[3D, Multiángulo]

5 En los últimos años, ha aumentado el uso de imágenes o videos combinados de imágenes o videos de diferentes escenas capturadas simultáneamente o de la misma escena capturada desde diferentes ángulos por una pluralidad de terminales, como la cámara ex113 y/o el teléfono inteligente ex115. Los videos capturados por los terminales se combinan sobre la base, por ejemplo, de la relación posicional relativa obtenida por separado entre los terminales, o en regiones de un video que tienen puntos característicos coincidentes.

10 Además de la codificación de imágenes en movimiento bidimensionales, el servidor puede codificar una imagen fija basada en el análisis de escena de una imagen en movimiento automáticamente o en un punto de tiempo especificado por el usuario, y transmitir la imagen fija codificada a un terminal de recepción. Además, cuando el servidor puede obtener la relación posicional relativa entre los terminales de captura de video, además de imágenes en movimiento bidimensionales, el servidor puede generar geometría tridimensional de una escena basada en video de la misma escena capturada desde diferentes ángulos. Cabe señalar que el servidor puede codificar por separado datos tridimensionales generados a partir de, por ejemplo, una nube de puntos, y puede, sobre la base de un resultado de reconocimiento o seguimiento de una persona u objeto utilizando datos tridimensionales, seleccionar o reconstruir y generar un video que se transmitirá a un terminal de recepción a partir de videos capturados por una pluralidad de terminales.

20 Esto permite al usuario disfrutar de una escena seleccionando libremente videos correspondientes a los terminales de captura de video, y permite al usuario disfrutar del contenido obtenido extrayendo, a partir de datos tridimensionales reconstruidos a partir de una pluralidad de imágenes o videos, un video desde un punto de vista seleccionado. Además, de forma similar al video, el sonido se puede grabar desde ángulos relativamente diferentes, y el servidor puede multiplexar, con el video, el audio desde un ángulo o espacio específico de acuerdo con el video, y transmitir el resultado.

En los últimos años, también se ha popularizado el contenido que es un compuesto del mundo real y un mundo virtual, tal como el contenido de realidad virtual (VR) y realidad aumentada (AR). En el caso de las imágenes de VR, el servidor puede crear imágenes a partir de los puntos de vista de los ojos izquierdo y derecho y realizar una codificación que tolere la referencia entre las imágenes de los dos puntos 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.

30 En el caso de las imágenes de AR, el servidor superpone información del objeto virtual existente en un espacio virtual a información de la cámara que representa un espacio del mundo real, sobre la base de una posición o movimiento tridimensional desde la perspectiva del usuario. El decodificador puede obtener o almacenar información sobre objetos virtuales y datos tridimensionales, generar imágenes bidimensionales basadas en el movimiento desde la perspectiva del usuario y, a continuación, generar datos superpuestos mediante la conexión de imágenes sin interrupciones. 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 basados en datos tridimensionales almacenados en el servidor de acuerdo con el movimiento recibido, y codificar y transmitir los datos superpuestos generados al decodificador. Cabe señalar que los datos superpuestos incluyen, además de los valores RGB, un valor  $\alpha$  que indica la transparencia, y el servidor establece el valor  $\alpha$  para las secciones distintas del objeto generado a partir de los 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, tal como un croma, y generar datos en los que las áreas distintas del objeto se establezcan como fondo.

La decodificación de los datos transmitidos de forma similar puede ser realizada por el cliente (es decir, los terminales), por el lado del servidor, o dividida entre ambos. En un ejemplo, un terminal puede transmitir una solicitud de recepción a un servidor, el contenido solicitado puede ser recibido y decodificado por otro terminal, y una señal descodificada se puede transmitir a un dispositivo que tenga una pantalla. Es posible reproducir datos de alta calidad de imagen mediante la descentralización del procesamiento y la selección adecuada del contenido independientemente de la capacidad de procesamiento del propio terminal de comunicaciones. En otro ejemplo, mientras un televisor, por ejemplo, recibe datos de imagen de gran tamaño, una región de una imagen, tal como un mosaico obtenido dividiendo la imagen, se puede decodificar y visualizar en un terminal o terminales personales de un espectador o espectadores del televisor. Esto permite a los espectadores compartir una visión de gran tamaño, así como a cada espectador comprobar su área asignada o inspeccionar una región con más detalle.

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 contenidos sin interrupciones incluso al cambiar a datos apropiados para la conexión actual, utilizando un estándar de sistema de transmisión tal como MPEG-

DASH. De este modo, el usuario puede cambiar de 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. Además, sobre la base de, por ejemplo, la información sobre la posición del usuario, la descodificación se puede realizar mientras se cambia qué terminal se encarga de la descodificación y qué terminal se encarga de 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 esté incrustado un dispositivo capaz de mostrar contenidos o en parte del suelo, información de mapeo mientras se está en movimiento. Además, también es posible cambiar la tasa de bit de los datos recibidos en función de la accesibilidad a los datos codificados en una red, tal como cuando los datos codificados se almacenan en caché en un servidor rápidamente accesible desde el terminal de recepción o cuando los datos codificados se copian en un servidor de borde en un servicio de provisión de contenidos.

[Codificación escalable]

El cambio de contenido se describirá con referencia a un flujo escalable, ilustrado en La FIG. 18, que está codificado por compresión mediante la implementación del método de codificación de imágenes en movimiento descrito en las realizaciones anteriores. El servidor puede tener una configuración en la que el contenido se cambia haciendo uso de la escalabilidad temporal y/o espacial de un flujo, que se consigue mediante la división y codificación de las capas, tal como se ilustra en la FIG. 18. Cabe señalar que puede haber una pluralidad de flujos individuales que tengan el mismo contenido pero diferente calidad. En otras palabras, al determinar hasta qué capa decodificar sobre la base de factores internos, tales como la capacidad de procesamiento en el lado del decodificador, y factores externos, tales como el ancho de banda de comunicación, el lado del decodificador puede cambiar libremente entre contenido de baja resolución y contenido de alta resolución mientras decodifica. Por ejemplo, en el caso de que el usuario quiera seguir viendo en casa, en un dispositivo tal como un televisor conectado a Internet, un video que había estado viendo previamente en el teléfono inteligente ex115 mientras se desplazaba, el dispositivo puede simplemente descodificar el mismo flujo hasta una capa diferente, lo que reduce la carga del lado del servidor.

Por otra parte, además de la configuración descrita anteriormente en la que la escalabilidad se logra como resultado de las imágenes que se codifican por capa y la capa de mejora está por encima de la capa de base, la capa de mejora puede incluir metadatos basados en, por ejemplo, información estadística sobre la imagen, y el lado del decodificador puede generar contenido de alta calidad de imagen mediante la realización de imágenes de superresolución en una imagen en la capa de base sobre la base de los metadatos. Las imágenes de superresolución pueden mejorar la relación SN manteniendo la resolución y/o aumentando la resolución. Los metadatos incluyen información para identificar un coeficiente de filtro lineal o no lineal utilizado en el procesamiento de superresolución, o información que identifica un valor de parámetro en el procesamiento de filtro, aprendizaje automático o método de mínimos cuadrados utilizado en el procesamiento de superresolución.

Alternativamente, también es aceptable una configuración en la que una imagen 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 mediante la selección de mosaico para decodificar. Además, al almacenar un atributo sobre el objeto (persona, auto, pelota, etc.) y una posición del objeto en el video (coordenadas en imágenes idénticas) como metadatos, el lado del decodificador puede identificar la posición de un objeto deseado sobre la base de los metadatos y determinar qué mosaico o mosaicos incluyen dicho objeto. Por ejemplo, como se ilustra en la FIG. 19, los metadatos se almacenan utilizando una estructura de almacenamiento de datos diferente de los datos de píxeles, como un mensaje SEI en HEVC. Estos metadatos indican, por ejemplo, la posición, el tamaño o el color del objeto principal.

Además, los metadatos se pueden almacenar en unidades de una pluralidad de imágenes, tales como unidades de flujo, secuencia o acceso aleatorio. De este modo, el decodificador puede obtener, por ejemplo, el momento en que aparece una persona específica en el video y, mediante el ajuste de la información de la unidad de imagen, puede identificar una imagen en la que está presente el objeto y la posición del objeto en la imagen.

[Optimización de páginas web]

La FIG. 20 ilustra un ejemplo de pantalla de visualización de una página web en, por ejemplo, computadora ex111. La FIG. 21 ilustra un ejemplo de pantalla de visualización de una página web en, por ejemplo, teléfono inteligente ex115. Como se ilustra en la FIG. 20 y la FIG. 21, una página web puede incluir una pluralidad de enlaces de imagen que son enlaces a contenido de imagen, y el aspecto de la página web difiere de acuerdo con el dispositivo utilizado para ver la página web. Cuando una pluralidad de enlaces de imagen son visibles en la pantalla, hasta que el usuario selecciona explícitamente un enlace de imagen, o hasta que el enlace de imagen está en el centro aproximado de la pantalla o el enlace de imagen entero cabe en la pantalla, el aparato de visualización (decodificador) muestra, como enlaces de imagen, imágenes fijas incluidas en el contenido o imágenes I, muestra video tal como un gif animado utilizando una pluralidad de imágenes fijas o imágenes I, por ejemplo, o recibe solo la capa base y decodifica y muestra el video.

Cuando el usuario selecciona un enlace de imagen, el aparato de visualización descodifica dando la máxima prioridad a la capa base. Cabe señalar que si hay información en el código HTML de la página web que indica que el contenido es escalable, el aparato de visualización puede descodificar 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 muy limitado, el aparato de visualización puede reducir el retardo entre el punto de tiempo en que se descodifica la imagen guía y el momento

5 en que se muestra la imagen descodificada (es decir, el retardo entre el inicio de la descodificación del contenido y la visualización del contenido) mediante la decodificación y solo muestra imágenes de referencia hacia delante (imagen I, imagen P, imagen de referencia hacia delante B). Además, el aparato de visualización puede ignorar deliberadamente la relación de referencia entre imágenes y descodificar groseramente todas las imágenes B y P como imágenes de referencia hacia adelante, y luego realizar una descodificación normal a medida que aumenta el número de imágenes recibidas.

[Conducción autónoma]

10 Al transmitir y recibir datos de imagen fija o de video, como información de mapeo bidimensional o tridimensional para la conducción autónoma o asistida de un automóvil, el terminal de recepción puede recibir, además de los datos de imagen pertenecientes a una o más capas, información sobre, por ejemplo, el tiempo o la construcción de la carretera como metadatos, y asociar los metadatos con los datos de imagen después de la descodificación. Cabe señalar que los metadatos pueden ser asignados por capa y, alternativamente, simplemente se pueden multiplexar con los datos de imagen.

15 En tal caso, debido a que el automóvil, dron, avión, etc., que incluye el terminal de recepción es móvil, el terminal de recepción puede recibir y decodificar sin interrupciones mientras cambia entre estaciones base entre las estaciones base ex106 a ex110 mediante la transmisión de la información que indica la posición del terminal de recepción al solicitar la recepción. Además, 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 por ejemplo, en qué medida se reciben los metadatos o en qué medida se actualiza la información de mapeo.

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

[Transmisión de Contenido Individual]

25 En el sistema de provisión de contenidos ex100, además de contenidos largos de alta calidad de imagen distribuidos por una entidad de distribución de video, también es posible la transmisión unidifusión o multidifusión de contenidos cortos de baja calidad de imagen procedentes de un individuo. Además, es probable que la popularidad de este tipo de contenidos procedentes de particulares siga aumentando. El servidor puede realizar primero un procesamiento de edición del contenido antes del procesamiento de codificación para refinar el contenido individual. Esto se puede lograr, por ejemplo, con la siguiente configuración.

30 En tiempo real, mientras se captura el contenido de video o imagen o después de que el contenido haya sido capturado y acumulado, el servidor realiza un procesamiento de reconocimiento basado en los datos en bruto o codificados, tal como procesamiento de errores de captura, procesamiento de búsqueda de escenas, análisis del significado y/o procesamiento de detección de objetos. A continuación, sobre la base del resultado del procesamiento de reconocimiento, el servidor -ya sea cuando se le solicite o automáticamente- edita el contenido, cuyos ejemplos incluyen: correcciones tales como enfoque y/o corrección de desenfoque por movimiento; eliminación de escenas de baja prioridad tales como escenas con poco brillo en comparación con otras imágenes o desenfocadas; ajuste de los bordes de los objetos; y ajuste del tono de color. El servidor codifica los datos editados sobre la base del resultado de la edición. Se sabe que los videos excesivamente largos tienden a recibir menos visualizaciones. En consecuencia, para mantener el contenido dentro de una longitud específica que se ajuste a la longitud del video original, el servidor puede, además de las escenas de baja prioridad descritas anteriormente, recortar automáticamente las escenas con poco movimiento sobre la base del resultado del procesamiento de la imagen. Alternativamente, el servidor puede generar y codificar un resumen de video basado en el resultado de un análisis del significado de una escena.

45 Tenga en cuenta que hay casos en los que los contenidos individuales pueden incluir contenidos que infrinjan derechos de autor, derechos morales, 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. En consecuencia, antes de la codificación, el servidor puede, por ejemplo, editar las imágenes para difuminar las caras de las personas en la periferia de la pantalla o por ejemplo difuminar el interior de una casa. Además, el servidor puede estar configurado para reconocer los rostros de personas que no sean una persona registrada en las imágenes que se van a codificar, y cuando dichos rostros aparezcan en una imagen, por ejemplo, aplicar un filtro de mosaico al rostro de la persona. Alternativamente, como pre- o pos-procesamiento para la codificación, el usuario puede especificar, por motivos de derechos de autor, que una región de una imagen que incluya una persona o una región del fondo sea procesada, y el servidor puede procesar la región especificada, por ejemplo, mediante el reemplazo de la región por una imagen diferente o difuminar la región. Si la región incluye una persona, la persona puede ser rastreada en la imagen en movimiento y la región de la cabeza se puede reemplazar por otra imagen a medida que la persona se mueve.

55 Además, debido a que existe una demanda de visualización en tiempo real de contenidos producidos por individuos, cuyo tamaño de datos tiende a ser pequeño, el descodificador recibe primero la capa base como máxima prioridad y realiza la descodificación y reproducción, aunque esto puede variar en función del ancho de banda. Cuando el contenido se reproduce dos o más veces, tal como cuando el descodificador recibe la capa de mejora durante la

descodificación y reproducción de la capa base y realiza un bucle de reproducción, el descodificador puede reproducir un video de alta calidad de imagen que incluye la capa de mejora. Si el flujo se codifica utilizando dicha codificación escalable, el video puede ser de baja calidad cuando se encuentra en un estado no seleccionado o al principio del video, pero puede ofrecer una experiencia en la que la calidad de imagen del flujo aumente progresivamente de forma inteligente. Esto no se limita solo a la codificación escalable; la misma experiencia se puede ofrecer al configurar un único 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 descodificación pueden ser realizadas por la LSI ex500, que normalmente está incluida en cada terminal. El LSI ex500 puede estar configurado en un único chip o en una pluralidad de chips. El software para la codificación y descodificación de imágenes en movimiento se pueden integrar en algún tipo de medio de registro (tal como un CD-ROM, un disco flexible o un disco duro) que sea legible, por ejemplo, por la computadora ex111, y la codificación y descodificación se puede realizar utilizando el software. Además, cuando el teléfono inteligente ex115 está equipado con una cámara, se pueden transmitir por los datos de video obtenidos por la cámara. En este caso, los datos de video son codificados por el LSI ex500 incluido en el teléfono inteligente ex115.

Cabe señalar que la LSI ex500 puede estar configurada para descargar y activar una aplicación. En tal caso, el terminal determina en primer lugar 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 descarga primero un códec o un software de aplicación y, a continuación, obtiene y reproduce el contenido.

Aparte del ejemplo de sistema de provisión de contenido ex100 que utiliza Internet ex101, al menos el codificador de imágenes en movimiento (codificador de imágenes) o el decodificador de imágenes en movimiento (decodificador de imágenes) descritos en las realizaciones anteriores se pueden implementar en un sistema de difusión 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 video multiplexados utilizando, por ejemplo, un satélite, aunque esto está orientado a la multidifusión, mientras que la unidifusión es más fácil con el sistema de provisión de contenidos ex100.

[Configuración de hardware]

La FIG. 22 ilustra un teléfono inteligente ex115. La FIG. 23 ilustra un ejemplo de configuración de 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 video e imágenes fijas, y una pantalla ex458 que muestra datos decodificados, tales como el video capturado por la cámara ex465 y el video 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 la salida de voz u otro audio, una unidad de entrada de audio ex456 tal como un micrófono para la entrada de audio, una memoria ex467 capaz de almacenar datos descodificados tales como video capturado o imágenes fijas, audio registrado, imágenes de video o fijas, y correo, así como datos descodificados, y una ranura ex464 que es una interfaz para SIM ex468 para autorizar el acceso a una red y a diversos datos. Cabe señalar que se puede utilizar una memoria externa en lugar de la memoria ex467.

Además, el controlador principal ex460, que controla exhaustivamente la pantalla ex458 y la interfaz de usuario ex466, el circuito de alimentación ex461, el controlador de entrada de la interfaz de usuario ex462, el procesador de señal de video 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, la ranura ex464 y la memoria ex467 están conectados a través del bus ex470.

Cuando el usuario enciende el botón de encendido del circuito de alimentación ex461, el teléfono inteligente ex115 se enciende en un estado operable mediante el suministro de energía a cada componente desde un paquete de baterías.

El teléfono inteligente ex115 realiza el procesamiento para, por ejemplo, llamar y transmitir datos, sobre la base del control realizado por el controlador principal ex460, que incluye una CPU, una ROM y una RAM. Cuando se efectúan 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 el procesamiento de espectro ensanchado por el modulador/demodulador ex452 y conversión digital-analógica y procesamiento de 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 en analógico-digital, se procesan con espectro ensanchado inverso mediante el modulador/demodulador ex452, se convierten en una señal de audio analógica mediante el procesador de señal de audio ex454 y, a continuación, se emiten desde la unidad de salida de audio ex457. En el modo de transmisión de datos, los datos de texto, imagen fija o video son transmitidos 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 similar de transmisión y recepción. En el modo de transmisión de datos, cuando se envía un video, una imagen fija, o video y audio, el procesador de señal de video ex455 codifica por compresión, mediante el método de codificación de imagen en movimiento descrito en las realizaciones anteriores,

una señal de video almacenada en la memoria ex467 o una señal de video introducida desde la cámara ex465, y transmite los datos de video codificados al multiplexor/demultiplexor ex453. Además, el procesador de señal de audio ex454 codifica una señal de audio grabada por la unidad de entrada de audio ex456 mientras la cámara ex465 está capturando, por ejemplo, un video o una imagen fija, y transmite los datos de audio codificados al multiplexor/demultiplexor ex453. El multiplexor/demultiplexor ex453 multiplexa los datos de video 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.

Cuando se recibe un video anexo en un correo electrónico o un chat, o un video enlazado 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 dividir los datos multiplexados en un flujo de bits de datos de video y un flujo de bits de datos de audio, suministra los datos de video codificados al procesador de señal de video ex455 a través del bus síncrono ex470, y suministra los datos de audio codificados al procesador de señal de audio ex454 a través del bus síncrono ex470. El procesador de señales de video ex455 descodifica la señal de video utilizando un método de descodificación de imágenes en movimiento correspondiente al método de codificación de imágenes en movimiento descrito en las realizaciones anteriores, y el video o una imagen fija incluida en el archivo de imágenes en movimiento vinculado se visualiza en la pantalla ex458 a través del controlador de pantalla ex459. Además, el procesador de señal de audio ex454 decodifica la señal de audio y emite audio desde la unidad de salida de audio ex457. Cabe señalar que, dado que la transmisión en tiempo real es cada vez más popular, hay casos en los que la reproducción del audio puede ser socialmente inapropiada de acuerdo con el entorno del usuario. En consecuencia, como valor inicial, es preferible una configuración en la que solo se reproducen datos de video, es decir, que no se reproduzca la señal de audio. El audio puede ser sincronizado y reproducido, solo cuando se recibe una entrada, tal como cuando el usuario hace clic en los datos de video.

Aunque en el ejemplo anterior se utilizó el teléfono inteligente ex115, son concebibles tres implementaciones: un terminal transceptor que incluye tanto un codificador como un decodificador; un terminal transmisor que incluye solo un codificador; y un terminal receptor que incluya solo un decodificador. Además, en la descripción del sistema de la difusión digital, se da un ejemplo en el que se reciben o transmiten datos multiplexados obtenidos como resultado de la multiplexación de datos de video con, por ejemplo, datos de audio, pero los datos multiplexados pueden ser datos de video multiplexados con datos distintos de los datos de audio, tales como datos de texto relacionados con el video. Además, se pueden recibir o transmitir los propios datos de video en lugar de los datos multiplexados. |

Aunque se describe que el controlador principal ex460, que incluye una CPU, controla los procesos de codificación o descodificación, los terminales a menudo incluyen una GPU. En consecuencia, es aceptable una configuración en la que un área grande se procesa 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 gestiona para permitir el uso común por la CPU y la GPU. Esto permite acortar el tiempo de codificación, mantener la naturaleza en tiempo real del flujo y reducir el retardo. En particular, el procesamiento relativo a la estimación del movimiento, el filtrado de desbloqueo, el desplazamiento adaptativo de muestras (SAO) y la transformación/cuantificación puede ser realizada eficazmente por la GPU en lugar de por la CPU en unidades de, por ejemplo, imágenes, todo a la vez.

#### APLICABILIDAD INDUSTRIAL

La presente divulgación es aplicable, por ejemplo, a receptores de televisión, grabadoras de video digital, sistemas de navegación para automóviles, teléfonos móviles, cámaras digitales y videocámaras digitales.

#### MARCAS DE REFERENCIA EN LOS DIBUJOS

00	codificador
102	divisor
104	sustractor
106	transformador
108	cuantificador
110	codificador de entropía
112, 204	cuantificador inverso
114, 206	transformador inverso
116, 208	sumador
118, 210	memoria de bloque

## ES 2 993 476 T3

120, 212	filtro de bucle
122, 214	memoria de trama
124, 216	Intra predictor
126, 218	Inter predictor
128, 220	controlador de predicción
200	decodificador
202	decodificador de entropía
1000	imagen actual
1001	bloque actual
1100	primera imagen de referencia
1110	primer vector de movimiento
1120	primer bloque de predicción
1121, 1221	subbloque de predicción
1122, 1222	píxel superior izquierdo
1130, 1130A	primer rango de referencia de interpolación
1131, 1131A, 1132, 1132A,	1231, 1231A, 1232, 1232A rango de referencia
1135, 1135A	primer rango de referencia de gradiente
1140	primera imagen de predicción
1150	Primera imagen de gradiente
1200	segunda imagen de referencia
1210	segundo vector de movimiento
1220	segundo bloque de predicción
1230, 1230A	segundo rango de referencia de interpolación
1235, 1235A	Segundo rango de referencia de gradiente
1240	segunda imagen de predicción
1250	segunda imagen de gradiente
1300	valor de estimación del movimiento local
1400	imagen de predicción final

## REIVINDICACIONES

1. Un codificador que codifica un bloque actual (100, 1001) en una imagen actual (1000), el codificador que comprende:

un procesador, y memoria, donde

5 usando la memoria, el procesador está configurado para:

derivar dos vectores de movimiento (1110, 1210) para el bloque actual (100, 1001) sobre la base de dos imágenes de referencia (1100, 1200) para obtener dos imágenes de predicción (1140, 1240);

10 obtener dos imágenes de predicción (1140, 1240) con exactitud de subpíxel haciendo referencia a las dos imágenes de referencia (1100, 1200) mediante la compensación de movimiento utilizando los dos vectores de movimiento (1110, 1210), en la que un píxel de píxel en cada una de las dos imágenes de predicción (1140, 1240) se interpola con exactitud de subpíxel haciendo referencia a píxeles en un rango de referencia de interpolación (1130, 1230) en las imágenes de referencia (1100, 1200);

15 obtener un valor de gradiente para cada uno de los píxeles de las dos imágenes de predicción (1140, 1240) utilizando un valor de píxel de cada uno de los píxeles en un rango de referencia de gradiente (1135, 1235), en el que el rango de referencia de gradiente (1135, 1235) incluye píxeles en un bloque de predicción (1120, 1220) correspondiente al bloque actual (100, 1001) y píxeles circundantes;

derivar un valor de estimación de movimiento local (1300) para cada uno de los subbloques basado en el valor del píxel y el valor del gradiente de cada uno de los píxeles de las dos imágenes de predicción (1140, 1240), se obtienen los subbloques mediante la partición del bloque actual (100, 1001);

20 generar una imagen de predicción final (1400) para el bloque actual (100,1001) utilizando el valor del píxel y el valor del gradiente de cada uno de los píxeles de las dos imágenes de predicción (1140, 1240), y el valor de estimación del movimiento local (1300) derivado para cada uno de los subbloques; y

codificar el bloque actual (100, 1001) en la imagen actual (1000) utilizando la imagen de predicción final (1400),

25 **caracterizado porque**

el rango de referencia de interpolación (1130, 1230) coincide con un rango de referencia normal al que se hace referencia para la compensación de movimiento para el bloque actual (100, 1001) en la inter predicción normal, en la que, en la inter predicción normal, se deriva un vector de movimiento por bloque mediante la estimación de movimiento, se realiza una compensación de movimiento por bloque para obtener dos imágenes de predicción que se interpolan con una exactitud sub-píxel utilizando el vector de movimiento derivado y sin utilizar el valor de estimación de movimiento local (1300), y se utiliza directamente una imagen con compensación de movimiento como imagen de predicción final.

2. Un decodificador que decodifica un bloque actual (100, 1001) en una imagen actual (1000), el decodificador que comprende:

35 un procesador; y

memoria, donde

usando la memoria, el procesador está configurado para:

derivar dos vectores de movimiento (1110, 1210) para el bloque actual (100, 1001) sobre la base de dos imágenes de referencia (1100, 1200) para obtener dos imágenes de predicción (1140, 1240);

40 obtener dos imágenes de predicción (1140, 1240) con exactitud subpíxel haciendo referencia a las dos imágenes de referencia (1100, 1200) mediante la realización de una compensación de movimiento utilizando los dos vectores de movimiento (1110, 1210), en la que un píxel de cada una de las dos imágenes de predicción (1140, 1240) se interpola con exactitud subpíxel haciendo referencia a píxeles de un rango de referencia de interpolación (1130, 1230) en las imágenes de referencia (1100, 1200);

45 obtener un valor de gradiente para cada uno de los píxeles de las dos imágenes de predicción (1140, 1240), utilizando un valor de píxel de cada uno de los píxeles en un rango de referencia de gradiente (1135, 1235), en el que el rango de referencia de gradiente (1135, 1235) incluye píxeles en un bloque de predicción (1120, 1220) correspondiente al bloque actual (100, 1001) y píxeles circundantes;

derivar un valor de estimación de movimiento local (1300) para cada uno de los subbloques sobre la base del valor del píxel y el valor del gradiente de cada uno de los píxeles de las dos imágenes de predicción (1140, 1240), los subbloques se obtienen mediante la partición del bloque actual (100, 1001);

5 generar una imagen de predicción final (1400) para el bloque actual (100,1001) utilizando el valor del píxel y el valor del gradiente para cada uno de los píxeles de las dos imágenes de predicción (1140, 1240), y el valor de estimación de movimiento local (1300) derivado para cada uno de los subbloques; y

decodificar el bloque actual (100, 1001) en la imagen actual (1000) utilizando la imagen de predicción final (1400),

10 **caracterizado porque**

el rango de referencia de interpolación (1130, 1230) coincide con un rango de referencia normal al que se hace referencia para la compensación de movimiento para el bloque actual (100, 1001) en la inter predicción normal, en la que, en la inter predicción normal, se obtiene un vector de movimiento por bloque a partir del flujo de bits, se realiza una compensación de movimiento por bloque para obtener dos imágenes de predicción que se interpolan con una exactitud sub-píxel utilizando el vector de movimiento derivado y sin utilizar el valor de estimación de movimiento local (1300), y se utiliza directamente una imagen con compensación de movimiento como imagen de predicción final.

3. Un programa para hacer que una computadora ejecute:

20 derivar dos vectores de movimiento (1110, 1210) para el bloque actual (100, 1001) a partir de dos imágenes de referencia (1100, 1200) para obtener dos imágenes de predicción (1140, 1240);

25 obtener dos imágenes de predicción (1140, 1240) con exactitud subpíxel haciendo referencia a las dos imágenes de referencia (1100, 1200) mediante la realización de una compensación de movimiento utilizando los dos vectores de movimiento (1110, 1210), en la que un píxel de cada una de las dos imágenes de predicción (1140, 1240) se interpola con exactitud subpíxel haciendo referencia a píxeles de un rango de referencia de interpolación (1130, 1230) en las imágenes de referencia (1100, 1200);

obtener dos imágenes de gradiente (1150, 1250) mediante la aplicación de un filtro de gradiente a las imágenes de referencia (1100, 1200) en un rango de referencia de gradiente (1135, 1235), en el que se obtiene un valor de gradiente para cada uno de los píxeles incluidos en las dos imágenes de predicción (1140, 1240);

30 derivar un valor de estimación de movimiento local (1300) para cada uno de los subbloques sobre la base del valor del píxel y el valor del gradiente de cada uno de los píxeles de las dos imágenes de predicción (1140, 1240), los subbloques se obtienen mediante la partición del bloque actual (100, 1001);

35 generar una imagen de predicción final (1400) para el bloque actual (100,1001) utilizando el valor del píxel y el valor del gradiente para cada uno de los píxeles de las dos imágenes de predicción (1140, 1240), y el valor de estimación de movimiento local (1300) derivado para cada uno de los subbloques; y

codificar o decodificar el bloque actual (100, 1001) en la imagen actual (1000) utilizando la imagen de predicción final (1400),

**caracterizado porque**

40 el rango de referencia de interpolación (1130, 1230) coincide con un rango de referencia normal al que se hace referencia para la compensación de movimiento para el bloque actual (100, 1001) en la inter predicción normal, en la que, en la inter predicción normal, se deriva un vector de movimiento por bloque mediante la estimación del movimiento u obtiene por bloques a partir de un flujo de bits, la compensación del movimiento se realiza por bloques para obtener dos imágenes de predicción que se interpolan con exactitud de subpíxel utilizando el vector de movimiento derivado y sin utilizar el valor de estimación del movimiento local (1300), y una imagen compensada por el movimiento se utiliza directamente como imagen de predicción final.

FIG. 1

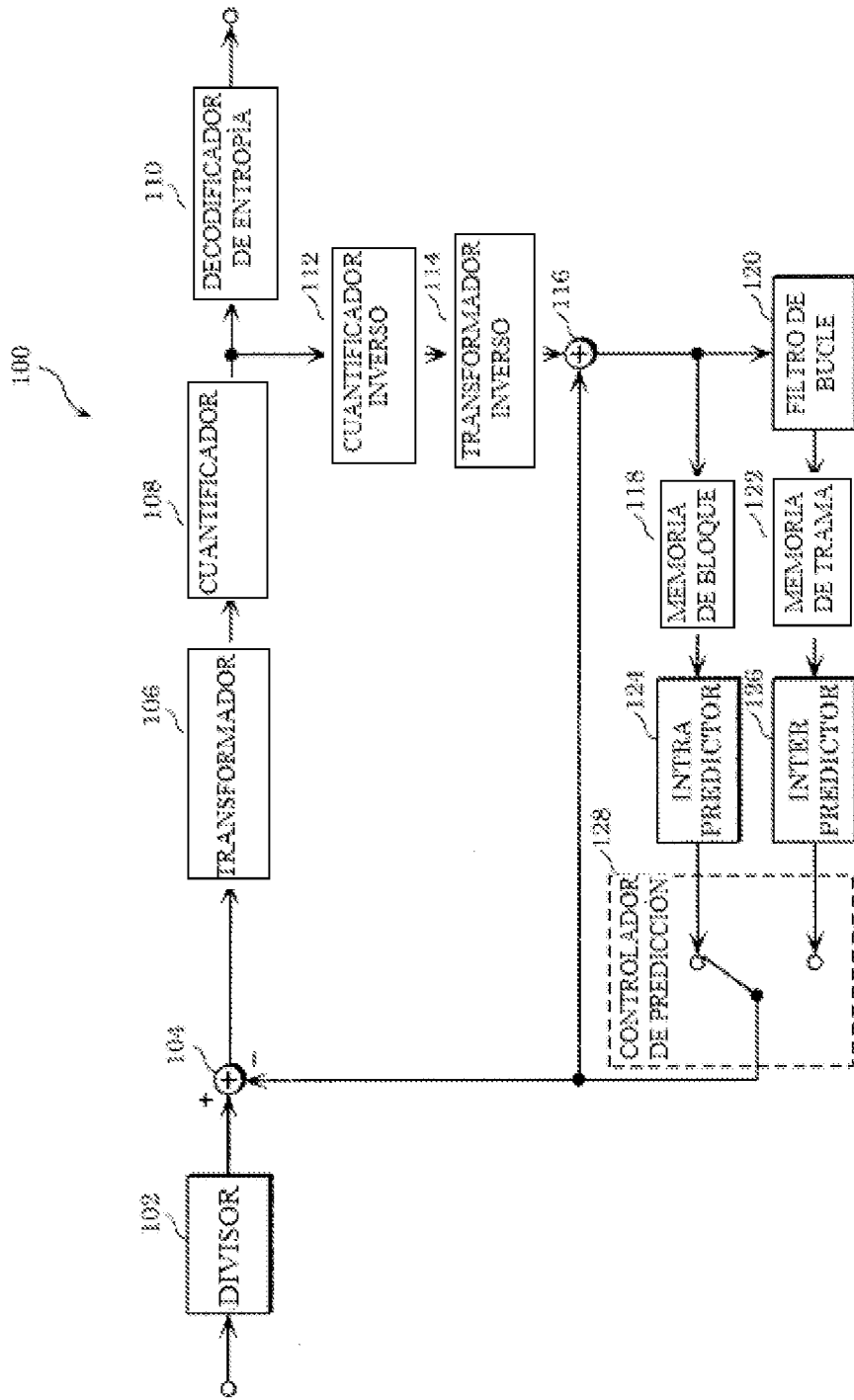


FIG. 2

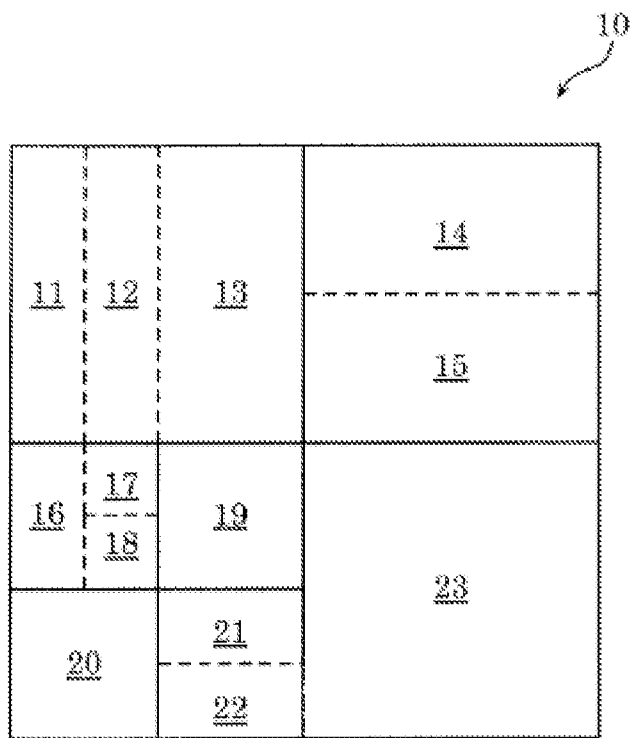


FIG. 3

TIPO DE TRANSFORMADA	FUNCIÓN BASE $T_i(j)$ , $i, j = 0, 1, \dots, N-1$
DCT · II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j+1)}{2N}\right)$ <p>DONDE <math>\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} &amp; i=0 \\ 1 &amp; i \neq 0 \end{cases}</math></p>
DCT · V	$T_i(j) = \omega_0 \cdot \omega_1 \cdot \sqrt{\frac{2}{2N-1}} \cdot \cos\left(\frac{2\pi \cdot i \cdot j}{2N-1}\right)$ <p>DONDE <math>\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} &amp; i=0 \\ 1 &amp; i \neq 0 \end{cases}</math>, <math>\omega_1 = \begin{cases} \sqrt{\frac{2}{N}} &amp; j=0 \\ 1 &amp; j \neq 0 \end{cases}</math></p>
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	$T_i(j) = \sqrt{\frac{2}{N+1}} \cdot \sin\left(\frac{\pi \cdot (i+1) \cdot (j+1)}{N+1}\right)$
DST · VII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \sin\left(\frac{\pi \cdot (2i+1) \cdot (j+1)}{2N+1}\right)$

FIG. 4A

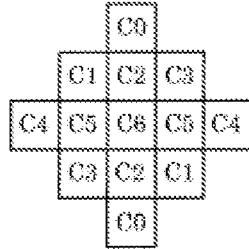


FIG. 4B

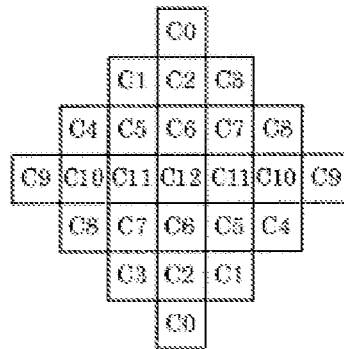


FIG. 4C

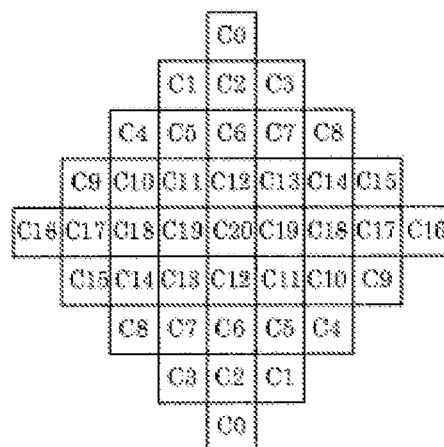


FIG. 5A

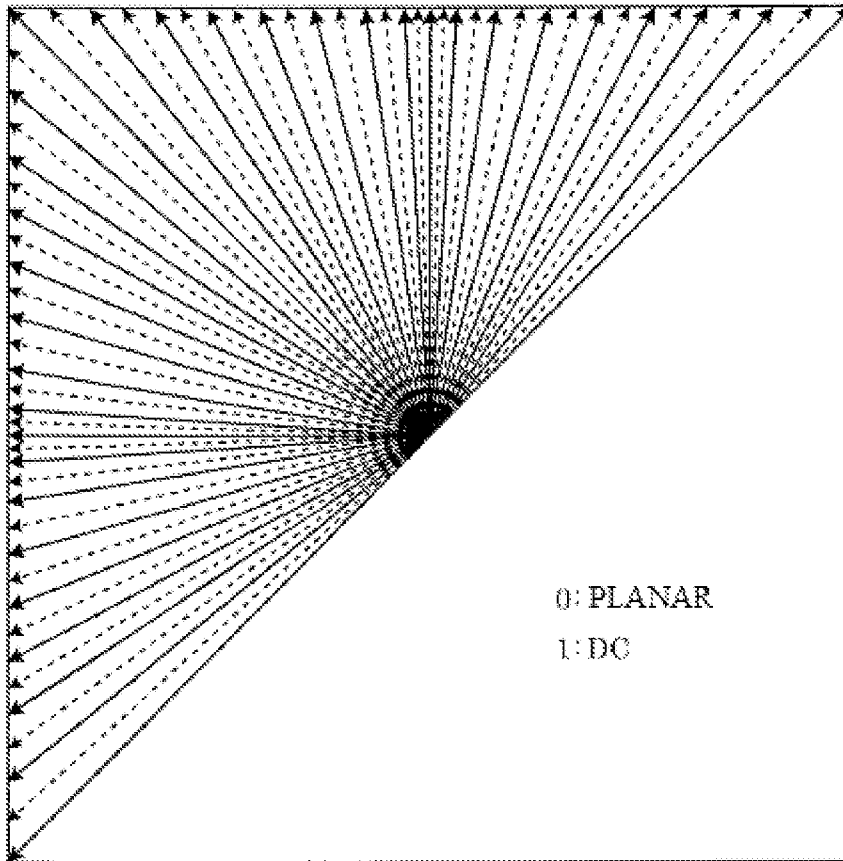


FIG. 5B

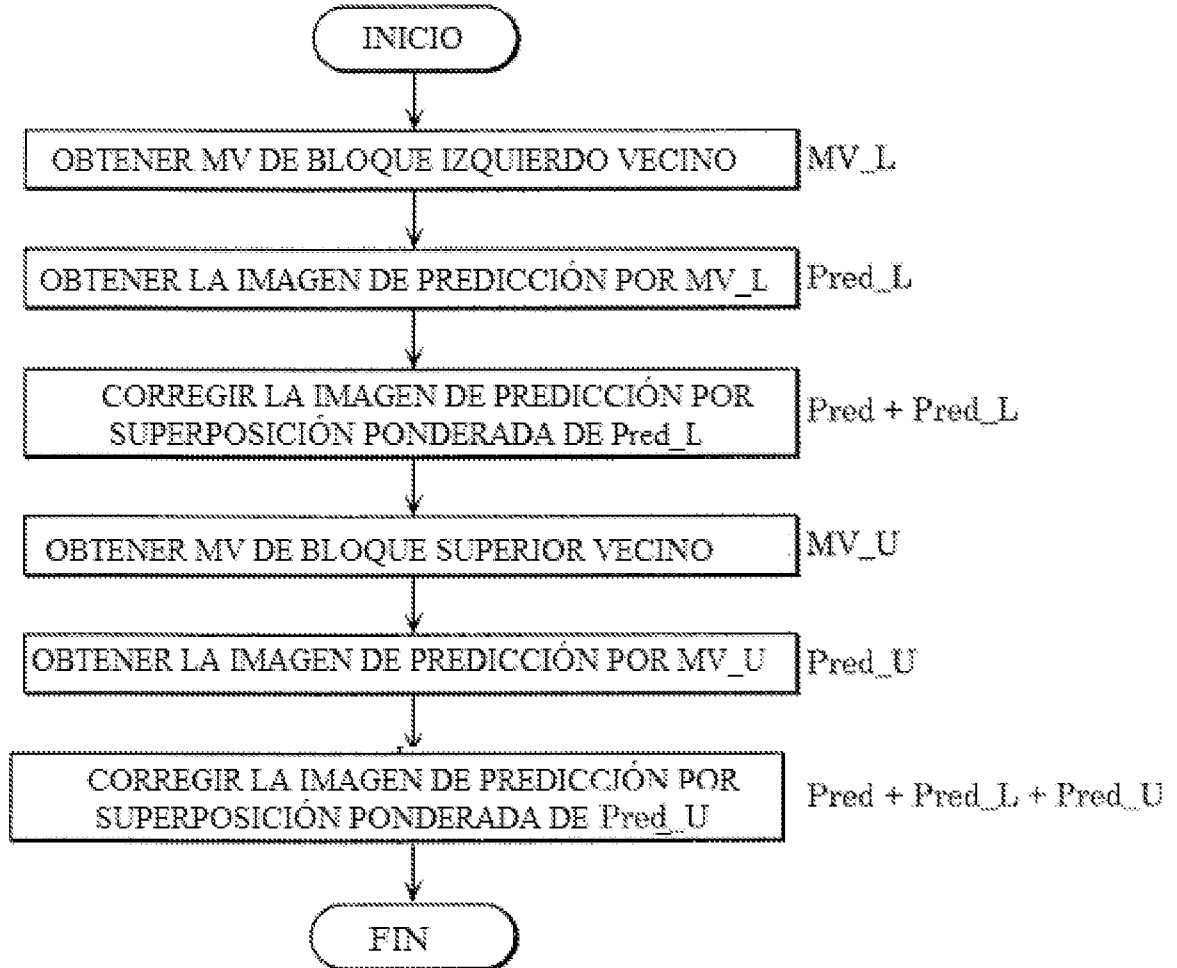


FIG. 5C

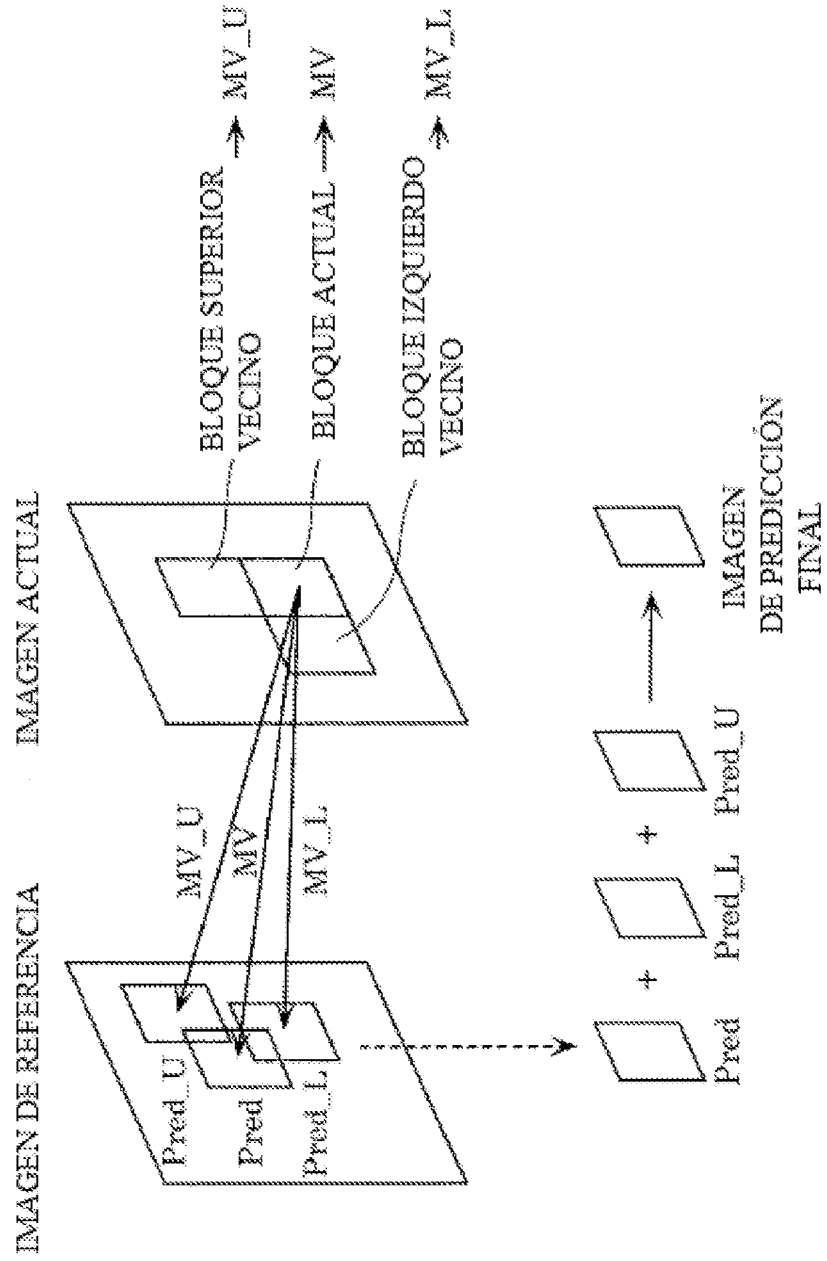


FIG. 5D

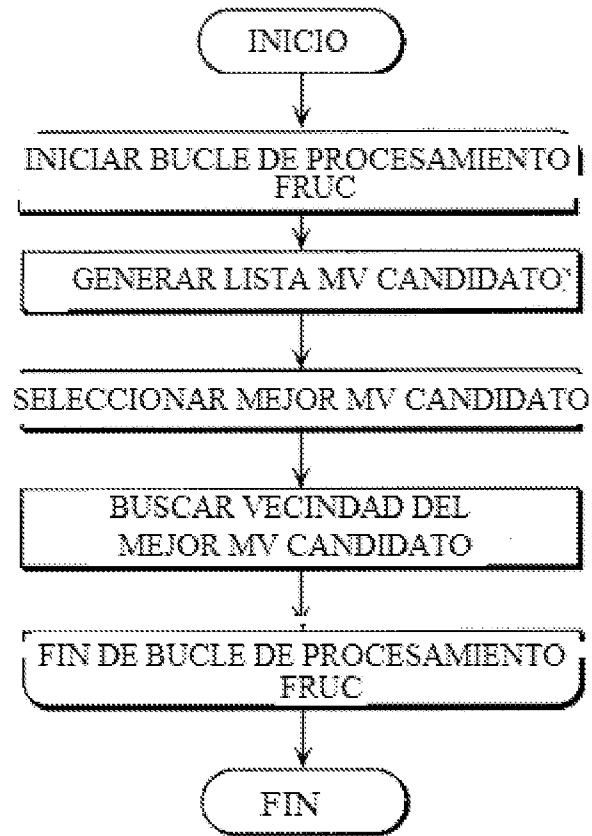


FIG. 6

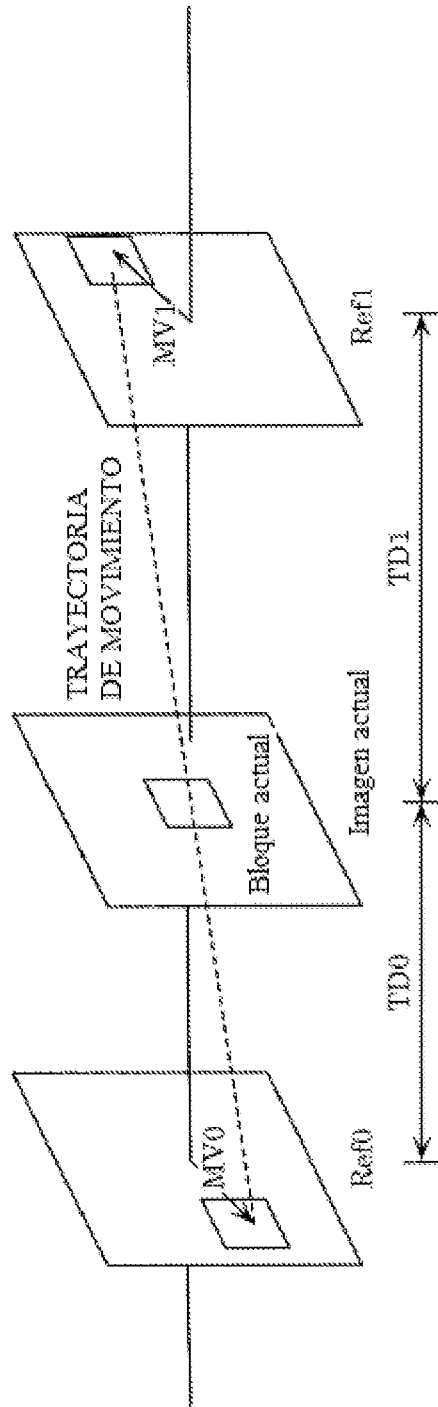


FIG. 7

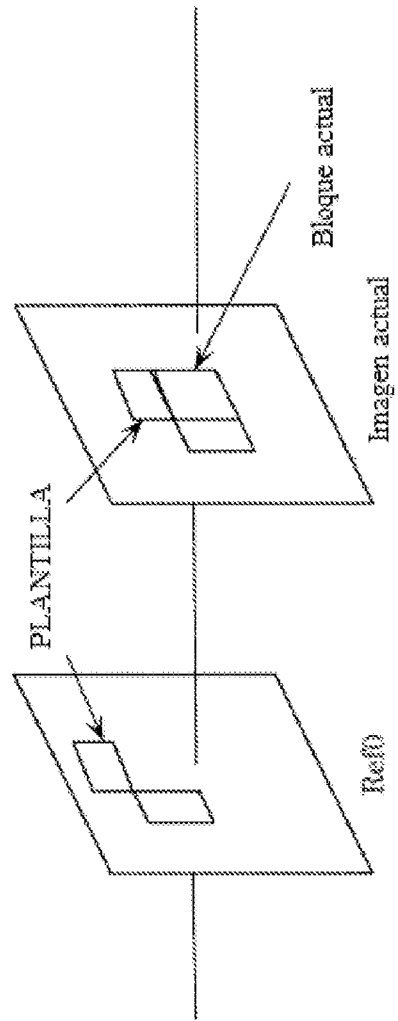


FIG. 8

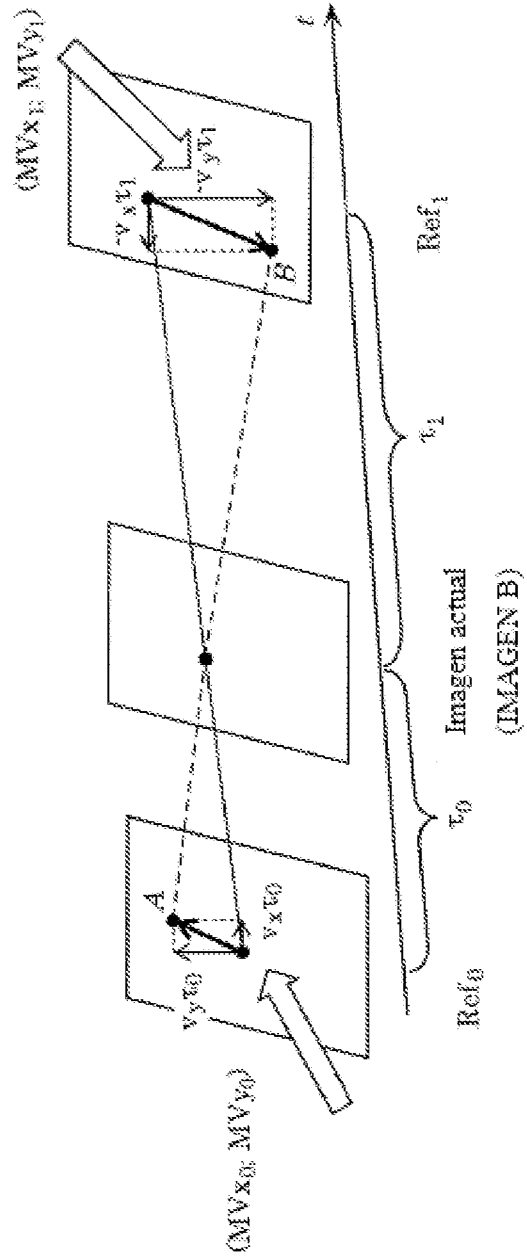


FIG. 9A

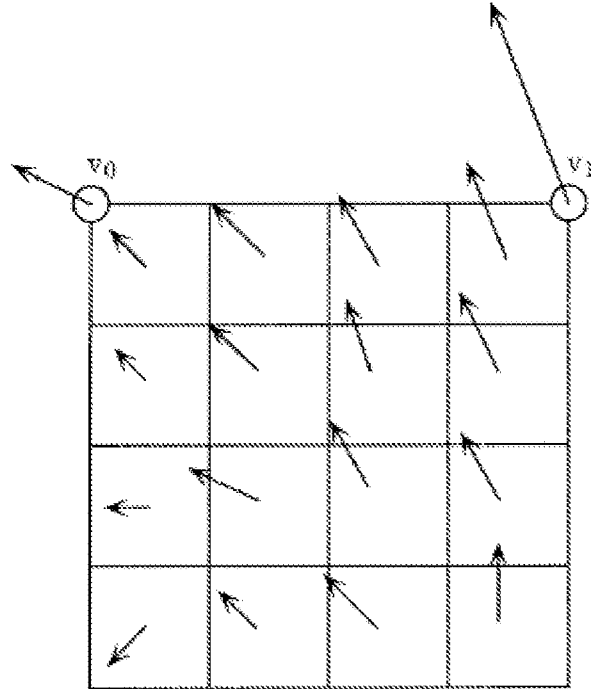
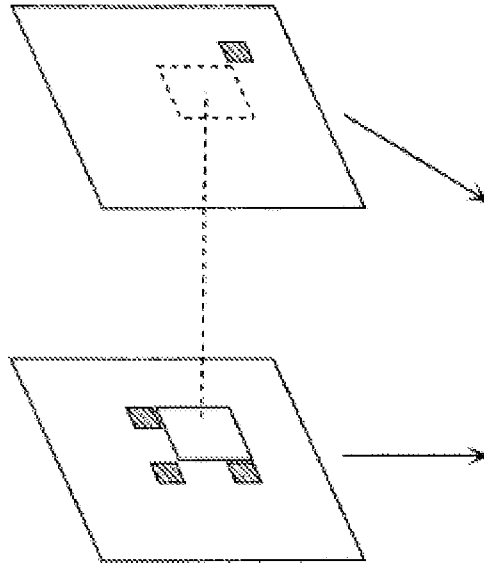


FIG. 9B  
 IMAGEN ACTUAL  
 IMAGEN DE REFERENCIA  
 CODIFICADA

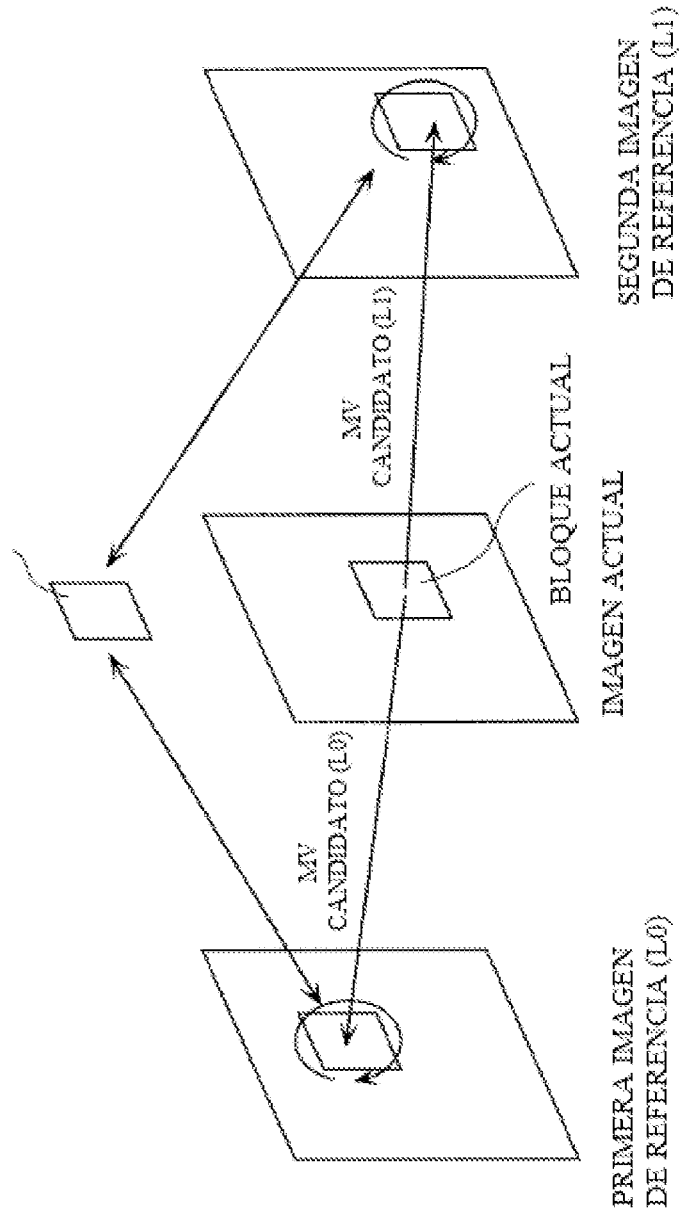


LISTA DE PREDICTOR MV CANDIDATO DE FUSION

PREDICTOR 1	PREDICTOR 2	PREDICTOR 3	PREDICTOR 4	PREDICTOR 5	PREDICTOR 6	PREDICTOR 7
MV VECINO ESPACIAL	MV VECINO ESPACIAL	MV VECINO ESPACIAL	MV VECINO ESPACIAL	MV COMBINADO	MV COMBINADO	MV CERO

SELECCIONAR UN PREDICTOR MV DE LA LISTA DEL PREDICTOR MV Y ASIGNAR COMO MV PARA EL BLOQUE ACTUAL

FIG. 9C



PLANTILLA GENERADA SOBRE LA BASE DEL PÍXEL DE REFERENCIA DEL  
MV CANDIDATO (L0) Y PÍXEL DE REFERENCIA DEL MV CANDIDATO (L1)

FIG. 9D

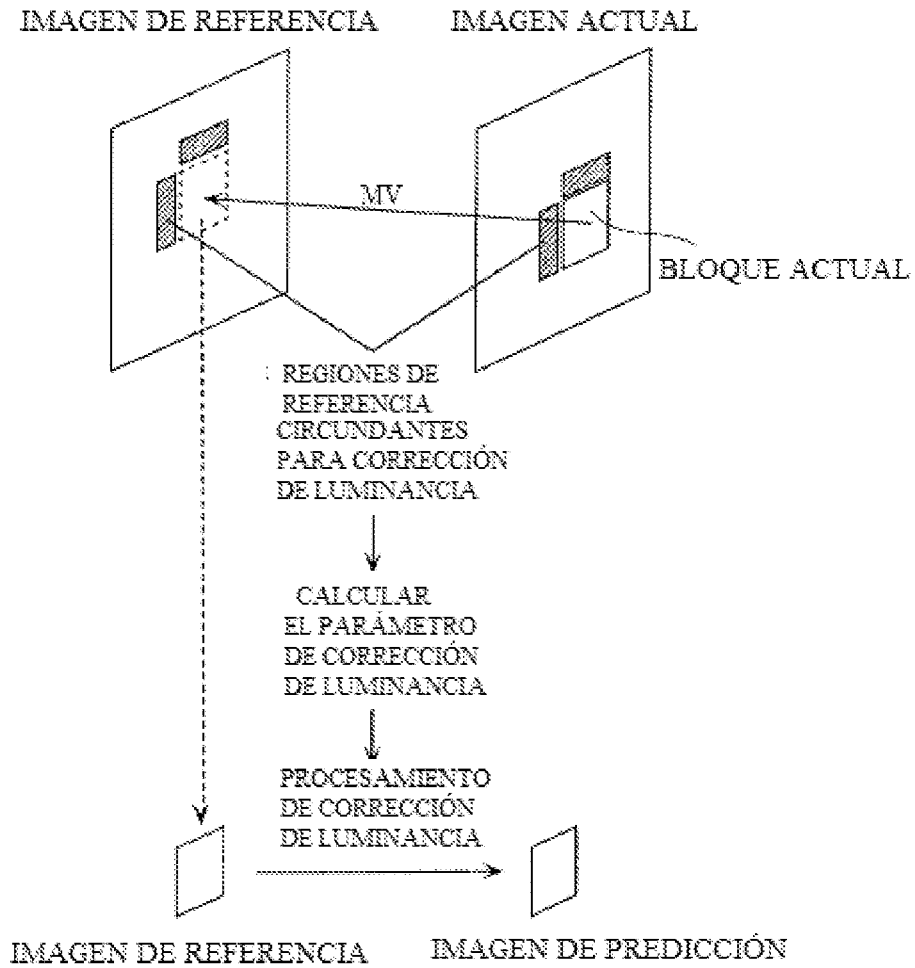


FIG. 10

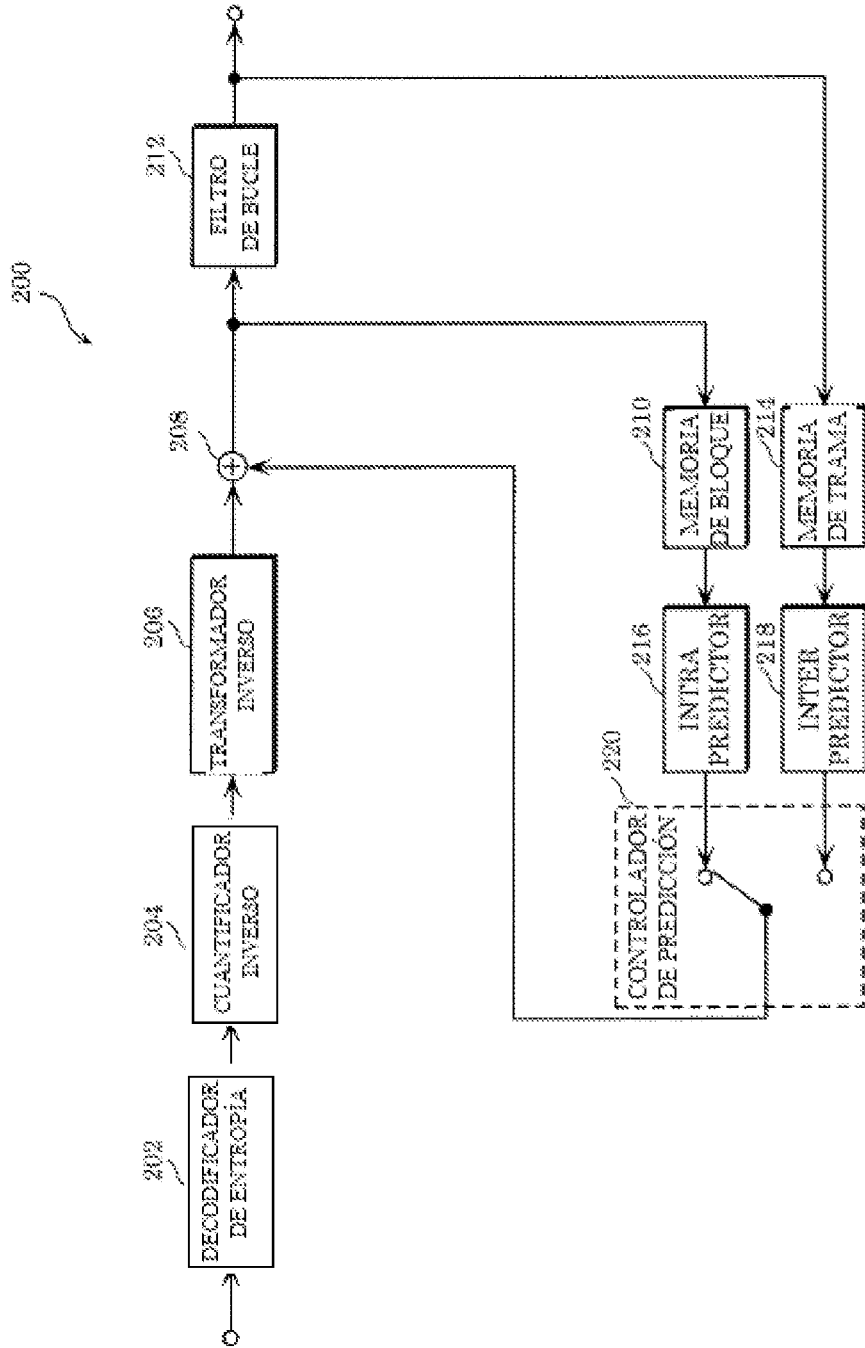


FIG. 11

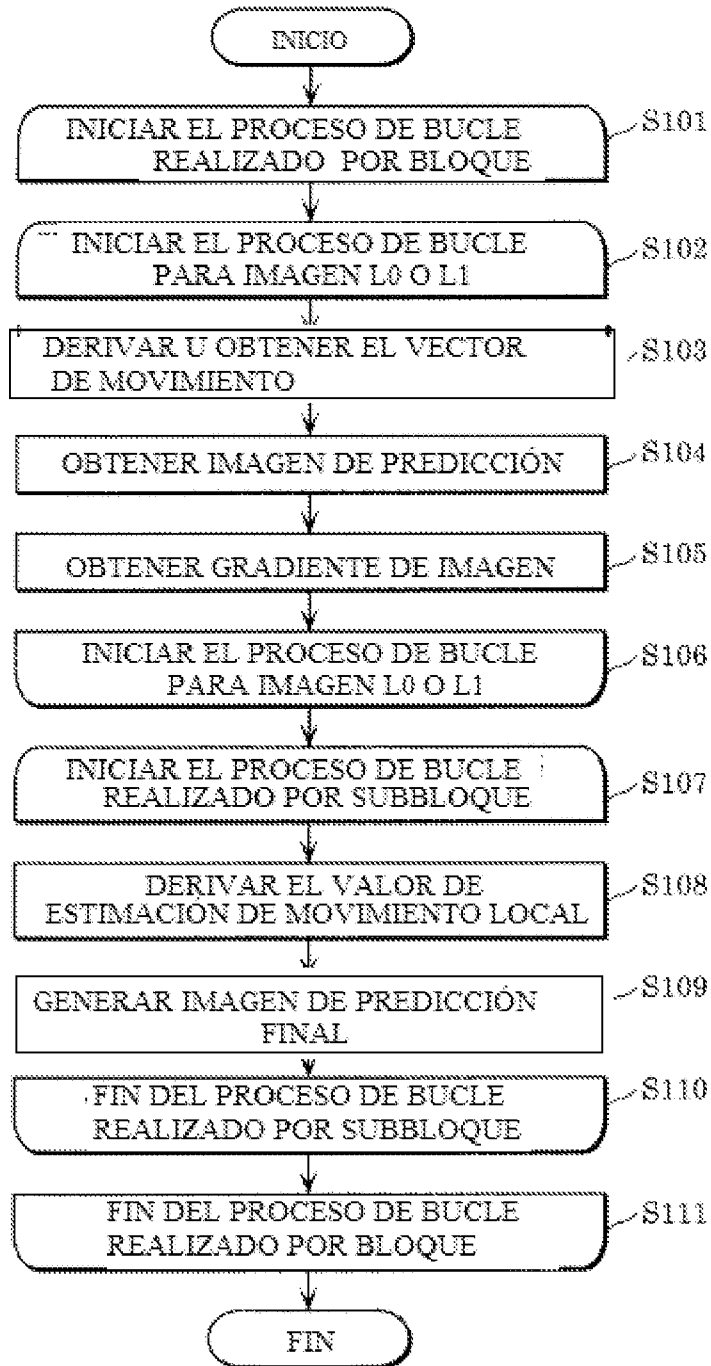


FIG. 12

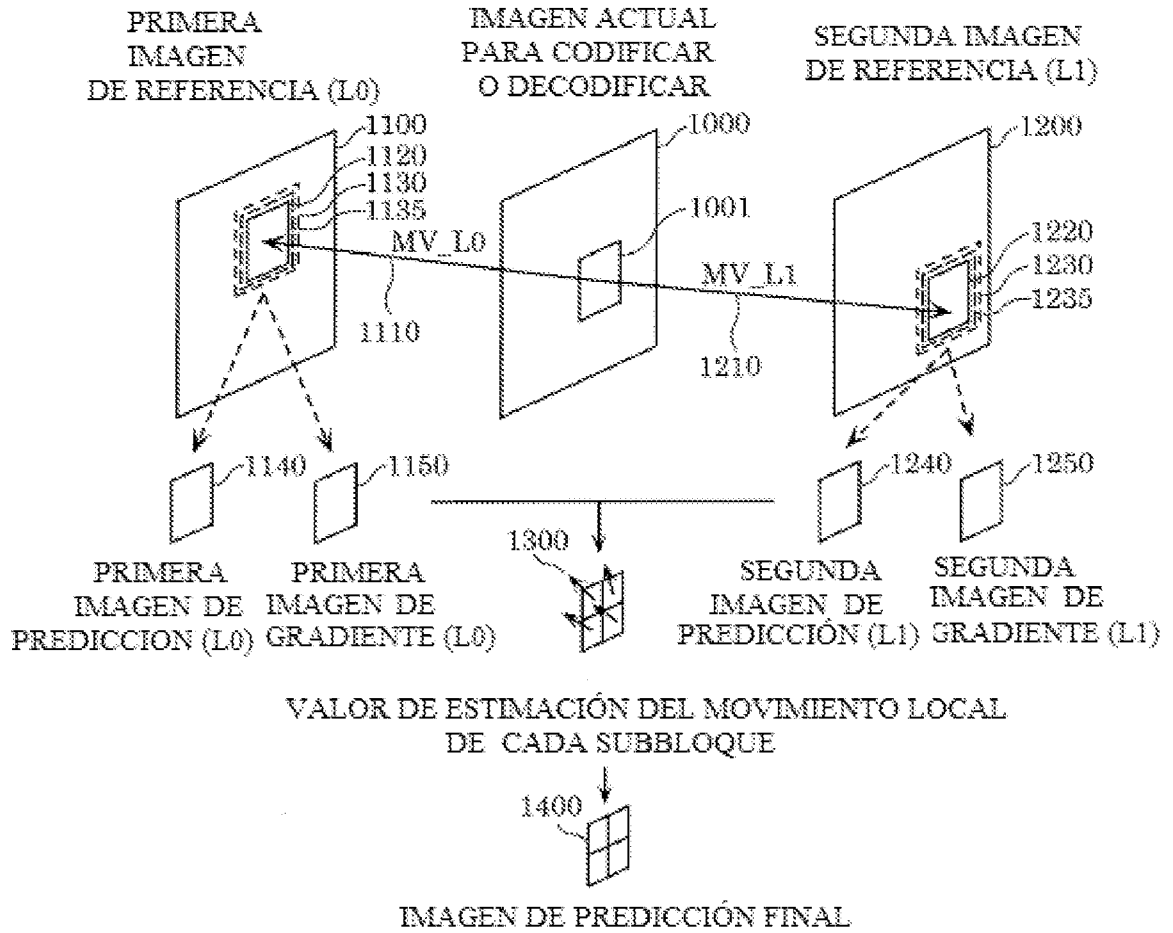


FIG. 13

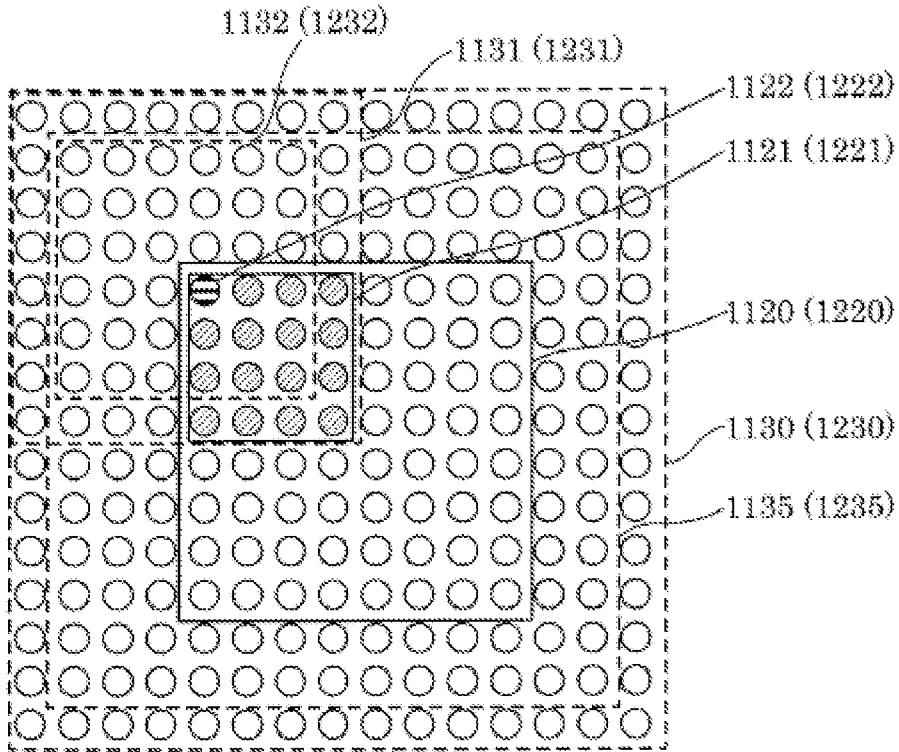


FIG. 14

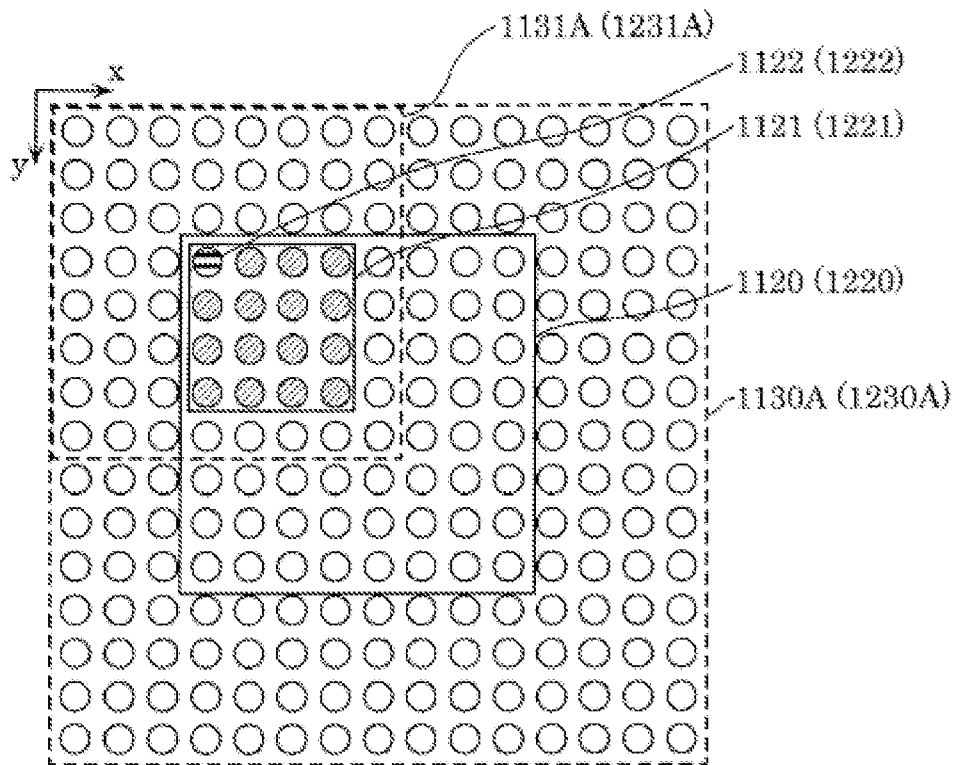


FIG. 15

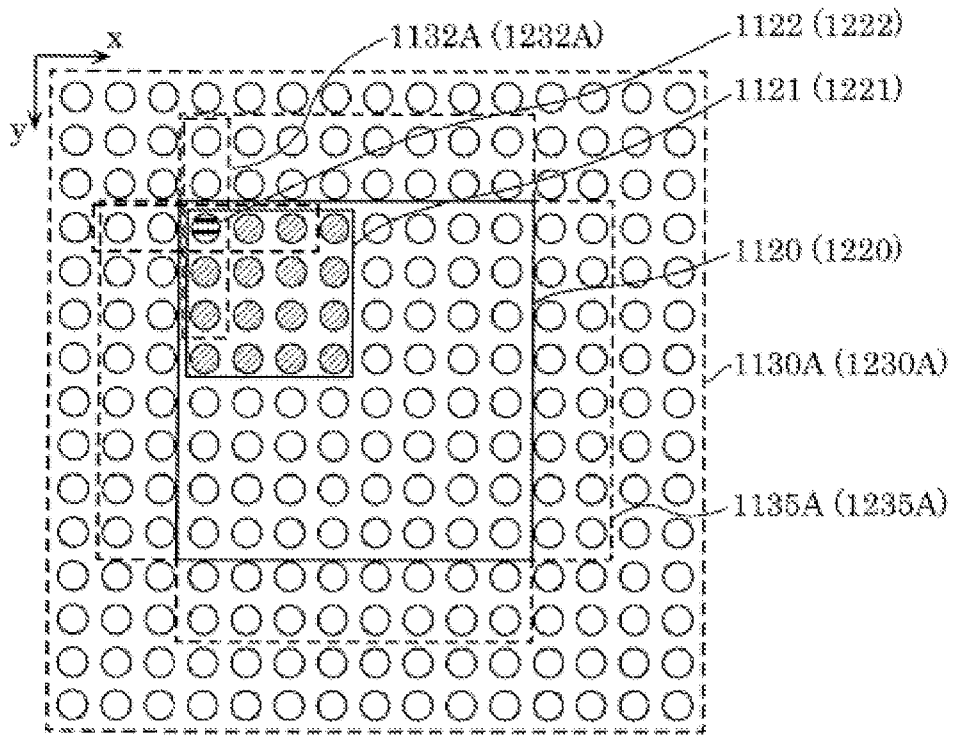
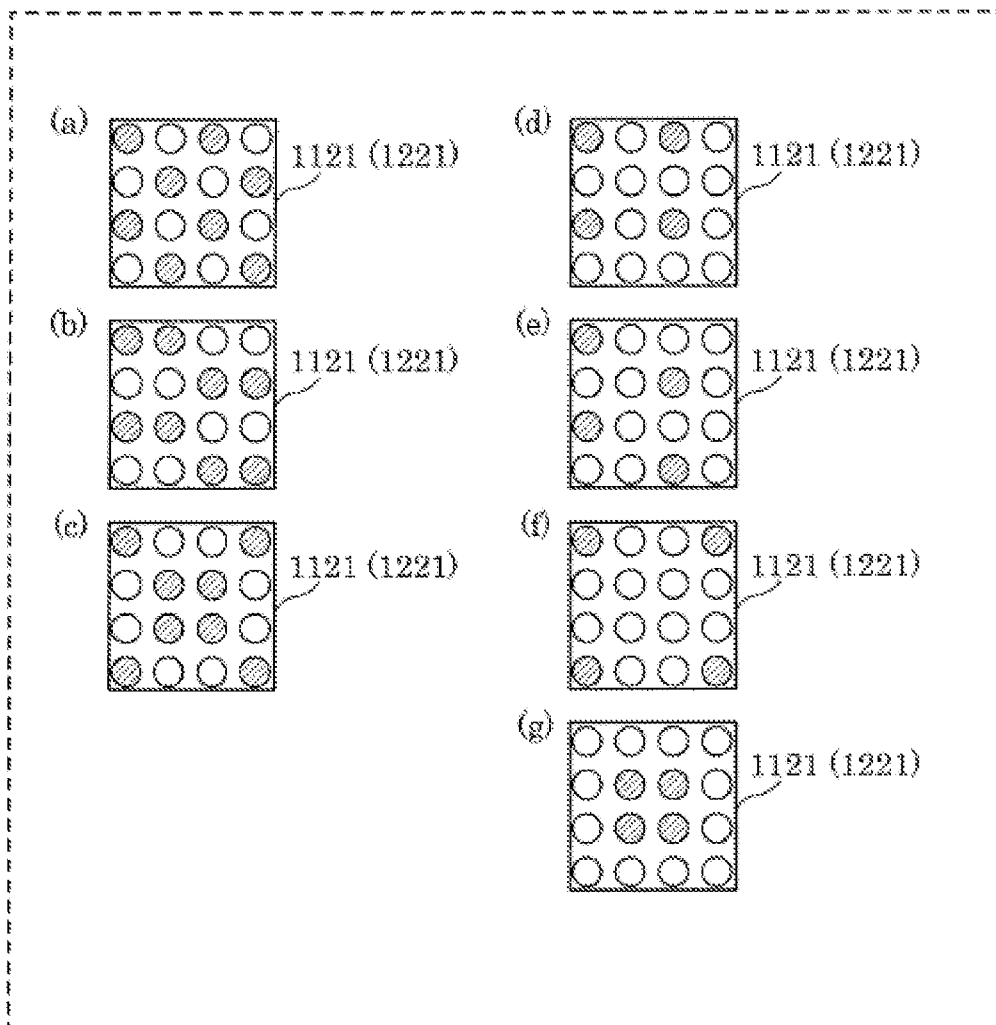


FIG. 16



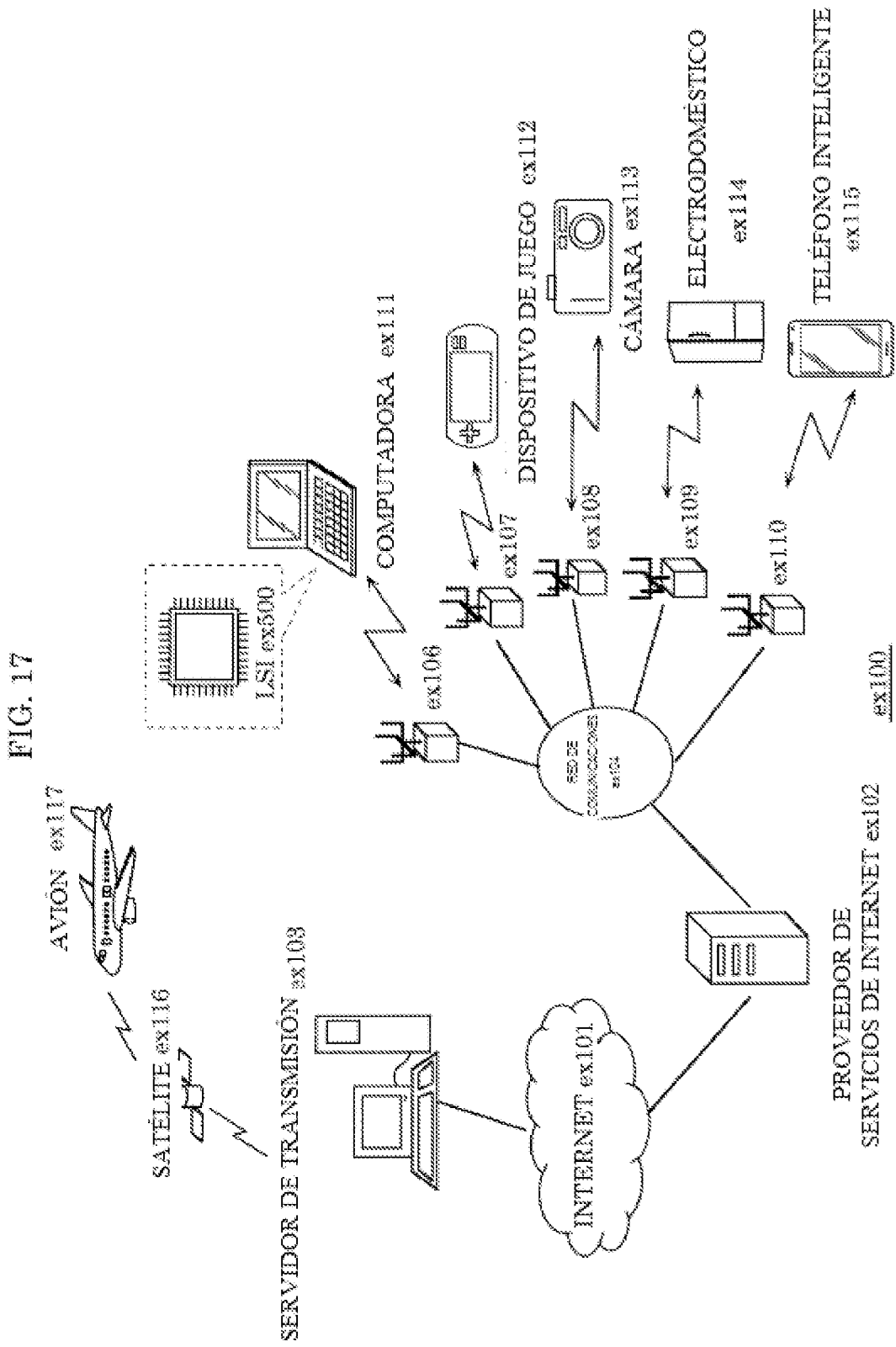


FIG. 18

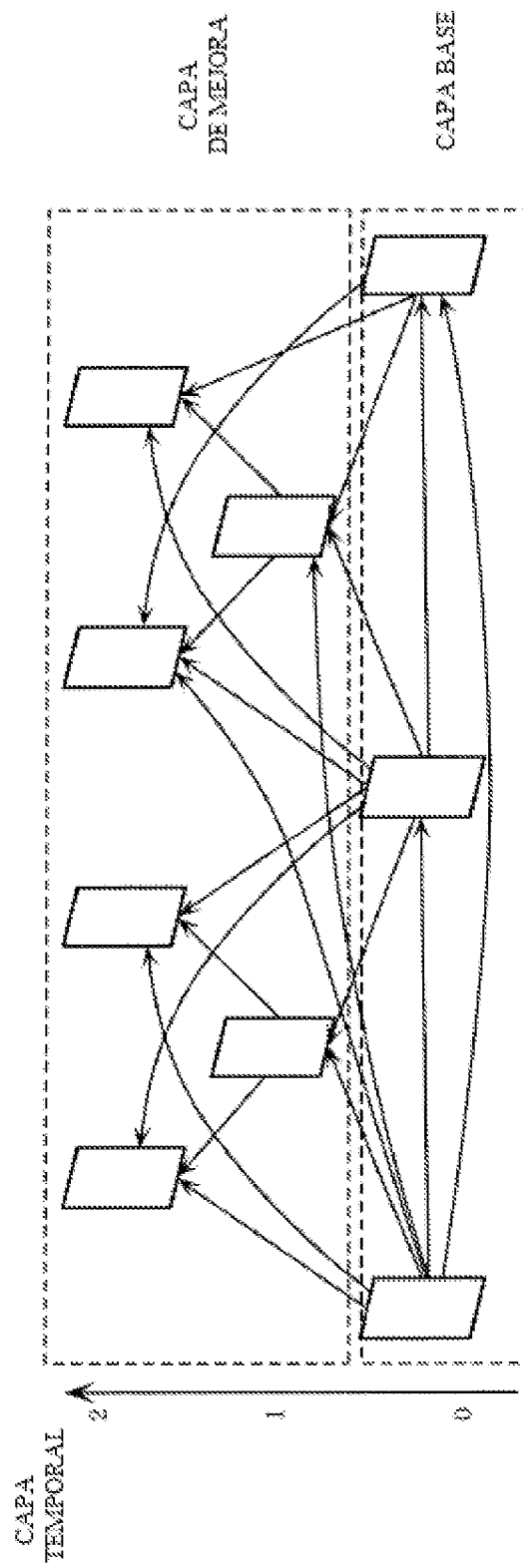


FIG. 19

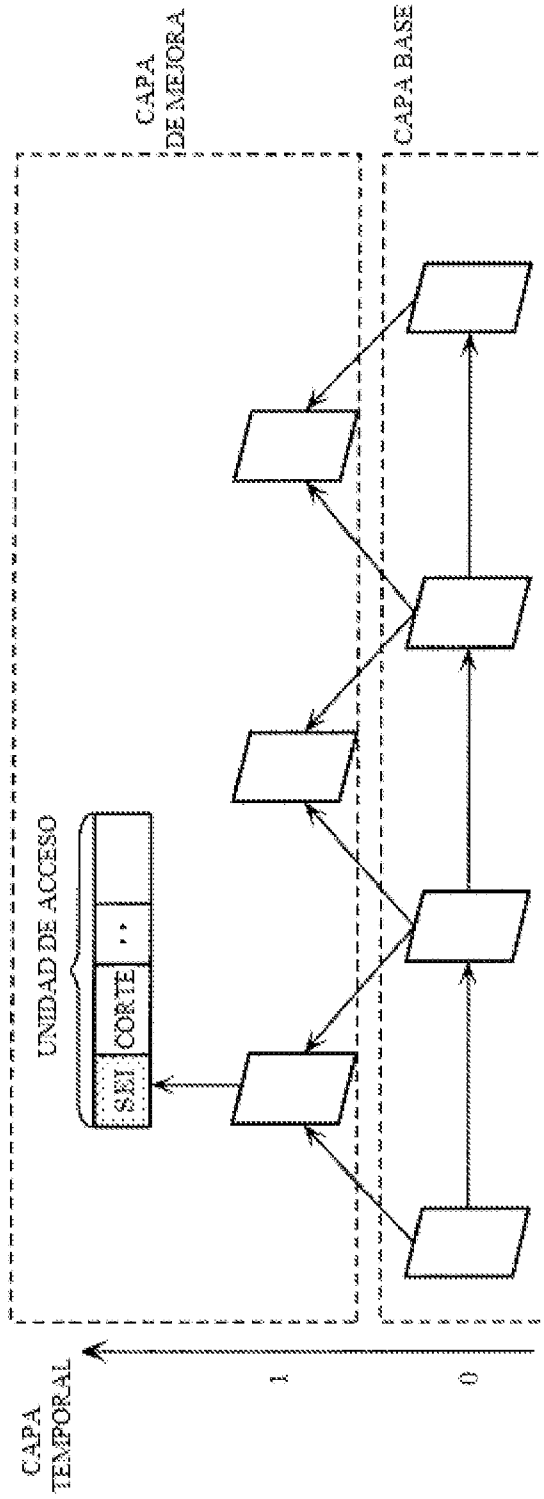


FIG. 20

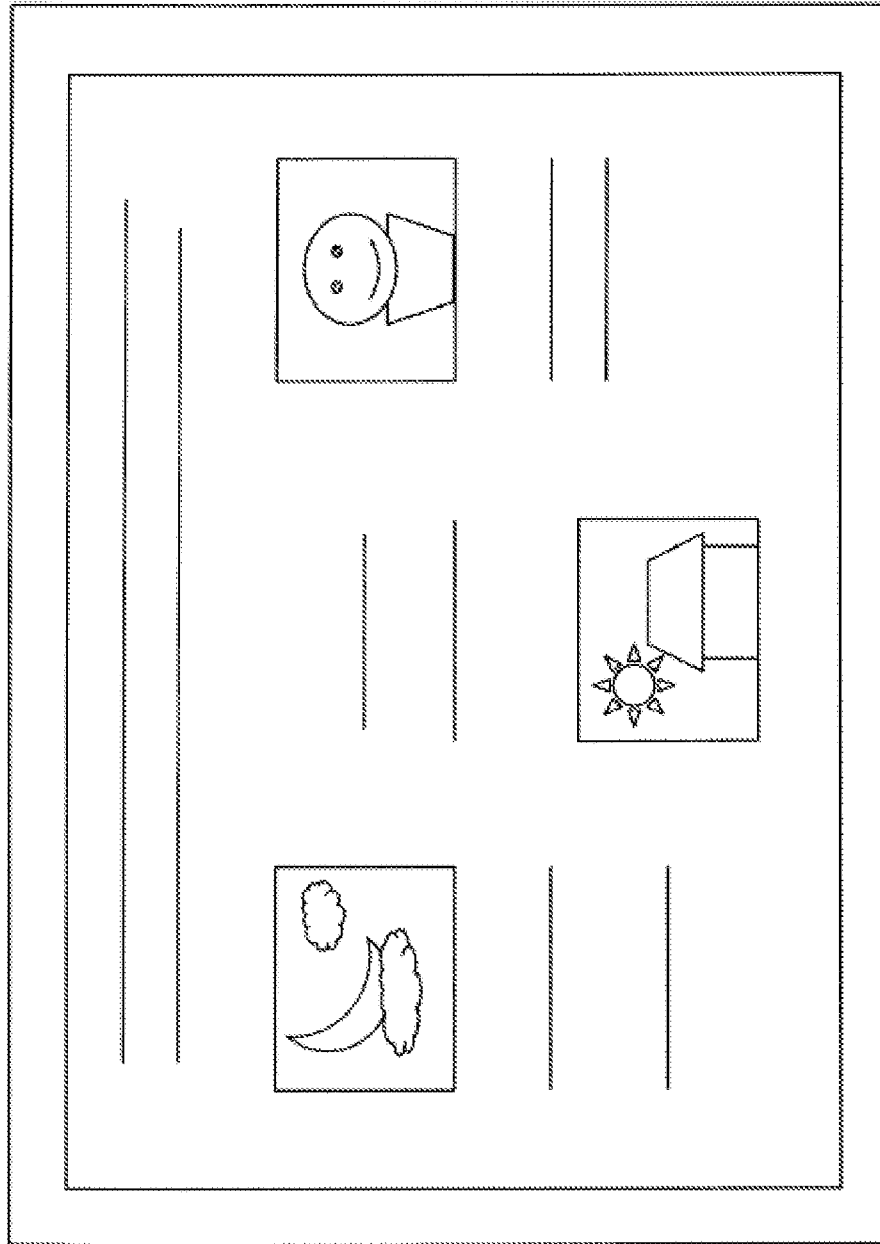


FIG. 21

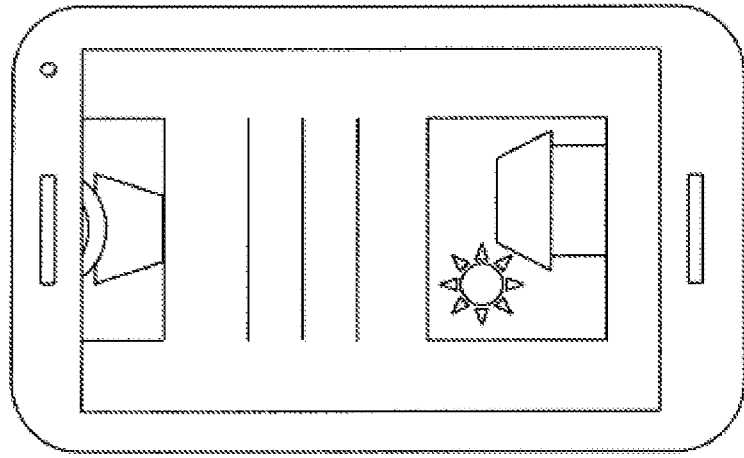


FIG. 22

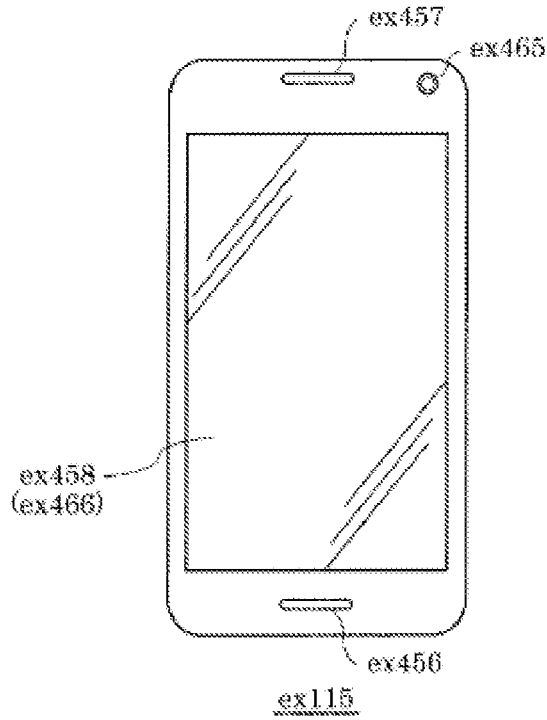


FIG. 23

