

(19)



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS
ESPAÑA



(11) Número de publicación: **2 984 938**

(51) Int. Cl.:

H04N 19/12 (2014.01)
H04N 19/157 (2014.01)
H04N 19/176 (2014.01)
H04N 19/60 (2014.01)
H04N 19/625 (2014.01)

(12)

TRADUCCIÓN DE PATENTE EUROPEA

T3

(96) Fecha de presentación y número de la solicitud europea: **11.07.2018 E 21199072 (6)**

(97) Fecha y número de publicación de la concesión europea: **08.05.2024 EP 3955572**

(54) Título: **Dispositivo de codificación, procedimiento de codificación, dispositivo de decodificación y procedimiento de decodificación**

(30) Prioridad:

13.07.2017 US 201762532050 P

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

31.10.2024

(73) Titular/es:

**PANASONIC INTELLECTUAL PROPERTY CORPORATION OF AMERICA (100.0%)
2050 W 190th Street, Suite 450
Torrance, CA 90504, US**

(72) Inventor/es:

**OHKAWA, MASATO;
SAITOU, HIDEO;
TOMA, TADAMASA;
NISHI, TAKAHIRO;
ABE, KIYOFUMI y
KANOH, RYUICHI**

(74) Agente/Representante:

GONZÁLEZ PESES, Gustavo Adolfo

ES 2 984 938 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, procedimiento de codificación, dispositivo de decodificación y procedimiento de decodificación

Campo técnico

- 5 La presente divulgación se refiere a la codificación y decodificación de una imagen/vídeo sobre una base de bloque a bloque.

Antecedentes de la técnica

Una norma de codificación de vídeo conocida como Codificación de Vídeo de Alta Eficiencia (HEVC) está normalizada por el Equipo de Colaboración Mixto sobre Codificación de Vídeo (JCT-VC).

- 10 El artículo por Chen J. y col., "Algorithm description of Joint Exploration Test Model 6 (JEM6)", Reunión del JVET de 31.05.2017, XP030150793, es una contribución a una reunión de la normalización de JVT y describe nuevas características y procedimientos de codificación.

Lista de citas

Referencia no de patente

- 15 NPL 1: H.265 (ISO/IEC 23008-2 HEVC (Codificación de Vídeo de Alta Eficiencia))

Sumario de la invención

Problema técnico

Existe una demanda para que tal tecnología de codificación y decodificación reduzca la carga de procesamiento mientras inhibe un descenso en la eficiencia de compresión.

- 20 En vista de esto, la presente divulgación proporciona un codificador, un decodificador, un procedimiento de codificación o un procedimiento de decodificación con capacidad de reducir una carga de procesamiento mientras inhibe un descenso en la eficiencia de compresión.

Solución al problema

- 25 La presente invención se define por las características de las reivindicaciones independientes. La divulgación habilitante para la invención se encuentra en la realización 2.

Las referencias a realizaciones que no entran dentro del alcance de las reivindicaciones deben entenderse como ejemplos útiles para la comprensión de la invención.

- 30 Obsérvese que estos aspectos generales o específicos pueden implementarse por un sistema, un procedimiento, un circuito integrado, un programa informático o un medio de grabación legible por ordenador, tal como un CD-ROM, o por cualquier combinación de sistemas, procedimientos, circuitos integrados, programas informáticos o medios de grabación.

Efectos ventajosos de la invención

- 35 La presente divulgación puede proporcionar un codificador, un decodificador, un procedimiento de codificación o un procedimiento de decodificación con capacidad de reducir carga de procesamiento mientras inhibe un descenso en la eficiencia de compresión.

Breve descripción de los dibujos

La Figura 1 es un diagrama de bloques que ilustra una configuración funcional de un codificador de acuerdo con la realización 1.

La Figura 2 ilustra un ejemplo de división de bloque de acuerdo con la realización 1.

- 40 La Figura 3 es un gráfico que indica funciones de base de transformada para cada tipo de transformada.

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

La Figura 4B ilustra otro ejemplo de una forma de filtro usada en ALF.

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

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

La Figura 5B es un diagrama de flujo para ilustrar un esquema de un procedimiento de corrección de imagen de predicción realizado a través de procesamiento de OBMC.

La Figura 5C es un diagrama conceptual para ilustrar un esquema de un procedimiento de corrección de imagen de predicción realizado a través de procesamiento de OBMC.

5 La Figura 5D ilustra un ejemplo de FRUC.

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

La Figura 7 es para ilustrar concordancia de patrones (concordancia de plantilla) entre una plantilla en la instantánea actual y un bloque en una instantánea de referencia.

10 La Figura 8 es para ilustrar un modelo que supone movimiento lineal uniforme.

La Figura 9A es para ilustrar la derivación de un vector de movimiento de cada subbloque a base de vectores de movimiento de bloques vecinos.

La Figura 9B es para ilustrar un esquema de un procedimiento de derivación de un vector de movimiento a través de modo de fusión.

15 La Figura 9C es un diagrama conceptual para ilustrar un esquema de procesamiento de DMVR.

La Figura 9D es para ilustrar un esquema de un procedimiento de generación de imágenes de predicción usando un procedimiento de corrección de luminancia realizado a través de procesamiento de LIC.

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

20 La Figura 11 es un diagrama de flujo que ilustra un procesamiento de transformación y cuantificación realizado por un codificador de acuerdo con la realización 2.

La Figura 12 es un diagrama de flujo que ilustra un procesamiento de cuantificación inversa y transformación inversa realizado por un decodificador de acuerdo con la realización 2.

25 La Figura 13 es un diagrama de flujo que ilustra un procesamiento de transformación y cuantificación realizado por un codificador de acuerdo con la realización 3.

La Figura 14 es un diagrama de flujo que ilustra un procesamiento de cuantificación inversa y transformación inversa realizado por un decodificador de acuerdo con la realización 3.

La Figura 15 es un diagrama de flujo que ilustra un procesamiento de transformación y cuantificación realizado por un codificador de acuerdo con la realización 4.

30 La Figura 16 es un diagrama de flujo que ilustra un procesamiento de cuantificación inversa y transformación inversa realizado por un decodificador de acuerdo con la realización 4.

La Figura 17 es un diagrama de flujo que ilustra un procesamiento de codificación realizado por un codificador de acuerdo con la realización 5.

35 La Figura 18 ilustra un ejemplo específico de sintaxis de acuerdo con la realización 5.

La Figura 19 es un diagrama de flujo que ilustra un procesamiento de decodificación realizado por un decodificador de acuerdo con la realización 5.

La Figura 20 es un diagrama de flujo que ilustra un procesamiento de codificación realizado por un codificador de acuerdo con la realización 6.

40 La Figura 21 es un diagrama de flujo que ilustra un procesamiento de decodificación realizado por un decodificador de acuerdo con la realización 6.

La Figura 22 ilustra una configuración global de un sistema de suministro de contenido de implementación de un servicio de distribución de contenido.

La Figura 23 ilustra un ejemplo de una estructura de codificación en codificación escalable.

La Figura 24 ilustra un ejemplo de una estructura de codificación en codificación escalable.

45 La Figura 25 ilustra un ejemplo de una pantalla de visualización de una página web.

La Figura 26 ilustra un ejemplo de una pantalla de visualización de una página web.

La Figura 27 ilustra un ejemplo de un teléfono inteligente.

La Figura 28 es un diagrama de bloques que ilustra un ejemplo de configuración de un teléfono inteligente.

Descripción de realizaciones ilustrativas

- 5 (Conocimiento subyacente que forma la base de la presente divulgación)

Se propone una transformada de frecuencia de dos etapas para bloques a los que se aplica intra predicción en software de Modelo de Prueba de Exploración Mixto (JEM) del Equipo de Exploración Mixto de Vídeo (JVET). La transformada de frecuencia de dos etapas usa transformada de múltiples núcleos explícita (EMT) como la transformada primaria y transformada secundaria no separable (NSST) como la transformada secundaria. La EMT selecciona de forma adaptativa una base de transformada de entre una pluralidad de bases de transformada para realizar una transformada desde el dominio espacial hasta el dominio de frecuencia.

10 Una transformada de frecuencia de dos etapas de este tipo tiene margen de mejora en términos de la cantidad de procesamiento.

15 A continuación en el presente documento, se describirán específicamente con referencia a los dibujos realizaciones a base de tal conocimiento como se describe anteriormente.

20 Obsérvese que las siguientes realizaciones describen ejemplos generales o específicos. Los valores numéricos, formas, materiales, elementos constituyentes, la disposición y conexión de los elementos constituyentes, etapas, el orden de procesamiento de las etapas, etc., ilustrados en las siguientes realizaciones son meros ejemplos y no pretenden limitar el ámbito de las reivindicaciones. Además, entre los elementos constituyentes en las siguientes realizaciones, aquellos no citados en ninguna de las reivindicaciones independientes que representan los conceptos más genéricos se describirán como elementos constituyentes opcionales.

REALIZACIÓN 1

25 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 procedimientos y/o configuraciones presentados en una descripción posterior de aspectos de la presente divulgación. Obsérvese que la realización 1 es solamente un ejemplo de un codificador y un decodificador a los que son aplicables los procedimientos y/o configuraciones presentados en la descripción de aspectos de la presente divulgación. Los procedimientos y/o configuraciones presentados en la descripción de aspectos de la presente divulgación también pueden implementarse en un codificador y un decodificador diferentes de aquellos de acuerdo con la realización 1.

30 Cuando los procedimientos y/o configuraciones presentados en la descripción de aspectos de la presente divulgación se aplican a la realización 1, por ejemplo, puede realizarse cualquiera de lo siguiente.

35 (1) con respecto al codificador o el decodificador de acuerdo con la realización 1, entre componentes incluidos en el codificador o el decodificador de acuerdo con la realización 1, sustituir un componente que corresponde a un componente presentado en la descripción de aspectos de la presente divulgación con un componente presentado en la descripción de aspectos de la presente divulgación;

40 (2) con respecto al codificador o el decodificador de acuerdo con la realización 1, implementar cambios discretionales a funciones o procedimientos implementados realizados por uno o más componentes incluidos en el codificador o el decodificador de acuerdo con la realización 1, tal como adición, sustitución o eliminación, etc., de tales funciones o procedimientos implementados, sustituyendo, a continuación, un componente que corresponde a un componente presentado en la descripción de aspectos de la presente divulgación con un componente presentado en la descripción de aspectos de la presente divulgación;

45 (3) con respecto al procedimiento implementado por el codificador o el decodificador de acuerdo con la realización 1, implementar cambios discretionales, tales como adición de procedimientos y/o sustitución, eliminación de uno o más de los procedimientos incluidos en el procedimiento, y sustituir, a continuación, un procedimiento que corresponde a un procedimiento presentado en la descripción de aspectos de la presente divulgación con un procedimiento presentado en la descripción de aspectos de la presente divulgación;

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

- 5 (5) combinar un componente que incluye una o más funciones incluidas en uno o más componentes incluidos en el codificador o el decodificador de acuerdo con la realización 1, o un componente que implementa uno o más procedimientos implementados por uno o más componentes incluidos en el codificador o el decodificador de acuerdo con la realización 1 con un componente presentado en la descripción de aspectos de la presente divulgación, un componente que incluye una o más funciones incluidas en un componente presentado en la descripción de aspectos de la presente divulgación, o un componente que implementa uno o más procedimientos implementados por un componente presentado en la descripción de aspectos de la presente divulgación;
- 10 (6) con respecto al procedimiento implementado por el codificador o el decodificador de acuerdo con la realización 1, entre procedimientos incluidos en el procedimiento, sustituir un procedimiento que corresponde a un procedimiento presentado en la descripción de aspectos de la presente divulgación con un procedimiento presentado en la descripción de aspectos de la presente divulgación; y (7) combinar uno o más procedimientos incluidos en el procedimiento implementado por el codificador o el decodificador de acuerdo con la realización 1 con un procedimiento presentado en la descripción de aspectos de la presente divulgación.

20 Obsérvese que la implementación de los procedimientos y/o configuraciones presentados en la descripción de aspectos de la presente divulgación no se limita a los ejemplos anteriores. Por ejemplo, los procedimientos y/o configuraciones presentados en la descripción de aspectos de la presente divulgación pueden implementarse en un dispositivo usado para un fin diferente del codificador de instantáneas/instantáneas en movimiento o el decodificador de instantáneas/instantáneas en movimiento desvelado en la realización 1. Además, los procedimientos y/o configuraciones presentados en la descripción de aspectos de la presente divulgación pueden implementarse independientemente. Además, los procedimientos y/o las configuraciones descritos en diferentes aspectos pueden combinarse.

- 25 [Esquema de codificador]
- 25 En primer lugar, se describirá el esquema del codificador de acuerdo con la realización 1. La Figura 1 es un diagrama de bloques que ilustra una configuración funcional del codificador 100 de acuerdo con la realización 1. El codificador 100 es una codificador de instantánea en movimiento/instantánea que codifica una instantánea en movimiento/instantánea bloque a bloque.

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

35 El codificador 100 se realiza como, por ejemplo, un procesador y una memoria genéricos. En este caso, cuando el procesador ejecuta un programa de software almacenado en la memoria, el procesador funciona como el divisor 102, el restador 104, el transformador 106, el cuantificador 108, el codificador 110 por entropía, el cuantificador 112 inverso, el transformador 114 inverso, el sumador 116, el filtro 120 de bucle, el intra predictor 124, el inter predictor 126 y el controlador 128 de predicción. Como alternativa, el codificador 100 puede realizarse como uno o más circuitos electrónicos especializados que corresponden al divisor 102, al restador 104, al transformador 106, al cuantificador 108, al codificador 110 por entropía, al cuantificador 112 inverso, al transformador 114 inverso, al sumador 116, al filtro 120 de bucle, al intra predictor 124, a inter predictor 126 y al controlador 128 de predicción.

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

- [Divisor]
- 45 El divisor 102 divide cada instantánea incluida en una instantánea en movimiento introducida en bloques y emite cada bloque al restador 104. Por ejemplo, el divisor 102 divide primero una instantánea en bloques de un tamaño fijo (por ejemplo, 128×128). El bloque de tamaño fijo también se denomina unidad de árbol de codificación (CTU). El divisor 102 divide, a continuación, cada bloque de tamaño fijo en bloques de tamaños variables (por ejemplo, 64×64 o más pequeño), a base de división recursiva de bloque de árbol cuádruple y/o árbol binario. El bloque de tamaño variable también se denomina una unidad de codificación (CU), una unidad de predicción (PU) o una unidad de transformada (TU). Obsérvese que en esta realización, no existe la necesidad de diferenciar CU, PU y TU; todos o algunos de los bloques en una instantánea pueden procesarse por CU, PU o TU.

55 La Figura 2 ilustra un ejemplo de división de bloque de acuerdo con la realización 1. En la Figura 2, las líneas continuas representan límites de bloque de bloques divididos por división de bloque de árbol cuádruple, y las líneas discontinuas representan límites de bloque de bloques divididos por división de bloque de árbol binario.

En este punto, el bloque 10 es un bloque de 128×128 píxeles cuadrado (bloque de 128×128). Este bloque de

128×128 10 se divide primero en cuatro bloques de 64×64 cuadrados (división de bloque de árbol cuádruple).

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

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

El bloque de 64×64 inferior izquierdo se divide primero en cuatro bloques de 32×32 cuadrados (división de 10 bloque de árbol cuádruple). El bloque superior izquierdo y el bloque inferior derecho entre los cuatro bloques de 32×32 se dividen adicionalmente.

15 El bloque de 32×32 superior izquierdo se divide verticalmente en dos bloques de 16×32 rectangulares y el bloque 16×32 derecho se divide horizontalmente adicionalmente en dos bloques de 16×16 (división de bloque de árbol binario). El bloque de 32×32 inferior derecho se divide horizontalmente en dos bloques de 32×16 (división de bloque de árbol binario). Como resultado, el bloque de 64×64 inferior izquierdo se divide en el bloque 16 de 16×32, dos bloques 17 y 18 de 16×16, dos bloques 19 y 20 de 32×32 y dos bloques 21 y 22 de 32×16.

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

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

20 Obsérvese que en la Figura 2, un bloque se divide en cuatro o dos bloques (división de bloque de árbol cuádruple o árbol binario), pero la división no se limita a este ejemplo. Por ejemplo, un bloque puede dividirse en tres bloques (división de bloque ternaria). La división que incluye tal división de bloque ternaria también se denomina división de árbol de múltiples tipos (MBT).

[Restador]

25 El restador 104 resta una señal de predicción (muestra de predicción) de una señal original (muestra original) por bloque dividido por el divisor 102. En otras palabras, el restador 104 calcula errores de predicción (también denominados residuales) de un bloque que hay que codificar (a continuación en el presente documento denominado un bloque actual). El restador 104 emite, a continuación, los errores de predicción calculados al transformador 106.

30 La señal original es una señal introducida en el codificador 100 y es una señal que representa una imagen para cada instantánea incluida en una instantánea en movimiento (por ejemplo, una señal de luminancia y dos señales de crominancia). A continuación en el presente documento, una señal que representa una imagen también se denomina una muestra.

[Transformador]

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

40 Obsérvese que el transformador 106 puede seleccionar de forma adaptativa un tipo de transformada de entre una pluralidad de tipos de transformada, y transformar errores de predicción en coeficientes de transformada usando una función de base de transformada que corresponde al tipo de transformada seleccionado. Esta clase de transformada también se denomina transformada de múltiples núcleos explícita (EMT) o transformada múltiple adaptativa (AMT).

45 Los tipos de transformada incluyen, por ejemplo, DCT-II, DCT-V, DCT-VIII, DST-I y DST-VII. La Figura 3 es un gráfico que indica funciones de base de transformada para cada tipo de transformada. En la Figura 3, N indica el número de píxeles introducidos. 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.

50 La información que indica si aplicar tal EMT o AMT (denominada, por ejemplo, una bandera de AMT) y la información que indica el tipo de transformada seleccionado se señaliza a nivel de CU. Obsérvese que la señalización de tal información no necesita realizarse a nivel de CU y puede realizarse a otro nivel (por ejemplo, a nivel de secuencia, nivel de instantánea, nivel de corte, nivel de losa o nivel de CTU).

Además, el transformador 106 puede aplicar una transformada secundaria a los coeficientes de transformada (resultado de transformada). Una transformada secundaria de este tipo también se denomina transformada

secundaria adaptativa (AST) o transformada secundaria no separable (NSST). Por ejemplo, el transformador 106 aplica una transformada secundaria a cada subbloque (por ejemplo, cada subbloque de 4×4) incluido en el bloque de los coeficientes de transformada que corresponden a los errores de intra predicción. La información que indica si aplicar NSST y la información relacionada con la matriz de transformada usada en NSST se señalizan a nivel de CU. Obsérvese que la señalización de tal información no necesita realizarse a nivel de CU y puede realizarse a otro nivel (por ejemplo, a nivel de secuencia, nivel de instantánea, nivel de corte, nivel de losa o nivel de CTU).

En este punto, una transformada separable es un procedimiento en el que se realiza una transformada una pluralidad de veces realizando de forma separada una transformada para cada dirección de acuerdo con el número de dimensiones introducidas. Una transformada no separable es un procedimiento de realización de una transformada colectiva en la que dos o más dimensiones en una entrada multidimensional se consideran colectivamente como una única dimensión.

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

Además, similar a lo anterior, después de que un bloque de 4×4 introducido se considera como un único conjunto que incluye 16 componentes, una transformada que realiza una pluralidad de rotaciones de Givens en el conjunto (es decir, una transformada de Hypercube-Givens) es también un ejemplo de una transformada no separable.

20 [Cuantificador]

El cuantificador 108 cuantifica los coeficientes de transformada emitidos desde el transformador 106. Más específicamente, el cuantificador 108 explora, en un orden de exploración predeterminado, los coeficientes de transformada del bloque actual, y cuantifica los coeficientes de transformada explorados a base de parámetros de cuantificación (QP) que corresponden a los coeficientes de transformada. El cuantificador 108 emite, a continuación, los coeficientes de transformada cuantificados (a continuación en el presente documento denominados como coeficientes cuantificados) del bloque actual al codificador 110 por entropía y al cuantificador 112 inverso.

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

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

35 [Codificador por entropía]

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

40 [Cuantificador inverso]

El cuantificador 112 inverso cuantifica inversamente coeficientes cuantificados, que son entradas desde el cuantificador 108. Más específicamente, el cuantificador 112 inverso cuantifica inversamente, en un orden de exploración predeterminado, coeficientes cuantificados del bloque actual. El cuantificador 112 inverso emite, a continuación, los coeficientes de transformada cuantificados inversamente del bloque actual al transformador 114 inverso.

45 [Transformador inverso]

El transformador 114 inverso restaura errores de predicción transformando inversamente coeficientes de transformada, que son entradas desde el cuantificador 112 inverso. Más específicamente, el transformador 114 inverso restaura los errores de predicción del bloque actual aplicando una transformada inversa que corresponde a la transformada aplicada por el transformador 106 en los coeficientes de transformada. El transformador 114 inverso emite, a continuación, los errores de predicción restaurados al sumador 116.

Obsérvese que dado que la información se pierde en la cuantificación, los errores de predicción restaurados no coinciden con los errores de predicción calculados por el restador 104. En otras palabras, los errores de predicción restaurados incluyen errores de cuantificación.

[Sumador]

El sumador 116 reconstruye el bloque actual sumando errores de predicción, que son entradas desde el transformador 114 inverso, y muestras de predicción, que son entradas desde el controlador 128 de predicción. El sumador 116 emite, a continuación, el bloque reconstruido a la memoria 118 de bloque y al filtro 120 de bucle. Un bloque reconstruido también se denomina un bloque decodificado local.

[Memoria de bloque]

La memoria 118 de bloque es almacenamiento para almacenar bloques en una instantánea que hay que codificar (a continuación en el presente documento denominada una instantánea actual) para referencia en intra predicción. Más específicamente, la memoria 118 de bloque almacena bloques reconstruidos emitidos desde el sumador 116.

[Filtro de bucle]

El filtro 120 de bucle aplica un filtro de bucle a bloques reconstruidos por el sumador 116, y emite los bloques reconstruidos filtrados a la memoria 122 de fotograma. Un filtro de bucle es un filtro usado en un bucle de codificación (filtro en bucle) e incluye, por ejemplo, un filtro de desbloqueo (DF), una compensación adaptativa de muestra (SAO) y un filtro de bucle adaptativo (ALF).

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

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

Por ejemplo, la direccionalidad de gradiente D se calcula comparando gradientes de una pluralidad de direcciones (por ejemplo, las direcciones horizontal, vertical y las dos diagonales). Además, por ejemplo, la actividad de gradiente A se calcula sumando gradientes de una pluralidad de direcciones y cuantificando la suma.

El filtro que hay que usar para cada subbloque se determina de entre la pluralidad de filtros a base del resultado de tal categorización.

La forma de filtro que hay que usar en ALF es, por ejemplo, una forma de filtro simétrica circular. La Figura 4A a la Figura 4C ilustran ejemplos de formas de filtro usadas en ALF. La Figura 4A ilustra un filtro con forma de diamante de 5×5 , la Figura 4B ilustra un filtro con forma de diamante de 7×7 y la Figura 4C ilustra un filtro con forma de diamante de 9×9 . La información que indica la forma de filtro se señaliza a nivel de instantánea. Obsérvese que la señalización de información que indica la forma de filtro no necesita realizarse a nivel de instantánea y puede realizarse a otro nivel (por ejemplo, a nivel de secuencia, nivel de corte, nivel de losa, nivel de CTU o nivel de CU).

La habilitación o deshabilitación de ALF se determina a nivel de instantánea o nivel de CU. Por ejemplo, para luminancia, la decisión de aplicar ALF o no se hace a nivel de CU y, para crominancia, la decisión de aplicar ALF o no se hace a nivel de instantánea. La información que indica si se habilita o deshabilita ALF se señaliza a nivel de instantánea o nivel de CU. Obsérvese que la señalización de información que indica si se habilita o deshabilita ALF no necesita realizarse a nivel de instantánea o nivel de CU y puede realizarse a otro nivel (por ejemplo, a nivel de secuencia, nivel de corte, nivel de losa o nivel de CTU).

Los coeficientes establecidos para la pluralidad de filtros seleccionables (por ejemplo, 15 o 25 filtros) se señaliza a nivel de instantánea. Obsérvese que la señalización de los coeficientes establecidos no necesita realizarse a nivel de instantánea y puede realizarse a otro nivel (por ejemplo, a nivel de secuencia, nivel de corte, nivel de losa, nivel de CTU, nivel de CU o nivel de subbloque).

[Memoria de fotograma]

La memoria 122 de fotograma es almacenamiento para almacenar instantáneas de referencia usadas en intra predicción y también se denomina una memoria intermedia de fotograma. Más específicamente, la memoria 122 de fotograma almacena bloques reconstruidos filtrados por el filtro 120 de bucle.

[Intra Predictor]

El intra predictor 124 genera una señal de predicción (señal de intra predicción) intra prediciendo el bloque

actual con referencia a un bloque o bloques en la instantánea actual y almacenada en la memoria 118 de bloque (también denominada intra predicción de trama). Más específicamente, el intra predictor 124 genera una señal de intra predicción intra prediciendo con referencia a muestras (por ejemplo, valores de luminancia y/o cromas) de un bloque o bloques vecinos al bloque actual y emite, a continuación, la señal de intra predicción al controlador 128 de predicción.

Por ejemplo, el intra predictor 124 realiza intra predicción usando 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.

El uno o más modos de predicción no direccionales incluyen, por ejemplo, modo de predicción plana y modo de predicción DC definidos en la norma H.265/Codificación de Vídeo de Alta Eficiencia (HEVC) (véase NPL 1).

La pluralidad de modos de predicción direccionales incluyen, por ejemplo, los 33 modos de predicción direccionales definidos en la norma H.265/HEVC. Obsérvese que la pluralidad de modos de predicción direccionales pueden incluir adicionalmente 32 modos de predicción direccionales además de los 33 modos de predicción direccionales (para un total de 65 modos de predicción direccionales). La Figura 5A ilustra 67 modos de intra predicción usados en intra predicción (dos modos de predicción no direccionales y 65 modos de predicción direccionales). Las flechas continuas representan las 33 direcciones definidas en la norma H.265/HEVC y las flechas discontinuas representan las 32 direcciones direccionales.

Obsérvese que un bloque de luminancia puede hacerse referencia en intra predicción de bloque de crominancia. En otras palabras, un componente de crominancia del bloque actual puede predecirse a base de un componente de luminancia del bloque actual. Tal intra predicción también se denomina predicción de modelo lineal de componentes cruzados (CCLM). Un modo de intra predicción de bloque de crominancia de este tipo que hace referencia un bloque de luminancia (denominado, por ejemplo, modo de CCLM) puede añadirse como uno de los modos de intra predicción de bloque de crominancia.

El intra predictor 124 puede corregir los valores de píxeles posteriores a intra predicción a base de gradientes de píxel de referencia horizontales/verticales. La intra predicción acompañada por esta clase de corrección también se denomina combinación de intra predicción dependiente de posición (PDPC). La información que indica si aplicar PDPC o no (denominada, por ejemplo, una bandera de PDPC) se señaliza, por ejemplo, a nivel de CU. Obsérvese que la señalización de esta información no necesita realizarse a nivel de CU y puede realizarse a otro nivel (por ejemplo, a nivel de secuencia, nivel de instantánea, nivel de corte, nivel de losa o nivel de CTU).

[Inter Predictor]

El inter predictor 126 genera una señal de predicción (señal de inter predicción) inter prediciendo el bloque actual con referencia a un bloque o bloques en una instantánea de referencia, que es diferente de la instantánea actual y se almacena en la memoria 122 de fotograma (también denominada inter predicción de trama). La inter predicción se realiza por bloque actual o por subbloque (por ejemplo, por bloque de 4×4) en el bloque actual. Por ejemplo, el inter predictor 126 realiza estimación de movimiento en una instantánea de referencia para el bloque actual o subbloque. El inter predictor 126 genera, a continuación, una señal de inter predicción del bloque actual o subbloque por compensación de movimiento usando información de movimiento (por ejemplo, un vector de movimiento) obtenido a partir de la estimación de movimiento. El inter predictor 126 emite, a continuación, la señal de inter predicción generada al controlador 128 de predicción.

Se señaliza la información de movimiento usada en compensación de movimiento. Puede usarse un predictor de vector de movimiento para la señalización del vector de movimiento. En otras palabras, puede señalizarse la diferencia entre el vector de movimiento y el predictor de vector de movimiento.

Obsérvese que la señal de inter predicción puede generarse usando información de movimiento para un bloque vecino además de información de movimiento para el bloque actual obtenido a partir de la estimación de movimiento. Más específicamente, la señal de inter predicción puede generarse por subbloque en el bloque actual calculando una suma ponderada de una señal de predicción a base de información de movimiento obtenida a partir de la estimación de movimiento y una señal de predicción a base de información de movimiento para un bloque vecino. Tal inter predicción (compensación de movimiento) también se denomina compensación de movimiento de bloques solapados (OBMC).

En un modo de OBMC de este tipo, la información que indica tamaño de subbloque para OBMC (denominado, por ejemplo, tamaño de bloque de OBMC) se señaliza a nivel de secuencia. Además, la información que indica si aplicar el modo de OBMC o no (denominada, por ejemplo, una bandera de OBMC) se señaliza a nivel de CU. Obsérvese que la señalización de tal información no necesita realizarse a nivel de secuencia y nivel de CU y puede realizarse a otro nivel (por ejemplo, a nivel de instantánea, nivel de corte, nivel de losa, nivel de CTU o nivel de subbloque).

A continuación en el presente documento, el modo de OBMC se describirá en detalle adicional. La Figura 5B

es un diagrama de flujo y la Figura 5C es un diagrama conceptual para ilustrar un esquema de un procedimiento de corrección de imagen de predicción realizado a través de procesamiento de OBMC.

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

- 5 A continuación, se obtiene una imagen de predicción (Pred_L) aplicando un vector de movimiento (MV_L) del bloques izquierdo vecino codificado al bloque actual, y se hace un primer pase de la corrección de la imagen de predicción superponiendo la imagen de predicción y Pred_L.

De manera similar, se obtiene una imagen de predicción (Pred_U) aplicando un vector de movimiento (MV_U) del bloques superior vecino codificado al bloque actual, y se hace un segundo pase de la corrección de la imagen de predicción superponiendo la imagen de predicción que resulta del primer pase y Pred_U. El resultado del segundo pase es la imagen de predicción final.

10 Obsérvese que el ejemplo anterior es de un procedimiento de corrección de dos pasos usando los bloques izquierdos y superiores vecinos, pero el procedimiento puede ser un procedimiento de corrección de tres o más pasos que también usa el bloques derecho y/o inferior vecino.

- 15 Obsérvese que la región sujeta a la superposición puede ser toda la región de píxel del bloques y, como alternativa, puede ser una región límite de bloques parcial.

Obsérvese que en este punto, el procedimiento de corrección de imagen de predicción se describe como que es una base de una única instantánea de referencia, pero lo mismo se aplica cuando una imagen de predicción se corrige a base de una pluralidad de instantáneas de referencia. En un caso de este tipo, después de que se obtienen imágenes de predicción corregidas resultantes de la realización de una corrección a base de cada de las instantáneas de referencia, las imágenes de predicción corregidas obtenidas se superponen adicionalmente para obtener la imagen de predicción final.

Obsérvese que la unidad del bloques actual puede ser un bloques de predicción y, como alternativa, puede ser un subbloque obtenido dividiendo adicionalmente el bloques de predicción.

- 25 Un ejemplo de un procedimiento para determinar si implementar un procesamiento de OBMC es usar una obmc_flag, que es una señal que indica si implementar un procesamiento de OBMC. Como un ejemplo específico, el codificador determina si el bloques actual pertenece a una región que incluye movimiento complicado. El codificador establece la obmc_flag a un valor de "1" cuando el bloques pertenece a una región que incluye movimiento complicado e implementa un procesamiento de OBMC cuando codifica, y establece la obmc_flag a un valor de "0" cuando el bloques no pertenece a una región que incluye movimiento complicado y codifica sin implementar un procesamiento de OBMC. El decodificador conmuta entre implementar o no procesamiento de OBMC decodificando la obmc_flag escrita en el flujo y realizando la decodificación de acuerdo con el valor de bandera.

35 Obsérvese que la información de movimiento puede derivarse en el lado de decodificador sin señalizarse. Por ejemplo, puede usarse un modo de fusión definido en la norma H.265/HEVC. Además, por ejemplo, la información de movimiento puede derivarse realizando una estimación de movimiento en el lado de decodificador. En este caso, se realiza estimación de movimiento sin usar los valores de píxeles del bloques actual.

- 40 En este punto, se describirá un modo para realizar una estimación de movimiento en el lado de decodificador. Un modo para realizar estimación de movimiento en el lado de decodificador también se denomina modo de derivación de vector de movimiento con patrón coincidente (PMMVD) o modo de conversión ascendente de tasa de trama (FRUC).

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

- 50 A continuación, se deriva un vector de movimiento para el bloques actual a partir del vector de movimiento del candidato seleccionado. Más específicamente, por ejemplo, el vector de movimiento para el bloques actual se calcula como el vector de movimiento del candidato seleccionado (mejor MV candidato), tal cual. Como alternativa, el vector de movimiento para el bloques actual puede derivarse mediante concordancia de patrones realizada en la vecindad de una posición en una instantánea de referencia que corresponde al vector de movimiento del candidato seleccionado. En otras palabras, cuando la vecindad del mejor MV candidato se busca a través del mismo procedimiento y se encuentra un MV que tiene un mejor valor de evaluación, el mejor

MV candidato puede actualizarse al MV que tiene el mejor valor de evaluación y el MV que tiene el mejor valor de evaluación puede usarse como el MV final para el bloque actual. Obsérvese que también es aceptable una configuración en la que no se implementa este procesamiento.

5 Pueden realizarse los mismos procedimientos en casos en los que el procesamiento se realiza en unidades de subbloques.

Obsérvese que se calcula un valor de evaluación calculando la diferencia en la imagen reconstruida mediante concordancia de patrones realizada entre una región en una instantánea de referencia que corresponde a un vector de movimiento y una región predeterminada. Obsérvese que el valor de evaluación puede calcularse usando alguna otra información además de la diferencia.

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

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

20 La Figura 6 es para ilustrar un ejemplo de concordancia de patrones (concordancia bilateral) entre dos bloques a lo largo de una trayectoria de movimiento. Como se ilustra en la Figura 6, en la primera concordancia de patrones, se derivan dos vectores de movimiento (MV0, MV1) encontrando la mejor concordancia entre dos bloques a lo largo de la trayectoria de movimiento del bloque actual (Bloque actual) en dos instantáneas de referencia diferentes (Ref0, Ref1). Más específicamente, puede derivarse una diferencia entre (i) una imagen reconstruida en una posición específica en una primera instantánea de referencia codificada (Ref0) especificada por un MV candidato y (ii) una instantánea reconstruida en una posición específica en una segunda instantánea de referencia codificada (Ref1) especificada por un MV simétrico escalado en un intervalo de tiempo de visualización del MV candidato, y el valor de evaluación para el bloque actual puede calcularse usando la diferencia derivada. El MV candidato que tiene el mejor valor de evaluación entre la pluralidad de MV candidatos puede seleccionarse como el MV final.

30 Bajo la suposición de 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 instantánea actual (Instantánea actual) y las dos instantáneas de referencia (Ref0, Ref1). Por ejemplo, cuando la instantánea actual está temporalmente entre las dos instantáneas de referencia y la distancia temporal desde la instantánea actual hasta las dos instantáneas de referencia es la misma, la primera concordancia de patrones deriva un vector de movimiento bidireccional basado en espejo.

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

40 La Figura 7 es para ilustrar un ejemplo de concordancia de patrones (concordancia de plantilla) entre una plantilla en la instantánea actual y un bloque en una instantánea de referencia. Como se ilustra en la Figura 7, en la segunda concordancia de patrones, se deriva un vector de movimiento del bloque actual buscando una instantánea de referencia (Ref0) para encontrar el bloque que mejor coincide con bloques vecinos del bloque actual (Bloque actual) en la instantánea actual (Instantánea actual). Más específicamente, puede derivarse una diferencia entre (i) una imagen reconstruida de una región codificada que es ambas o una de la región vecina izquierda y vecina superior y (ii) una instantánea reconstruida en la misma posición en una instantánea de referencia codificada (Ref0) especificada por un MV candidato, y el valor de evaluación para el bloque actual puede calcularse usando la diferencia derivada. El MV candidato que tiene el mejor valor de evaluación entre la pluralidad de MV candidatos puede seleccionarse como el mejor MV candidato.

50 La información que indica si aplicar el modo de FRUC o no (denominado, por ejemplo, una bandera de FRUC) se señaliza a nivel de CU. Además, cuando se aplica el modo de FRUC (por ejemplo, cuando la bandera de FRUC se establece a verdadero), la información que indica el procedimiento de concordancia de patrones (primera concordancia de patrones o segunda concordancia de patrones) se señaliza a nivel de CU. Obsérvese que la señalización de tal información no necesita realizarse a nivel de CU y puede realizarse a otro nivel (por ejemplo, a nivel de secuencia, nivel de instantánea, nivel de corte, nivel de losa, nivel de CTU o nivel de subbloque).

En este punto, se describirá un modo para derivar un vector de movimiento a base de un modelo que supone movimiento lineal uniforme. Este modo también se denomina un modo de flujo óptimo bidireccional (BIO).

La Figura 8 es para ilustrar un modelo que supone movimiento lineal uniforme. En la Figura 8, (v_x, v_y) indica un vector de velocidad, y t_0 y t_1 indican distancias temporales entre la instantánea actual (Instantánea actual) y dos instantáneas de referencia (Ref₀, Ref₁). (MVx_0, MVy_0) indica un vector de movimiento que corresponde a la instantánea de referencia Ref₀ y (MVx_1, MVy_1) indica un vector de movimiento que corresponde a la instantánea de referencia Ref₁.

En este punto, bajo la suposición de movimiento lineal uniforme exhibida por el vector de velocidad (v_x, v_y) , (MVx_0, MVy_0) y (MVx_1, MVy_1) se representan como (v_{xT_0}, v_{yT_0}) y $(-v_{xT_1}, -v_{yT_1})$, respectivamente, y se proporciona la siguiente ecuación de flujo óptico.

CÁLCULO. 1

10 $\partial I^{(k)} / \partial t + v_x \partial I^{(k)} / \partial x + v_y \partial I^{(k)} / \partial y = 0. \quad (1)$

En este punto, $I^{(k)}$ indica a valor de luminancia de instantánea de referencia k ($k = 0, 1$) después de compensación de movimiento. Esta ecuación de flujo óptico muestra que la suma de (i) la derivada de tiempo del valor de luminancia, (ii) el producto de la velocidad horizontal y el componente horizontal del gradiente espacial de una instantánea de referencia y (iii) el producto de la velocidad vertical y el componente vertical del gradiente espacial de una instantánea de referencia es igual a cero. Un vector de movimiento de cada bloque obtenido a partir de, por ejemplo, una lista de fusión se corrige píxel a píxel a base de una combinación de la ecuación de flujo óptico e interpolación de Hermite.

20 Obsérvese que un vector de movimiento puede derivarse en el lado de decodificador usando un procedimiento distinto de derivar un vector de movimiento a base de un modelo que supone movimiento lineal uniforme. Por ejemplo, un vector de movimiento puede derivarse para cada subbloque a base de vectores de movimiento de bloques vecinos.

En este punto, se describirá un modo en el que se deriva un vector de movimiento para cada subbloque a base de vectores de movimiento de bloques vecinos. Este modo también se denomina modo de predicción de compensación de movimiento afín.

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

CÁLCULO. 2

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

35 En este punto, x e y son las posiciones horizontales y verticales del subbloque, respectivamente, y w es un coeficiente ponderado predeterminado.

Un modo de predicción de compensación de movimiento afín de este tipo puede incluir un número de modos de diferentes procedimientos de derivación de los vectores de movimiento de los puntos de control de esquina superior izquierda y superior derecha. La información que indica un modo de predicción de compensación de movimiento afín de este tipo (denominada, por ejemplo, una bandera afín) se señala a nivel de CU. Obsérvese que la señalización de información que indica el modo de predicción de compensación de movimiento afín no necesita realizarse a nivel de CU y puede realizarse a otro nivel (por ejemplo, a nivel de secuencia, nivel de instantánea, nivel de corte, nivel de losa, nivel de CTU o nivel de subbloque).

[Controlador de predicción]

45 El controlador 128 de predicción selecciona o bien la señal de intra predicción o bien la señal de inter predicción, y emite la señal de predicción seleccionada al restador 104 y al sumador 116.

En este punto, se proporcionará un ejemplo de derivación de un vector de movimiento a través de modo de fusión en una instantánea actual. La Figura 9B es para ilustrar un esquema de un procedimiento de derivación de un vector de movimiento a través de modo de fusión.

50 En primer lugar, se genera una lista de predictores de MV en la que se registran predictores de MV candidatos. Ejemplos de predictores de MV candidatos incluyen: predictor de MV espacialmente vecinos, que son MV de bloques codificados colocados en la vecindad espacial del bloque actual; un predictor de MV temporalmente

vecino, que es un MV de un bloque en una instantánea 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 combinando los valores de MV del predictor de MV espacialmente vecino y el predictor de MV temporalmente vecino; y un predictor de MV cero, que es un MV cuyo valor es cero.

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

Adicionalmente, en el codificador de longitud variable, un merge_idx, que es una señal que indica qué predictor de MV se selecciona, se escribe y codifica en el flujo.

- 10 Obsérvese que los predictores de MV registrados en la lista de predictores de MV ilustrados en la Figura 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 Figura 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 proporcionados en el ejemplo en la Figura 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 proporcionados en el ejemplo en la Figura 9B.

- 15 Obsérvese que el MV final puede determinarse realizando procesamiento de DMVR (que se describirá posteriormente) usando el MV del bloque actual derivado a través del modo de fusión.

En este punto, se proporcionará un ejemplo de determinación de un MV usando procesamiento de DMVR.

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

- 20 En primer lugar, se considera que el MVP más apropiado establecido para el bloque actual es el MV candidato, los píxeles de referencia se obtienen de una primera instantánea de referencia, que es una instantánea procesada en la dirección L0 de acuerdo con el MV candidato, y una segunda instantánea de referencia, que es una instantánea procesada en la dirección L1 de acuerdo con el MV candidato, y se genera una plantilla calculando el promedio de los píxeles de referencia.

- 25 A continuación, usando la plantilla, se buscan las regiones circundantes de los MV candidatos de la primera y la segunda instantáneas de referencia y se determina que el MV con el menor coste sea el MV final. Obsérvese que el valor de coste se calcula usando, por ejemplo, la diferencia entre cada valor de píxel en la plantilla y cada valor de píxel en las regiones buscadas, así como el valor de MV.

Obsérvese que los esquemas de los procedimientos descritos en este punto son fundamentalmente los mismos tanto en el codificador como el decodificador.

- 30 Obsérvese que puede usarse un procesamiento distinto del procesamiento exactamente como se describe anteriormente, siempre que el procesamiento sea capaz de derivar el MV final buscando los alrededores del MV candidato.

En este punto, se proporcionará un ejemplo de un modo que genera una imagen de predicción usando procesamiento de LIC.

- 35 La Figura 9D es para ilustrar un esquema de un procedimiento de generación de imágenes de predicción usando un procedimiento de corrección de luminancia realizado a través de procesamiento de LIC.

En primer lugar, se extrae un MV para obtener, de una instantánea de referencia codificada, una imagen de referencia que corresponde al bloque actual.

- 40 A continuación, se extrae la información que indica cómo cambió el valor de luminancia entre la instantánea de referencia y la instantánea actual y se calcula un parámetro de corrección de luminancia usando los valores de luminancia de píxel 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 instantánea de referencia especificada por el MV.

- 45 La imagen de predicción para el bloque actual se genera realizando un procedimiento de corrección de luminancia usando el parámetro de corrección de luminancia en la imagen de referencia en la instantánea de referencia especificada por el MV.

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

- 50 Además, aunque una imagen de predicción se genera a partir de una única instantánea de referencia en este ejemplo, en casos en los que una imagen de predicción se genera también a partir de una pluralidad de instantáneas de referencia, la imagen de predicción se genera después de realizar un procedimiento de corrección de luminancia, a través del mismo procedimiento, en las imágenes de referencia obtenidas de las

instantáneas de referencia.

Un ejemplo de un procedimiento para determinar si implementar procesamiento de LIC es usando una lic_flag, que es una señal que indica si implementar el procesamiento de LIC. Como un ejemplo específico, el codificador determina si el bloque actual pertenece a una región de cambio de luminancia. El codificador establece la lic_flag a un valor de "1" cuando el bloque pertenece a una región de cambio de luminancia e implementa el procesamiento de LIC cuando codifica, y establece la 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 de LIC. El decodificador conmuta entre implementar o no el procesamiento de LIC decodificando la lic_flag escrita en el flujo y realizando la decodificación de acuerdo con el valor de bandera.

- 5 Un ejemplo de un procedimiento diferente de determinación de si implementar el procesamiento de LIC es determinar el mismo de acuerdo con si el procesamiento de LIC se determinó para implementarse para un bloques circundante. En un ejemplo específico, cuando se usa el modo de fusión en el bloque actual, puede determinarse si se aplicó procesamiento de LIC en la codificación de en bloques codificado circundante seleccionado tras derivar el MV en el procesamiento de modo de fusión, y si implementar o no procesamiento de LIC puede conmutarse a base del resultado de la determinación. Obsérvese que en este ejemplo, lo mismo se aplica al procesamiento realizado en el lado de decodificador.
- 10
- 15

[Esquema de decodificador]

20 A continuación, se describirá un decodificador con capacidad de decodificar una señal codificada (flujo de bits codificado) emitida desde el codificador 100. La Figura 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 instantánea en movimiento/instantánea que decodifica una instantánea en movimiento/instantánea bloque a bloque.

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

30 El decodificador 200 se realiza como, por ejemplo, un procesador y una memoria genéricos. En este caso, cuando el procesador ejecuta un programa de software almacenado en la memoria, el procesador funciona como el decodificador 202 por entropía, el cuantificador 204 inverso, el transformador 206 inverso, el sumador 208, el filtro 212 de bucle, el intra predictor 216, el inter predictor 218 y el controlador 220 de predicción. Como alternativa, el decodificador 200 puede realizarse como uno o más circuitos electrónicos especializados que corresponden al decodificador 202 por entropía, al cuantificador 204 inverso, al transformador 206 inverso, al sumador 208, al filtro 212 de bucle, al intra predictor 216, al inter predictor 218 y al controlador 220 de predicción.

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

35 [Decodificador por entropía]

El decodificador 202 por entropía decodifica por entropía un flujo de bits codificado. Más específicamente, por ejemplo, la aritmética del decodificador 202 por entropía decodifica un flujo de bits codificado en una señal binaria. El decodificador 202 por entropía binariza, a continuación, la señal binaria. Con esto, el decodificador 202 por entropía emite coeficientes cuantificados de cada bloque al cuantificador 204 inverso.

40 [Cuantificador inverso]

El cuantificador 204 inverso cuantifica inversamente coeficientes cuantificados de un bloque que hay que decodificar (a continuación en el presente documento denominado un bloque actual), que son entradas desde el decodificador 202 por entropía. Más específicamente, el cuantificador 204 inverso cuantifica inversamente coeficientes cuantificados del bloque actual a base de parámetros de cuantificación que corresponden a los coeficientes cuantificados. El cuantificador 204 inverso emite, a continuación, los coeficientes cuantificados inversos (es decir, coeficientes de transformada) del bloque actual al transformador 206 inverso.

[Transformador inverso]

El transformador 206 inverso restaura errores de predicción transformando inversamente coeficientes de transformada, que son entradas desde el cuantificador 204 inverso.

50 Por ejemplo, cuando la información analizada de un flujo de bits codificado indica la aplicación de EMT o AMT (por ejemplo, cuando la bandera de AMT se establece a verdadero), el transformador 206 inverso transforma inversamente los coeficientes de transformada del bloque actual a base de información que indica el tipo de transformada analizada.

Además, por ejemplo, cuando la información analizada de un flujo de bits codificado indica la aplicación de

NSST, el transformador 206 inverso aplica una transformada inversa secundaria a los coeficientes de transformada.

[Sumador]

- 5 El sumador 208 reconstruye el bloque actual sumando errores de predicción, que son entradas desde el transformador 206 inverso, y muestras de predicción, que es una entrada desde el controlador 220 de predicción. El sumador 208 emite, a continuación, el bloque reconstruido a la memoria 210 de bloque y al filtro 212 de bucle.

[Memoria de bloque]

- 10 La memoria 210 de bloque es almacenamiento para almacenar bloques en una instantánea que hay que decodificar (a continuación en el presente documento denominada una instantánea actual) para su referencia en intra predicción. Más específicamente, la memoria 210 de bloque almacena bloques reconstruidos emitidos desde el sumador 208.

[Filtro de bucle]

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

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

[Memoria de fotograma]

- 20 La memoria 214 de fotograma es almacenamiento para almacenar instantáneas de referencia usadas en intra predicción y también se denomina una memoria intermedia de fotograma. Más específicamente, la memoria 214 de fotograma almacena bloques reconstruidos filtrados por el filtro 212 de bucle.

[Intra Predictor]

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

- 30 Obsérvese que cuando se selecciona un modo de intra predicción en el que un bloque de crominancia se intra predice a partir de un bloque de luminancia, el intra predictor 216 puede predecir el componente de crominancia del bloque actual a base del componente de luminancia del bloque actual.

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

35 [Inter Predictor]

- El inter predictor 218 predice el bloque actual con referencia a una instantánea de referencia almacenada en la memoria 214 de fotograma. La inter predicción se realiza por bloque actual o por subbloque (por ejemplo, por bloque de 4×4) en el bloque actual. Por ejemplo, el inter predictor 218 genera una señal de inter predicción del bloque actual o subbloque mediante compensación de movimiento usando información de movimiento (por ejemplo, un vector de movimiento) analizada de un flujo de bits codificado, y emite la señal de inter predicción al controlador 220 de predicción.

- 40 Obsérvese que cuando la información analizada del flujo de bits codificado indica la aplicación del modo de OBMC, el inter predictor 218 genera la señal de inter predicción usando información de movimiento para un bloque vecino además de información de movimiento para el bloque actual obtenido a partir de la estimación de movimiento.

- 45 Además, cuando la información analizada del flujo de bits codificado indica aplicación del modo de FRUC, el inter predictor 218 deriva información de movimiento realizando estimación de movimiento de acuerdo con el procedimiento de concordancia de patrones (concordancia bilateral o concordancia de plantilla) analizada del flujo de bits codificado. El inter predictor 218 realiza, a continuación, compensación de movimiento usando la información de movimiento derivada.

50 Además, cuando tiene que aplicarse el modo de BIO, el inter predictor 218 deriva un vector de movimiento a

base de un modelo que supone movimiento lineal uniforme. Además, cuando la información analizada del flujo de bits codificado indica que tiene que aplicarse el modo de predicción de compensación de movimiento afín, el inter predictor 218 deriva un vector de movimiento de cada subbloque a base de vectores de movimiento de bloques vecinos.

5 [Controlador de predicción]

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

REALIZACIÓN 2

A continuación, se describirá la realización 2. En un aspecto de la presente realización, se describirán en detalle la transformada y la transformada inversa. Obsérvese que un codificador y un decodificador de acuerdo con la presente realización tienen sustancialmente las mismas configuraciones que las del codificador y del decodificador de acuerdo con la realización 1. Como tal, se omitirán las ilustraciones y descripciones de los mismos.

[Procesamiento del transformador y del cuantificador del codificador]

15 En primer lugar, el procesamiento del transformador 106 y del cuantificador 108 del codificador 100 de acuerdo con la presente realización se describirá específicamente con referencia a la Figura 11. La Figura 11 es un diagrama de flujo que ilustra el procesamiento de transformación y cuantificación realizado por el codificador 100 de acuerdo con la realización 2.

20 En primer lugar, el transformador 106 selecciona una primera base de transformada para un bloque actual que hay que codificar de entre uno o más primeros candidatos de base de transformada (S101). Por ejemplo, el transformador 106 selecciona de forma fija una base de transformada de DCT-II como la primera base de transformada para el bloque actual. El transformador 106 puede seleccionar la primera base de transformada usando un modo de selección de base adaptativo, por ejemplo.

25 El modo de selección de base adaptativo es un modo en el que se selecciona de forma adaptativa una base de transformada de entre una pluralidad de candidatos de base de transformada predeterminados sobre la base de un coste que se basa en una cantidad de código y/o la diferencia entre una imagen original y una imagen reconstruida. El modo de selección de base adaptativo también se denomina un modo de EMT o un modo de AMT. Por ejemplo, la pluralidad de bases de transformada ilustradas en la Figura 6 pueden usarse como la pluralidad de candidatos de base de transformada. Obsérvese que la pluralidad de candidatos de base 30 de transformada no se limitan a la pluralidad de bases de transformada ilustradas en la Figura 6. La pluralidad de candidatos de base de transformada pueden incluir una base de transformada equivalente a no transformada, por ejemplo.

35 Codificando, en un flujo de bits, la información de identificación que indica cuál del modo de selección de base adaptativo y un modo de base fija, en el que se usa una base de transformada fija (por ejemplo, la base de DCT de tipo II), se habilita, es posible usar de forma selectiva el modo de selección de base adaptativo y el modo de base fijo. La información de identificación corresponde a información de identificación que indica si se habilita el modo de selección de base adaptativo. En un caso de este tipo, existen casos en los que la información de identificación permite la determinación en cuanto a si la primera base de transformada es la misma que una base de transformada predeterminada. Por ejemplo, con la EMT, dado que existe información 40 de identificación (emt_cu_flag) que indica, en unidades, tales como CU, cuál del modo de selección de base adaptativo y del modo de base fijo se habilita, es posible determinar, usando la información de identificación, si la primera base de transformada es la misma que una base de transformada predeterminada.

45 El transformador 106 realiza, a continuación, una primera transformada en el residual del bloque actual usando la primera base de transformada seleccionada en la etapa S102, para generar primeros coeficientes de transformada (S102). La primera transformada corresponde a la transformada primaria.

50 El transformador 106 determina si la primera base de transformada seleccionada en la etapa S101 es la misma que una base de transformada predeterminada (S103). Por ejemplo, el transformador 106 determina si la primera base de transformada es la misma que una cualquiera de la pluralidad de bases de transformada predeterminadas. El transformador 106 puede determinar si la primera base de transformada es la misma que una base de transformada predeterminada, por ejemplo.

Por ejemplo, puede usarse una base de transformada de la DCT de tipo II (es decir, DCT-II) y/o una base de transformada similar como la base de transformada predeterminada. Una base de transformada predeterminada de este tipo puede predefinirse por una norma, por ejemplo. Por ejemplo, la base de transformada predeterminada puede determinarse a base de un parámetro de codificación o similar.

55 En este punto, cuando la primera base de transformada es la misma que la base de transformada

predeterminada (Sí en S103), el transformador 106 selecciona una segunda base de transformada para el bloque actual de entre uno o más segundos candidatos de base de transformada (S104). El transformador 106 realiza una segunda transformada en los primeros coeficientes de transformada usando la segunda base de transformada seleccionada, para generar segundos coeficientes de transformada (S105). La segunda transformada corresponde a la transformada secundaria. El cuantificador 108 cuantifica los segundos coeficientes de transformada generados (S106) y finaliza el procesamiento de transformación y cuantificación.

En la segunda transformada, puede realizarse una transformada secundaria conocida como una NSST o puede realizarse una transformada que usa de forma selectiva una segunda base de transformada entre la pluralidad de segundos candidatos de base de transformada. En este momento, en la selección de una segunda base de transformada, la base de transformada que hay que seleccionar puede ser fija. En otras palabras, puede seleccionarse una base de transformada fija predeterminada como la segunda base de transformada. Puede usarse una base de transformada equivalente a ninguna segunda transformada como la segunda base de transformada.

La NSST puede ser una transformada de frecuencia-espacial que sigue la DCT o DST. Por ejemplo, la NSST puede ser una transformada de Karhunen Loveve (KLT) realizada en los coeficientes de transformada de la DCT o DST obtenida fuera de línea o una transformada de Hypercube-Givens (HyGT) que representa una base equivalente a la de KLT y se representa mediante una combinación de transformadas de rotación.

Por otra parte, cuando la primera base de transformada es diferente de la base de transformada predeterminada (No en S103), el transformador 106 omite la selección de una segunda base de transformada (S104) y la segunda transformada (S105). En otras palabras, el transformador 106 no realiza la segunda transformada. En un caso de este tipo, se cuantifican los primeros coeficientes de transformada generados en la etapa S207 (S106) y finaliza el procesamiento de transformación y cuantificación.

Cuando la segunda transformada se omite como en el caso anterior, la información que indica que la segunda transformada no se realiza puede notificarse al decodificador. Cuando se omite la segunda transformada, puede realizarse una segunda transformada usando una segunda base de transformada equivalente a ninguna transformada, e información que indica una segunda base de transformada de este tipo puede notificarse al decodificador.

Obsérvese que el cuantificador 112 inverso y el transformador 114 inverso del codificador 100 pueden reconstruir el bloque actual realizando inversamente el procesamiento realizado por el transformador 106 y el cuantificador 108.

[Procesamiento del cuantificador inverso y del transformador inverso del decodificador]

A continuación, se describirán específicamente el procesamiento del cuantificador 204 inverso y del transformador 206 inverso del decodificador 200 de acuerdo con la presente realización con referencia a la Figura 12. La Figura 12 es un diagrama de flujo que ilustra un procesamiento de cuantificación inversa y transformación inversa realizado por el decodificador 200 de acuerdo con la realización 2.

En primer lugar, el cuantificador 204 inverso cuantifica inversamente coeficientes cuantificados de un bloque actual que hay que decodificar (S601). El transformador 206 inverso determina si una primera base de transformada inversa para el bloque actual es la misma que una base de transformada inversa predeterminada (S602). Una base de transformada inversa que corresponde a la base de transformada predeterminada usada por el codificador 100 se usa como la base de transformada inversa predeterminada.

Cuando la primera base de transformada inversa es la misma que la base de transformada inversa predeterminada (Sí en S602), el transformador 206 inverso selecciona una segunda base de transformada inversa para el bloque actual (S603). Seleccionar una base de transformada inversa (la primera base de transformada inversa o la segunda base de transformada inversa) en el decodificador 200 es determinar una base de transformada inversa a base de información predeterminada. Por ejemplo, puede usarse una señal de selección de base como la información predeterminada. También puede usarse un modo de intra predicción o un tamaño de bloque, por ejemplo, como la información predeterminada.

El transformador 206 inverso realiza una segunda transformada inversa en los coeficientes cuantificados inversos del bloque actual usando la segunda base de transformada inversa seleccionada, para generar segundos coeficientes de transformada inversa (S604). Además, el transformador 206 inverso selecciona una primera base de transformada inversa (S605). Usando la primera base de transformada inversa seleccionada, el transformador 206 inverso realiza una primera transformada inversa en los segundos coeficientes de transformada inversa generados en S605 (S606) y finaliza el procesamiento de cuantificación inversa y transformación inversa.

Por otra parte, cuando la primera base de transformada inversa es diferente de la base de transformada inversa predeterminada (No en S602), el transformador 206 inverso omite la selección de una segunda base de transformada inversa (S603) y la segunda transformada inversa (S604). En otras palabras, el transformador

206 inverso no realiza la segunda transformada inversa y selecciona la primera base de transformada inversa (S605). Usando la primera base de transformada inversa seleccionada, el transformador 206 inverso realiza la primera transformada inversa en los coeficientes cuantificados inversamente en S501 (S606) y finaliza el procesamiento de cuantificación inversa y transformación inversa.

5 [Efectos ventajosos, etc.]

Los inventores han encontrado el problema de que la codificación convencional implica una cantidad enorme de procesamiento en la búsqueda de una combinación óptima de una base de transformada y un parámetro de transformada (por ejemplo, coeficientes de filtro) tanto en la primera transformada como en la segunda transformada. Por el contrario, el codificador 100 y el decodificador 200 de acuerdo con la presente realización

10 pueden omitir la segunda transformada de acuerdo con la primera base de transformada. Esto resulta en la reducción en el procesamiento de búsqueda de una combinación óptima de una base de transformada y un parámetro de transformada tanto en la primera transformada como en la segunda transformada, habilitando por lo tanto la reducción en la carga de procesamiento mientras inhibe un descenso en la eficiencia de compresión.

15 Como se describe anteriormente, el codificador 100 y el decodificador 200 de acuerdo con la presente realización pueden omitir la segunda transformada cuando la primera base de transformada es diferente de una base de transformada predeterminada. Los primeros coeficientes de transformada generados a través de la primera transformada se ven afectados por la primera base de transformada. Por lo tanto, la mejora en la tasa de compresión conseguida a través de la segunda transformada realizada en los primeros coeficientes de transformada a menudo depende de la primera base de transformada. Como tal, omitiendo la segunda transformada cuando la primera base de transformada es diferente de una base de transformada predeterminada que conduce a una mayor mejora en la tasa de compresión, es posible reducir la carga de procesamiento mientras inhibe un descenso en la eficiencia de compresión.

25 Con la DCT de tipo II en particular, dado que coeficientes de frecuencia significativa transformados a menudo se concentran en la banda de frecuencia baja, es probable que los efectos ventajosos de la segunda transformada sean mayores. En vista de esto, con el uso de la base de DCT de tipo II como la base de transformada predeterminada, se realiza la segunda transformada si la mejora en la eficiencia de compresión facilitada por la segunda transformada es significativa, y si no, se omite la segunda transformada. De este modo, se espera una reducción en la carga de procesamiento mientras se inhibe un descenso adicionalmente 30 en la eficiencia de compresión.

Obsérvese que el procesamiento anterior es aplicable tanto a una señal de luminancia como a una señal de crominancia, y también puede aplicarse a cada señal de R, G y B cuando la señal de entrada está en el formato RGB. Además, bases que son seleccionables en la primera transformada o la segunda transformada pueden ser diferentes entre la señal de luminancia y la señal de crominancia. Por ejemplo, la señal de luminancia tiene una banda de frecuencia más ancha que la banda de frecuencia de la señal de crominancia. Por lo tanto, en la transformada de la señal de luminancia, pueden ser seleccionables más tipos de bases que en la transformada de la señal de crominancia.

35 Obsérvese que el número de bases de transformada predeterminadas no se limita a uno. Es decir, puede haber una pluralidad de bases de transformada predeterminadas. En un caso de este tipo, la determinación en cuanto a si la primera base de transformada es la misma que una cualquiera de la pluralidad de bases de transformada es suficiente.

40 Obsérvese que este aspecto puede implementarse en combinación con uno o más de los otros aspectos de acuerdo con la presente divulgación. Además, parte de los procedimientos en los diagramas de flujo, parte de los elementos constituyentes de los aparatos y parte de la sintaxis descrita en este aspecto pueden 45 implementarse en combinación con otros aspectos.

REALIZACIÓN 3

A continuación, se describirá la realización 3. Un aspecto de la presente realización es diferente de la realización 2 en que el procesamiento de transformación difiere dependiendo de si se usa intra predicción para un bloque actual que hay que codificar/decodificar. A continuación en el presente documento, la presente 50 realización se describirá con referencia a los dibujos, centrándose en las diferencias de la realización 2. Obsérvese que a lo largo de los dibujos mencionados anteriormente, se proporcionan los mismos números de referencia a las etapas de procesamiento que son sustancialmente las mismas que la de la realización 2 y se omitirán o simplificarán las descripciones solapantes.

[Procesamiento del transformador y del cuantificador del codificador]

55 En primer lugar, el procesamiento del transformador 106 y del cuantificador 108 del codificador 100 de acuerdo con la presente realización se describirá específicamente con referencia a la Figura 13. La Figura 13 es un diagrama de flujo que ilustra el procesamiento de transformación y cuantificación realizado por el codificador

100 de acuerdo con la realización 3.

En primer lugar, el transformador 106 determina cuál de intra predicción e inter predicción tiene que usarse para un bloque actual que hay que codificar (S201). Por ejemplo, el transformador 106 determina cuál de intra predicción e inter predicción tiene que usarse, sobre la base de un coste que se basa en una cantidad de código y/o la diferencia entre una imagen original y una imagen reconstruida obtenida decodificando localmente una imagen comprimida. Por ejemplo, el transformador 106 puede determinar cuál de intra predicción e inter predicción tiene que usarse, sobre la base de información (por ejemplo, el tipo de instantánea) diferente del coste que se basa en la cantidad de código y/o la diferencia.

En este punto, cuando se determina usar inter predicción para el bloque actual (inter en S201), el transformador 106 selecciona una primera base de transformada para el bloque actual de entre uno o más primeros candidatos de base de transformada (S202). Por ejemplo, el transformador 106 selecciona de forma fija una base de transformada de DCT-II como la primera base de transformada para el bloque actual. El transformador 106 puede seleccionar la primera base de transformada de entre una pluralidad de primeros candidatos de base de transformada, por ejemplo.

El transformador 106 realiza, a continuación, la primera transformada en la residual del bloque actual usando la primera base de transformada seleccionada en la etapa S202, para generar los primeros coeficientes de transformada (S203). El cuantificador 108 cuantifica los primeros coeficientes de transformada generados (S204) y finaliza el procesamiento de transformación y cuantificación.

Por otra parte, cuando se determina usar intra predicción para el bloque actual (intra en S201), el transformador 106 realiza las etapas S101 a S105 de la misma manera que en la realización 2. A continuación, el cuantificador 108 cuantifica los primeros coeficientes de transformada generados en la etapa S102 o los segundos coeficientes de transformada generados en la etapa S105 (S204) y finaliza el procesamiento de transformación y cuantificación.

[Procesamiento del cuantificador inverso y del transformador inverso del decodificador]

A continuación, se describirán específicamente el procesamiento del cuantificador 204 inverso y del transformador 206 inverso del decodificador 200 de acuerdo con la presente realización con referencia a la Figura 14. La Figura 14 es un diagrama de flujo que ilustra un procesamiento de cuantificación inversa y transformación inversa realizado por el decodificador 200 de acuerdo con la realización 3.

En primer lugar, el cuantificador 204 inverso cuantifica inversamente coeficientes cuantificados de un bloque actual que hay que decodificar (S601). El transformador 206 inverso determina cuál de intra predicción e inter predicción tiene que usarse para el bloque actual (S701). Por ejemplo, el transformador 206 inverso determina cuál de intra predicción e inter predicción tiene que usarse, a base de información obtenida a partir de un flujo de bits.

En este punto, cuando se determina usar inter predicción para el bloque actual (inter en S701), el transformador 206 inverso selecciona una primera base de transformada inversa para el bloque actual (S702). Usando la primera base de transformada inversa seleccionada en S503, el transformador 206 inverso realiza la primera transformada inversa en los coeficientes cuantificados inversos del bloque actual (S703) y finaliza el procesamiento de cuantificación inversa y transformación inversa.

Por otra parte, cuando se determina usar intra predicción para el bloque actual (intra en S701), el transformador 206 inverso realiza las etapas S602 a S606 de la misma manera que en la realización 2 y finaliza el procesamiento de cuantificación inversa y transformación inversa.

[Efectos ventajosos, etc.]

El codificador 100 y el decodificador 200 de acuerdo con la presente realización pueden omitir la segunda transformada de acuerdo con intra/inter predicción y la primera base de transformada. Esto resulta en la reducción en el procesamiento de búsqueda de una combinación óptima de una base de transformada y un parámetro de transformada tanto en la primera transformada como en la segunda transformada, habilitando por lo tanto la reducción en la carga de procesamiento mientras inhibe un descenso en la eficiencia de compresión.

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

REALIZACIÓN 4

A continuación, se describirá la realización 4. Un aspecto de la presente realización es diferente de las

realizaciones 2 y 3 en que el procesamiento de transformación difiere de acuerdo con un modo de intra predicción para un bloque actual que hay que codificar/decodificar. A continuación en el presente documento, la presente realización se describirá con referencia a los dibujos, centrándose en las diferencias de las realizaciones 2 y 3. Obsérvese que a lo largo de los dibujos mencionados anteriormente, se proporcionan los mismos números de referencia a las etapas de procesamiento que son sustancialmente las mismas que las de la realización 2 o 3 y se omitirán o simplificarán las descripciones solapantes.

[Procesamiento del transformador y del cuantificador del codificador]

En primer lugar, el procesamiento del transformador 106 y del cuantificador 108 del codificador 100 de acuerdo con la presente realización se describirá específicamente con referencia a la Figura 15. La Figura 15 es un diagrama de flujo que ilustra el procesamiento de transformación y cuantificación realizado por el codificador 100 de acuerdo con la realización 4.

De la misma manera que en la realización 2, el transformador 106 determina cuál de intra predicción e inter predicción tiene que usarse para un bloque actual que hay que codificar (S201). En este punto, cuando se determina usar inter predicción para el bloque actual (inter en S201), el transformador 106 realiza la etapa S202 y la etapa S203 de la misma manera que en la realización 2. A continuación, el cuantificador 108 cuantifica los primeros coeficientes de transformada generados en la etapa S203 (S302).

Por otra parte, cuando se determina usar intra predicción para el bloque actual (intra en S201), el transformador 106 realiza la etapa S101 y la etapa S102 de la misma manera que en la realización 1. El transformador 106 determina, a continuación, si el modo de intra predicción para el bloque actual es un modo predeterminado (S106). Por ejemplo, el transformador 106 determina si el modo de intra predicción es un modo predeterminado sobre la base de un coste que se basa en una cantidad de código y/o la diferencia entre una imagen original y una imagen reconstruida. Obsérvese que la determinación en cuanto a si el modo de intra predicción es el modo predeterminado puede realizarse a base de información diferente del coste.

El modo predeterminado puede predefinirse por una norma, por ejemplo. El modo predeterminado puede determinarse a base de un parámetro de codificación, por ejemplo. Por ejemplo, puede usarse un modo de predicción direccional en una dirección diagonal como el modo predeterminado.

Los modos de predicción direccionales son modos de intra predicción en los que se usa una dirección particular para predecir un bloque actual. En modos de predicción direccionales, los valores de pixel se predicen extendiendo los valores de los píxeles de referencia en una dirección específica. Obsérvese que un valor de píxel es el valor de una unidad de píxel que forma una instantánea y es un valor de luminancia o un valor de crominancia, por ejemplo. Por ejemplo, los modos de predicción direccionales son modos de intra predicción que excluyen el modo de predicción DC y el modo de predicción plana.

Los modos de predicción direccionales en direcciones diagonales son modos de predicción direccionales, teniendo cada uno una dirección inclinada con respecto a las direcciones horizontales y verticales. Por ejemplo, los modos de predicción direccionales en direcciones diagonales pueden ser, entre modos de predicción direccionales en 65 direcciones identificadas por los números 2 a 66 en orden comenzando desde la izquierda inferior a la derecha superior (véase la Figura 5A), modos de predicción direccionales en 3 direcciones identificadas por 2 (inferior izquierda), 34 (superior izquierda) y 66 (superior derecha). En otro ejemplo, los modos de predicción direccionales en direcciones diagonales pueden ser modos de predicción direccionales en 7 direcciones identificadas por 2 a 3 (inferior izquierda), 33 a 35 (superior izquierda) y 65 a 66 (superior derecha) entre los modos de predicción direccionales en las 65 direcciones.

Cuando el modo de intra predicción no es el modo predeterminado (No en S301), el transformador 106 determina si la primera base de transformada seleccionada en la etapa S101 es la misma que una base de transformada predeterminada (S103).

Cuando el modo de intra predicción es el modo predeterminado (Sí en S301) o cuando la primera base de transformada es la misma que la base de transformada predeterminada (Sí en S103), el transformador 106 selecciona una segunda base de transformada para el bloque actual de entre uno o más segundos candidatos de base de transformada (S104). El transformador 106 realiza la segunda transformada en los primeros coeficientes de transformada usando la segunda base de transformada seleccionada, para generar segundos coeficientes de transformada (S105). El cuantificador 108 cuantifica los segundos coeficientes de transformada generados (S302) y finaliza el procesamiento de transformación y cuantificación.

Cuando el modo de intra predicción es diferente del modo predeterminado (No en S301) y la primera base de transformada es diferente de la base de transformada predeterminada (No en S103), el transformador 106 omite la selección de una segunda base de transformada (S104) y la segunda transformada (S105). En otras palabras, el transformador 106 no realiza la segunda transformada. En un caso de este tipo, se cuantifican los primeros coeficientes de transformada generados en la etapa S102 (S302) y finaliza el procesamiento de transformación y cuantificación.

[Procesamiento del cuantificador inverso y del transformador inverso del decodificador]

A continuación, se describirán específicamente el procesamiento del cuantificador 204 inverso y del transformador 206 inverso del decodificador 200 de acuerdo con la presente realización con referencia a la Figura 16. La Figura 16 es un diagrama de flujo que ilustra un procesamiento de cuantificación inversa y transformación inversa realizado por el decodificador 200 de acuerdo con la realización 4.

En primer lugar, el cuantificador 204 inverso cuantifica inversamente coeficientes cuantificados de un bloque actual que hay que decodificar (S601). El transformador 206 inverso determina cuál de intra predicción e inter predicción tiene que usarse para el bloque actual (S701).

Cuando se determina usar inter predicción para el bloque actual (inter en S701), el transformador 206 inverso realiza la etapa S702 y la etapa S703 de la misma manera que en la realización 3 y finaliza el procesamiento de cuantificación inversa y transformación inversa.

Por otra parte, cuando se determina usar intra predicción para el bloque actual (intra en S701), el transformador 206 inverso determina si el modo de intra predicción para el bloque actual es un modo predeterminado (S801). El modo predeterminado usado en el decodificador 200 es el mismo que el modo predeterminado usado en el codificador 100.

Cuando el modo de intra predicción no es el modo predeterminado (No en S801), el transformador 206 inverso determina si una primera base de transformada inversa para el bloque actual es la misma que una base de transformada inversa predeterminada (S602).

Cuando el modo de intra predicción es el modo predeterminado (Sí en S801) o cuando la primera base de transformada inversa es la misma que la base de transformada inversa predeterminada (Sí en S602), las etapas S603 a S606 se realizan de la misma manera que en la realización 2 y el procesamiento de transformada inversa y cuantificación inversa finaliza.

Por otra parte, cuando el modo de intra predicción es diferente del modo predeterminado (No en S801) y la primera base de transformada inversa es diferente de la base de transformada inversa predeterminada (No en S602), el transformador 206 inverso omite la selección de una segunda base de transformada inversa (S603) y la segunda transformada inversa (S604). En otras palabras, el transformador 206 inverso no realiza la segunda transformada inversa y selecciona una primera base de transformada inversa (S605). Usando la primera base de transformada inversa seleccionada, el transformador 206 inverso realiza la primera transformada inversa en los coeficientes cuantificados inversamente en S501 (S606) y finaliza el procesamiento de cuantificación inversa y transformación inversa.

[Efectos ventajosos, etc.]

Como se describe anteriormente, el codificador 100 y el decodificador 200 de acuerdo con la presente realización pueden omitir la segunda transformada de acuerdo con el modo de intra predicción y la primera base de transformada. Esto resulta en la reducción en el procesamiento de búsqueda de una combinación óptima de una base de transformada y un parámetro de transformada tanto en la primera transformada como en la segunda transformada, habilitando por lo tanto la reducción en la carga de procesamiento mientras inhibe un descenso en la eficiencia de compresión.

En particular, cuando un modo de predicción direccional en una dirección diagonal es el modo predeterminado, se realiza la segunda transformada si el modo de predicción direccional en una dirección diagonal se usa para el bloque actual, y si no, puede omitirse la segunda transformada. Esto habilita la reducción en la carga de procesamiento mientras inhibe un descenso en la eficiencia de compresión.

En la primera transformada, se realiza generalmente DCT o DST que es separable en la dirección vertical y la dirección horizontal. En un caso de este tipo, la primera transformada no usa la correlación en direcciones diagonales. Por lo tanto, la primera transformada no es suficiente para agregar de forma suficiente coeficientes cuando se usa un modo de predicción direccional en una dirección diagonal que tiene una alta correlación en una dirección diagonal. En vista de esto, cuando se usa un modo de predicción direccional en una dirección diagonal para intra predicción, se realiza la segunda transformada usando una segunda base de transformada que usa la correlación en una dirección diagonal. De este modo, es posible agregar adicionalmente los coeficientes y mejorar la eficiencia de compresión.

Obsérvese que los órdenes de procesamiento de las etapas en los diagramas de flujo en la Figura 15 y la Figura 16 no se limitan a los ilustrados en la Figura 15 y la Figura 16. Por ejemplo, en la Figura 15, la determinación en cuanto a si el modo de intra predicción es el modo predeterminado (S801) y la determinación en cuanto a si la primera base de transformada es la misma que la base de transformada predeterminada (S602) puede realizarse en orden inverso o puede realizarse simultáneamente.

Obsérvese que este aspecto puede implementarse en combinación con uno o más de los otros aspectos de

acuerdo con la presente divulgación. Además, parte de los procedimientos en los diagramas de flujo, parte de los elementos constituyentes de los aparatos y parte de la sintaxis descrita en este aspecto pueden implementarse en combinación con otros aspectos.

REALIZACIÓN 5

- 5 A continuación, se describirá la realización 5. En un aspecto de la presente realización, se describirá la codificación/decodificación de información con respecto una transformada/transformada inversa. A continuación en el presente documento, la presente realización se describirá con referencia a los dibujos, centrándose en las diferencias de las realizaciones 2 a 4. Obsérvese que dado que el procesamiento de transformación y cuantificación y el procesamiento de cuantificación inversa y transformación inversa de
10 acuerdo con la presente realización son sustancialmente los mismos que los de la realización 4, se omitirán las descripciones de los mismos.

[Procesamiento del codificador por entropía del codificador]

- 15 Con referencia a la Figura 17, lo siguiente describe específicamente el procesamiento de codificación para información con respecto a una transformada realizada por el codificador 110 por entropía del codificador 100 de acuerdo con la presente realización. La Figura 17 es un diagrama de flujo que ilustra un procesamiento de codificación realizado por el codificador 100 de acuerdo con la realización 5.

- 20 Cuando se usa inter predicción para el bloque actual (inter en S401), el codificador 110 por entropía codifica una primera señal de selección de base en un flujo de bits (S402). En este punto, la primera señal de selección de base es información o datos que indican la primera base de transformada seleccionada en la etapa S202 en la Figura 15.

Codificar una señal en un flujo de bits es situar un código que indica información en un flujo de bits. El código se genera mediante codificación aritmética binaria adaptativa según contexto (CABAC), por ejemplo. Obsérvese que el código no siempre necesita generarse usando CABAC o codificación por entropía. Por ejemplo, el código puede ser la propia información (una bandera de 0 o 1, por ejemplo).

- 25 A continuación, el codificador 110 por entropía codifica coeficientes cuantificados en la etapa S302 en la Figura 15 (S403) y finaliza el procesamiento de codificación.

- 30 Cuando se usa intra predicción para el bloque actual (intra en S401), el codificador 110 por entropía codifica, en un flujo de bits, una señal de modo de intra predicción que indica el modo de intra predicción para el bloque actual (S404). El codificador 110 por entropía codifica adicionalmente una primera señal de selección de base en el flujo de bits (S405). En este punto, la primera señal de selección de base es información o datos que indican la primera base de transformada seleccionada en la etapa S101 ilustrada en la Figura 15.

- 35 En este punto, cuando se ha realizado la segunda transformada (Sí en S406), el codificador 110 por entropía codifica una segunda señal de selección de base en el flujo de bits (S407). En este punto, la segunda señal de selección de base es información o datos que indican la segunda base de transformada seleccionada en la etapa S104. Por otra parte, cuando no se realiza la segunda transformada (No en S406), el codificador 110 por entropía omite la codificación de la segunda señal de selección de base (S407). Es decir, el codificador 110 por entropía no codifica la segunda señal de selección de base.

Finalmente, el codificador 110 por entropía codifica coeficientes cuantificados en la etapa S302 (S408) y finaliza el procesamiento de codificación.

- 40 [Sintaxis]

La Figura 18 ilustra un ejemplo específico de sintaxis de acuerdo con la realización 5.

- 45 En la Figura 18, se codifican en un flujo de bits una señal de modo de predicción (pred_mode), una señal de modo de intra predicción (pred_mode_dir) y una señal de modo de selección adaptativo (emt_mode) y, según sea necesario, una primera señal de selección de base (primary_transform_type) y una segunda señal de selección de base (secondary_transform_type).

La señal de modo de predicción (pred_mode) indica cuál de intra predicción e inter predicción tiene que usarse para un bloque actual que hay que codificar/decodificar (en este punto, una unidad de codificación). A base de la señal de modo de predicción, el transformador 206 inverso del decodificador 200 puede determinar si usar intra predicción para el bloque actual.

- 50 La señal de modo de intra predicción (pred_mode_dir) indica un modo de intra predicción para cuando tiene que usarse intra predicción para un bloque actual que hay que codificar/decodificar. A base de la señal de modo de intra predicción, el transformador 206 inverso del decodificador 200 puede determinar si el modo de intra predicción para el bloque actual es un modo predeterminado.

La señal de modo de selección adaptativo (emt_mode) indica si usar, para un bloque actual que hay que codificar/decodificar, un modo de selección de base adaptativo en el que se selecciona de forma adaptativa una base de transformada de entre una pluralidad de candidatos de base de transformada. En este punto, cuando la señal de modo de selección adaptativo está "ACTIVADA", se selecciona una base de transformada de entre la DCT de tipo V, la DCT de tipo VIII, la DST de tipo I y la DST de tipo VII. Por otra parte, cuando la señal de modo de selección adaptativo está "DESACTIVADA", se selecciona la DCT de tipo II. A base de la señal de modo de selección adaptativo, el transformador 206 inverso del decodificador 200 puede determinar si la primera base de transformada inversa del bloque actual es la misma que una base de transformada inversa predeterminada.

- 5 La primera señal de selección de base (primary_transform_type) indica una primera base de transformada/base de transformada inversa usada para una transformada/transformada inversa de un bloque actual que hay que codificar/decodificar. La primera señal de selección de base se codifica en un flujo de bits cuando la señal de modo de selección adaptativo está "ACTIVADA". Por otra parte, cuando la señal de modo de selección adaptativo está "DESACTIVADA", la primera señal de selección de base no se codifica. El transformador 206 inverso del decodificador 200 puede seleccionar una primera base de transformada inversa a base de la primera señal de selección de base.
- 10 15

- 20 La segunda señal de selección de base (secondary_transform_type) indica una segunda base de transformada/base de transformada inversa usada para una transformada/transformada inversa de un bloque actual que hay que codificar/decodificar. La segunda señal de selección de base se codifica en un flujo de bits cuando la señal de modo de selección adaptativo está "ACTIVADA" y la señal de modo de intra predicción es "2", "34" o "66". Cada una de las señales de modo de intra predicción "2", "34" y "66" indica un modo de predicción direccional en una dirección diagonal. Es decir, la segunda señal de selección de base se codifica en un flujo de bits cuando la primera base de transformada es la misma que la base de DCT de tipo II y el modo de intra predicción es un modo de predicción direccional en una dirección diagonal. Por otra parte, la segunda señal de selección de base no se codifica en un flujo de bits cuando el modo de intra predicción no es un modo de predicción direccional en una dirección diagonal. El transformador 206 inverso del decodificador 200 puede seleccionar una segunda base de transformada inversa a base de la segunda señal de selección de base.
- 25

- 30 Obsérvese que en este punto, las bases de la DCT de tipo V, la DCT de tipo VIII, la DST de tipo I y la DST de tipo VII se usan como bases de transformada seleccionables en el modo de selección de base adaptativo; sin embargo, la presente divulgación no se limita a estas. Por ejemplo, puede usarse la DCT de tipo IV en lugar de la DCT de tipo V. Dado que la DCT de tipo IV puede usar parcialmente el procesamiento de la DCT de tipo II, puede reducirse la carga de procesamiento. Adicionalmente, puede usarse la DST de tipo IV. Dado que la DST de tipo IV puede usar parcialmente el procesamiento de la DCT de tipo IV, puede reducirse la carga de procesamiento.

- 35 [Procesamiento del decodificador por entropía del decodificador]

A continuación, el procesamiento del decodificador 202 por entropía del decodificador 200 de acuerdo con la presente realización se describirá específicamente con referencia a la Figura 19. La Figura 19 es un diagrama de flujo que ilustra un procesamiento de decodificación realizado por el decodificador 200 de acuerdo con la realización 5.

- 40 Cuando tiene que usarse inter predicción para un bloque actual que hay que decodificar (inter en S901), el decodificador 202 por entropía decodifica una primera señal de selección de base de un flujo de bits (S902).

Decodificar una señal de un flujo de bits es analizar un código que indica información de un flujo de bits, y reconstruir la información a partir del código analizado. La reconstrucción del código a la información se realiza usando decodificación aritmética binaria adaptativa basada en contexto (CABAD), por ejemplo. Obsérvese que la reconstrucción del código a la información no siempre necesita realizarse usando CABAD o decodificación por entropía. Por ejemplo, el análisis del mero código es suficiente cuando el propio código analizado indica la información (una bandera de 0 o 1, por ejemplo).

A continuación, el decodificador 202 por entropía decodifica los coeficientes cuantificados del flujo de bits (S903) y finaliza el procesamiento de decodificación.

- 50 Cuando tiene que usarse intra predicción para el bloque actual (intra en S901), el decodificador 202 por entropía decodifica una señal de modo de intra predicción de un flujo de bits (S904). El decodificador 202 por entropía decodifica adicionalmente una primera señal de selección de base del flujo de bits (S905).

En este punto, cuando tiene que realizarse la segunda transformada inversa (Sí en S906), el decodificador 202 por entropía decodifica una segunda señal de selección de base del flujo de bits (S907). Por otra parte, cuando la segunda transformada inversa no tiene que realizarse (No en S906), el decodificador 202 por entropía omite la decodificación de la segunda señal de selección de base (S907). Es decir, el decodificador 202 por entropía no decodifica la segunda señal de selección de base.

Finalmente, el decodificador 202 por entropía decodifica los coeficientes cuantificados del flujo de bits (S908) y finaliza el procesamiento de decodificación.

[Efectos ventajosos, etc.]

- Como se describe anteriormente, el codificador 100 y el decodificador 200 de acuerdo con la presente realización pueden codificar la primera señal de selección de base y la segunda señal de selección de base en un flujo de bits. Codificando la señal de modo de intra predicción y la primera señal de selección de base antes de la segunda señal de selección de base, es posible determinar, antes de la decodificación de la segunda señal de selección de base, si omitir la segunda transformada inversa. Por consiguiente, en el caso de omitir la segunda transformada inversa, también es posible omitir la codificación de la segunda señal de selección de base y, por lo tanto, puede mejorarse la eficiencia de compresión.

REALIZACIÓN 6

- A continuación, se describirá la realización 6. Un aspecto de la presente realización es diferente de la realización 5 en que se codifica la información que indica un modo de intra predicción en el que se realiza la segunda transformada. A continuación en el presente documento, la presente realización se describirá con referencia a los dibujos, centrándose en las diferencias de la realización 5. Obsérvese que a lo largo de los dibujos mencionados anteriormente, se proporcionan los mismos números de referencia a las etapas de procesamiento que son sustancialmente las mismas que la de la realización 5 y se omitirán o simplificarán las descripciones solapantes.

[Procesamiento del codificador por entropía del codificador]

- Con referencia a la Figura 20, lo siguiente describe específicamente el procesamiento de codificación para información con respecto a una transformada realizada por el codificador 110 por entropía del codificador 100 de acuerdo con la presente realización. La Figura 20 es un diagrama de flujo que ilustra un procesamiento de codificación realizado por el codificador 100 de acuerdo con la realización 6.

- Cuando se usa inter predicción para un bloque actual que hay que codificar (inter en S401), el codificador 110 por entropía realiza la etapa S402 y la etapa S403 de la misma manera que en la realización 5 y finaliza el procesamiento de codificación.

- Por otra parte, cuando se usa intra predicción para el bloque actual (intra en S401), el codificador 110 por entropía codifica una segunda señal de modo de predicción objetivo de transformada en un flujo de bits (S501). La segunda señal de modo de predicción objetivo de transformada indica un modo predeterminado para determinar si realizar la segunda transformada inversa. Específicamente, la segunda señal de modo de predicción objetivo de transformada indica un número de modo de intra predicción (2, 34 o 66, por ejemplo), por ejemplo.

- Obsérvese que la unidad de codificación para la segunda señal de modo de predicción objetivo de transformada puede ser una unidad de codificación (CU) o una unidad de árbol de codificación (CTU), o puede ser un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de instantánea (PPS) o una unidad de corte que corresponde a la norma H.265/HEVC.

Después de eso, el codificador 110 por entropía realiza las etapas S404 a S408 de la misma manera que en la realización 5 y finaliza el procesamiento de codificación.

[Procesamiento del decodificador por entropía del decodificador]

- A continuación, el procesamiento del decodificador 202 por entropía del decodificador 200 de acuerdo con la presente realización se describirá específicamente con referencia a la Figura 21. La Figura 21 es un diagrama de flujo que ilustra un procesamiento de decodificación realizado por el decodificador 200 de acuerdo con la realización 6.

- Cuando tiene que usarse inter predicción para un bloque actual que hay que decodificar (inter en S901), el decodificador 202 por entropía realiza la etapa S902 y la etapa S903 de la misma manera que en la realización 5 y finaliza el procesamiento de decodificación.

Por otra parte, cuando tiene que usarse intra predicción para el bloque actual (intra en S901), el decodificador 202 por entropía decodifica una segunda señal de modo de predicción objetivo de transformada de un flujo de bits (S1001).

- Después de eso, el codificador 202 por entropía realiza las etapas S904 a S908 de la misma manera que en la realización 5 y finaliza el procesamiento de decodificación.

[Efectos ventajosos, etc.]

Como se describe anteriormente, el codificador 100 y el decodificador 200 de acuerdo con la presente realización pueden codificar, en un flujo de bits, la segunda señal de modo de predicción objetivo de transformada que indica un modo predeterminado que es el modo de intra predicción en el que se realiza la segunda transformada/transformada inversa. Por consiguiente, el modo predeterminado puede determinarse libremente en el lado del codificador 100 y puede mejorarse la eficiencia de compresión.

Obsérvese que el orden en el que se codifican las señales puede determinarse por adelantado, y pueden codificarse diversas señales en un orden diferente del orden de codificación anteriormente mencionado.

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

REALIZACIÓN 7

Pueden hacerse diversas modificaciones a las realizaciones 2 a 6.

Por ejemplo, en cada una de las realizaciones anteriores, la primera base de transformada puede fijarse de acuerdo con el tamaño del bloque actual que hay que codificar/decodificar. Por ejemplo, cuando el tamaño de bloque es menor que un cierto tamaño (por ejemplo, píxeles de 4×4 , píxeles de 4×8 o píxeles de 8×4), la primera base de transformada puede fijarse a una base de transformada de DST de tipo VII y, en este momento, puede omitirse la codificación de la primera señal de selección de base.

Adicionalmente, por ejemplo, en cada una de las realizaciones anteriores, puede codificarse una señal que indica si habilitar el procesamiento de omisión de la selección de la primera base de transformada y la primera transformada o la selección de la segunda base de transformada y la segunda transformada. Por ejemplo, cuando se habilita el procesamiento de omisión de la segunda transformada, la segunda señal de selección de base puede no codificarse y, por lo tanto, la operación de decodificación se vuelve diferente de la operación de decodificación realizada cuando se deshabilita el procesamiento de omisión de la segunda transformada. La unidad de codificación para una señal de este tipo puede ser una unidad de codificación (CU) o una unidad de árbol de codificación (CTU), o puede ser un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de instantánea (PPS) o una unidad de corte que corresponde a la norma H.265/HEVC.

Por ejemplo, en cada una de las realizaciones anteriores, puede omitirse la selección de la primera base de transformada y la primera transformada o puede omitirse la selección de la segunda base de transformada y la segunda transformada, a base del tipo de instantánea (I, P, B), el tipo de corte (I, P, B), el tamaño de bloque, el número de coeficientes distintos de cero, un parámetro de cuantificación o Temporal_id (capa de codificación jerárquica).

Obsérvese que cuando el codificador realiza tales operaciones como se describe anteriormente, el decodificador también realiza correspondientes operaciones. Por ejemplo, cuando se codifica la información que indica si habilitar el procesamiento de omisión de la primera transformada o segunda transformada, el decodificador decodifica esa información para determinar si la primera o segunda transformada y si se codifica la primera o segunda señal de selección de base.

Obsérvese que en las realizaciones 5 y 6, una pluralidad de señales (por ejemplo, la señal de modo de intra predicción, la señal de modo de selección adaptativo, la primera señal de selección de base y la segunda señal de selección de base) se codifican en un flujo de bits; sin embargo, en las realizaciones 2 a 4, estas señales no necesitan codificarse en un flujo de bits. Por ejemplo, estas señales pueden notificarse desde el codificador 100 al decodificador 200 de forma separada del flujo de bits.

Obsérvese que en la presente realización, las respectivas posiciones, en el flujo de bits, de la pluralidad de señales (por ejemplo, la señal de modo de intra predicción, la señal de modo de selección adaptativo, la primera señal de selección de base y la segunda señal de selección de base) no se limitan particularmente. La pluralidad de señales se codifican en al menos una de una pluralidad de encabezamientos, por ejemplo. Por ejemplo, un conjunto de parámetros de vídeo, un conjunto de parámetros de secuencia, un conjunto de parámetros de instantánea y un encabezamiento de corte pueden usarse como la pluralidad de encabezamientos. Obsérvese que cuando una señal se ubica en dos o más capas (por ejemplo, un conjunto de parámetros de instantánea y un encabezamiento de corte), la señal en una capa inferior (por ejemplo, el encabezamiento de corte) sobrescribe la señal en una capa superior (por ejemplo, el conjunto de parámetros de instantánea).

REALIZACIÓN 8

Como se describe en cada una de las realizaciones anteriores y variaciones, cada bloque funcional puede realizarse habitualmente como una MPU y memoria, por ejemplo. Además, los procedimientos realizados por cada uno de los bloques funcionales se realizan habitualmente por una unidad de ejecución de programa, tal como un procesador, software de lectura y ejecución (un programa) grabado en un medio de grabación, tal

como una ROM. El software puede distribuirse a través de, por ejemplo, descarga y puede grabarse en un medio de grabación, tal como memoria de semiconductores y distribuida. Obsérvese que cada bloque funcional también puede realizarse, por supuesto, como hardware (circuito especializado).

5 Además, el procesamiento descrito en cada una de las realizaciones y variaciones puede realizarse a través de procesamiento integrado usando un único aparato (sistema) y, como alternativa, puede realizarse a través de procesamiento descentralizado usando una pluralidad de aparatos. Además, el procesador que ejecuta el programa anteriormente descrito puede ser un único procesador o una pluralidad de procesadores. En otras palabras, puede realizarse el procesamiento integrado y, como alternativa, puede realizarse el procesamiento descentralizado.

10 A continuación, se describirán ejemplos de aplicación del procedimiento de codificación de instantáneas en movimiento (procedimiento de codificación de imágenes) y el procedimiento de decodificación de instantáneas en movimiento (procedimiento de decodificación de imágenes) descritos en cada una de las realizaciones anteriores y variaciones y un sistema que emplea los mismos. El sistema se caracteriza como que incluye un codificador de imagen que emplea el procedimiento de codificación de imágenes, un decodificador de imagen que emplea el procedimiento de decodificación de imágenes y un codificador de imagen/decodificador que incluye tanto el codificador de imagen como el decodificador de imagen. Puede modificarse otras configuraciones incluidas en el sistema sobre una base de caso a caso.

[Ejemplos de uso]

20 La Figura 22 ilustra una configuración global de sistema ex100 de suministro de contenido de implementación de un servicio de distribución de contenido. El área en la que se proporciona el servicio de comunicación se divide en células de tamaños deseados y las estaciones ex106, ex107, ex108, ex109 y ex110 base, que son estaciones inalámbricas fijas, se ubican en respectivas células.

25 En el sistema ex100 de suministro de contenido, dispositivos que incluyen el ordenador ex111, el dispositivo ex112 de juegos, la cámara ex113, el electrodoméstico ex114 y el teléfono ex115 inteligente se conectan a la internet ex101 a través del proveedor ex102 de servicio de internet o la red ex104 de comunicaciones y las estaciones ex106 a ex110 base. El sistema ex100 de suministro de contenido puede combinar y conectar cualquier combinación de los elementos anteriores. Los dispositivos pueden conectarse juntos directa o indirectamente a través de una red de telefonía o comunicación de campo cercano en lugar de a través de las estaciones ex106 a ex110 base, que son estaciones inalámbricas fijas. Además, el servidor ex103 de difusión en continuo se conecta a dispositivos que incluyen el ordenador ex111, el dispositivo ex112 de juegos, la cámara ex113, el electrodoméstico ex114 y el teléfono ex115 inteligente a través de, por ejemplo, la internet ex101. El servidor ex103 de difusión en continuo también se conecta a, por ejemplo, un terminal en una zona de acceso en el avión ex117 a través del satélite ex116.

30 35 Obsérvese que en lugar de las estaciones ex106 a ex110 base, pueden usarse puntos de acceso o zonas de acceso inalámbrico. El servidor ex103 de difusión en continuo puede conectarse a la red ex104 de comunicaciones directamente en lugar de a través de la internet ex101 o el proveedor ex102 de servicio de internet, y puede conectarse al avión ex117 directamente en lugar de a través del satélite ex116.

40 La cámara ex113 es un dispositivo con capacidad de capturar imágenes fijas y vídeo, tal como una cámara digital. El teléfono ex115 inteligente es un dispositivo de teléfono inteligente, teléfono celular o teléfono de sistema de teléfonos personales (PHS) que puede operar bajo las normas de sistema de comunicaciones móviles de las típicas 2G, 3G, 3.9G y sistemas 4G, así como el sistema 5G de próxima generación.

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

45 En el sistema ex100 de suministro de contenido, un terminal que incluye una función de captura de imágenes y/o vídeo es capaz de, por ejemplo, difusión en continuo conectando al servidor ex103 de difusión en continuo a través de, por ejemplo, la estación ex106 base. Cuando se difunde en continuo en directo, un terminal (por ejemplo, el ordenador ex111, el dispositivo ex112 de juegos, la cámara ex113, el electrodoméstico ex114, el teléfono ex115 inteligente o el avión ex117) realiza el procesamiento de codificación descrito en las realizaciones anteriores y variaciones en contenido de imágenes fijas o vídeo capturado por un usuario a través del terminal, multiplexa datos de vídeo obtenidos a través de la codificación y datos de audio obtenidos codificando audio que corresponde al vídeo, y transmite los datos obtenidos al servidor ex103 de difusión en continuo. En otras palabras, el terminal funciona como el codificador de imagen de acuerdo con un aspecto de la presente divulgación.

50 55 El servidor ex103 de difusión en continuo difunde en continuo los datos de contenido transmitidos a los clientes que solicitan el flujo. Ejemplos de cliente incluyen el ordenador ex111, el dispositivo ex112 de juegos, la cámara ex113, el electrodoméstico ex114, el teléfono ex115 inteligente y terminales dentro del avión ex117, que tienen capacidad de decodificar los datos codificados descritos anteriormente. Los dispositivos que reciben los datos difundidos en continuo decodifican y reproducen los datos recibidos. En otras palabras, cada uno de los

dispositivos funciona como el decodificador de imagen de acuerdo con un aspecto de la presente divulgación.

[Procesamiento descentralizado]

El servidor ex103 de difusión en continuo puede realizarse como una pluralidad de servidores u ordenadores entre los que se dividen las tareas, tales como el procesamiento, grabación y difusión en continuo de datos.

5 Por ejemplo, el servidor ex103 de difusión en continuo puede realizarse como una red de distribución de contenidos (CDN) que difunde en continuo contenido a través de una red que conecta múltiples servidores de borde ubicados a lo largo de todo el mundo. En una CDN, un servidor de borde físicamente cerca del cliente se asigna dinámicamente al cliente. El contenido se almacena en caché y difunde en continuo al servidor de borde para reducir los tiempos de carga. En el caso de, por ejemplo, alguna clase de un error o un cambio en
10 conectividad debido a, por ejemplo, un pico en tráfico, es posible difundir en continuo datos de forma estable a velocidades altas dado que es posible evitar partes afectadas de la red dividiendo, por ejemplo, el procesamiento entre una pluralidad de servidores de borde o comutando las tareas de difusión en continuo a un servidor de borde diferente y continuando la difusión en continuo.

15 La descentralización no se limita solo a la división de procesamiento de difusión en continuo; la codificación de los datos capturados puede dividirse entre y realizarse por los terminales, en el lado de servidor, o ambos. En un ejemplo, en una codificación típica, el procesamiento se realiza en dos bucles. El primer bucle es para detectar cómo de complicada es la imagen en una base de fotograma a fotograma o escena a escena, o detectar la carga de codificación. El segundo bucle es para un procesamiento que mantiene la calidad de imagen y mejora la eficiencia de codificación. Por ejemplo, es posible reducir la carga de procesamiento de los
20 terminales y mejorar la calidad y eficiencia de codificación del contenido haciendo que los terminales realicen el primer bucle de la codificación y haciendo que el lado de servidor que recibió el contenido realice el segundo bucle de la codificación. En un caso de este tipo, tras la recepción de una petición de decodificación, es posible que los datos codificados que resultan del primer bucle realizado por un terminal se reciban y reproduzcan en otro terminal en aproximadamente tiempo real. Esto hace posible realizar una difusión en continuo suave y en
25 tiempo real.

En otro ejemplo, la cámara ex113 o similar extrae una cantidad de características de una imagen, comprime datos relacionados con la cantidad de características como metadatos, y transmite los metadatos comprimidos a un servidor. Por ejemplo, el servidor determina la significancia de un objeto a base de la cantidad de características y cambia la precisión de cuantificación, por consiguiente, para realizar compresión adecuada para el significado de la imagen. Los datos de cantidad de características es particularmente efectivo en la mejora de la precisión y eficiencia de predicción de 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 codificación de longitud variable (VLC), puede tratarse por el terminal, y la codificación que tiene una carga de procesamiento relativamente alta, tal como codificación aritmética binaria adaptativa según contexto (CABAC), puede tratarse por el servidor.

30 En otro ejemplo más, existen casos en los que una pluralidad de vídeos de aproximadamente la misma escena se capturan por una pluralidad de terminales en, por ejemplo, un estadio, un centro comercial o una fábrica. En un caso de este tipo, por ejemplo, la codificación puede descentralizarse dividiendo tareas de procesamiento entre la pluralidad de terminales que capturaron los vídeos y, si fuera necesario, otros terminales que no
35 capturaron los vídeos y el servidor, sobre una base por unidad. Las unidades pueden ser, por ejemplo, grupos de instantáneas (GOP), instantáneas o losas que resultan de la división de una instantánea. Esto hace posible reducir tiempos de carga y conseguir una difusión en continuo que está más cercana al tiempo real.

40 Además, dado que los vídeos son de aproximadamente la misma escena, la gestión y/o la instrucción puede efectuarse por el servidor de modo que los vídeos capturados por los terminales pueden hacerse referencia de forma cruzada. Además, el servidor puede recibir datos codificados desde los terminales, cambiar la relación de referencia entre artículos de datos o corregir o sustituir las propias instantáneas y, a continuación, realizar la codificación. Esto hace posible generar una difusión en continuo con calidad y eficiencia aumentadas para los artículos individuales de datos.

45 Además, el servidor puede difundir en continuo datos de vídeo después de realizar una transcodificación para convertir el formato de codificación de los datos de vídeo. Por ejemplo, el servidor puede convertir el formato de codificación desde MPEG a VP y puede convertir H.264 a H.265.

50 De esta manera, la codificación puede realizarse por un terminal o uno o más servidores. Por consiguiente, aunque el dispositivo que realiza la codificación se denomina un "servidor" o "terminal" en la siguiente descripción, algunos o todos de los procedimientos realizados por el servidor pueden realizarse por el terminal y, asimismo, algunos o todos de los procedimientos realizados por el terminal pueden realizarse por el servidor. Esto también se aplica a procedimientos de decodificación.

[3D, múltiples ángulos]

En años recientes, ha aumentado el uso de imágenes o vídeos combinados a partir de imágenes o vídeos de

diferentes escenas capturadas simultáneamente o la misma escena capturada desde ángulos diferentes por una pluralidad de terminales, tales como la cámara ex113 y/o el teléfono ex115 inteligente. Los vídeos capturados por los terminales se combinan a base de, por ejemplo, la relación posicional relativa obtenida de forma separada entre los terminales, o regiones en un vídeo que tiene puntos de característica concordantes.

5 Además de la codificación de instantáneas en movimiento bidimensionales, el servidor puede codificar una imagen fija a base de análisis de escena de una instantánea en movimiento o bien automáticamente o bien en un punto en tiempo especificado por el usuario, y transmitir la imagen fija codificada a un terminal de recepción. Adicionalmente, cuando el servidor puede obtener la relación posicional relativa entre los terminales de captura de vídeo, además de instantáneas en movimiento bidimensionales, el servidor puede generar geometría

10 tridimensional de una escena a base de vídeo de la misma escena capturada desde ángulos diferentes. Obsérvese que el servidor puede codificar de forma separada datos tridimensionales generados desde, por ejemplo, una nube de punto y puede seleccionar o reconstruir y generar, a base de un resultado de reconocimiento o seguimiento de una persona u objeto usando datos tridimensionales, un vídeo que hay que transmitir a un terminal de recepción desde vídeos capturados por una pluralidad de terminales.

15 Esto permite que el usuario disfrute una escena seleccionando libremente vídeos que corresponden a los terminales de captura de vídeo y permite que el usuario disfrute del contenido obtenido extrayendo, de los datos tridimensionales reconstruidos a partir de una pluralidad de imágenes o vídeos, un vídeo desde un punto de vista seleccionado. Adicionalmente, similar a con el vídeo, el sonido puede grabarse desde ángulos relativamente diferentes, y el servidor puede multiplexar, con el vídeo, audio desde un ángulo específico o 20 espacio de acuerdo con el vídeo, y transmitir el resultado.

En años recientes, también se ha vuelto popular el contenido que es un compuesto del mundo real y un mundo virtual, tal como contenido de realidad virtual (RV) y realidad aumentada (RA). En el caso de imágenes de RV, el servidor puede crear imágenes desde los puntos de vista de ambos ojos izquierdo y derecho y realizar codificación que tolera la referencia entre las dos imágenes de punto de vista, tal como codificación de múltiples 25 vistas (MVC) y, como alternativa, puede codificar las imágenes como flujos separados sin referencia. Cuando las imágenes se decodifican como flujos separados, los flujos pueden sincronizarse cuando se reproducen para recrear un espacio tridimensional virtual de acuerdo con el punto de vista del usuario.

En el caso de imágenes de RA, el servidor superpone información de objeto virtual existente en un espacio 30 virtual en información de cámara que representa un espacio de mundo real, a base de una posición o movimiento tridimensional desde la perspectiva del usuario. El decodificador puede obtener o almacenar información de objeto virtual y datos tridimensionales, generar imágenes bidimensionales a base de movimiento desde la perspectiva del usuario y, a continuación, generar datos superpuestos conectando sin discontinuidades las imágenes. Como alternativa, el decodificador puede transmitir, al servidor, movimiento 35 desde la perspectiva del usuario además de una petición para información de objeto virtual, y el servidor puede generar datos superpuestos a base de datos tridimensionales almacenados en el servidor de acuerdo con el movimiento recibido, y codificar y difundir en continuo los datos superpuestos generados al decodificador. Obsérvese que datos superpuestos incluyen, además de valores RGB, un valor α que indica transparencia, y el servidor establece el valor α para secciones distintas del objeto generado a partir de datos tridimensionales 40 a, por ejemplo, 0 y puede realizar la codificación mientras esas secciones son transparentes. Como alternativa, el servidor puede establecer el segundo plano a un valor RGB predeterminado, tal como una clave de crominancia, y generar datos en los que áreas distintas del objeto se establecen como el segundo plano.

La decodificación de datos difundidos en continuo de forma similar puede realizarse por el cliente (es decir, los terminales), en el lado de servidor, o dividirse entre los mismos. En un ejemplo, un terminal puede transmitir una petición de recepción a un servidor, el contenido solicitado puede recibirse y decodificarse por otro terminal 45 y puede transmitirse una señal decodificada a un dispositivo que tiene un visualizador. Es posible reproducir datos de calidad de imagen alta descentralizando el procesamiento y seleccionando apropiadamente contenido independientemente de la capacidad de procesamiento del propio terminal de comunicaciones. En otro ejemplo más, mientras una TV, por ejemplo, está recibiendo datos de imagen que son grandes en tamaño, una región de una instantánea, tal como una pieza obtenida dividiendo la instantánea, puede decodificarse y visualizarse 50 en un terminal o terminales personales de un espectador o espectadores de la TV. Esto hace posible que los espectadores compartan una vista de instantánea grande, así como para que cada espectador compruebe su área asignada o inspeccione una región en detalle adicional de cerca.

En el futuro, tanto en interiores como exteriores, en situaciones en las que son posibles una pluralidad de conexiones inalámbricas a través de distancias cercanas, medianas y lejanas, se espera que sea capaz de recibir sin discontinuidades contenido incluso cuando se comután a datos apropiados para la conexión actual, usando una norma de sistema de difusión en continuo, tal como MPEG-DASH. Con esto, el usuario puede comutar entre datos en tiempo real mientras selecciona libremente un decodificador o aparato de visualización que incluye no únicamente su propio terminal, sino también, por ejemplo, visualizadores dispuestos en interiores o exteriores. Además, a base de, por ejemplo, información sobre la posición del usuario, la decodificación puede realizarse mientras se comuta qué terminal trata la decodificación y qué terminal trata la visualización de contenido. Esto hace posible, mientras en ruta a un destino, visualizar, en la pared de un

edificio cercano en el que se embebe un dispositivo con capacidad de visualizar contenido o en parte de la tierra, información de mapa mientras está en movimiento. Además, también es posible comutar la tasa de bits de los datos recibidos a base de la accesibilidad a los datos codificados en una red, tales como cuando los datos codificados se almacenan en caché en un servidor rápidamente accesible desde el terminal de recepción o cuando datos codificados se copian a un servidor de borde en un servicio de distribución de contenidos.

5 [Codificación escalable]

la comutación de contenido se describirá con referencia a un flujo escalable, ilustrado en la Figura 23, que se codifica por compresión a través de la implementación del procedimiento de codificación de instantáneas en movimiento descrito en las realizaciones anteriores y variaciones. El servidor puede tener una configuración en

10 la que el contenido se comuta mientras hace uso de la escalabilidad temporal y/o espacial de un flujo, que se consigue mediante la división en y la codificación de capas, como se ilustra en la Figura 23. Obsérvese que puede haber una pluralidad de flujos individuales que son del mismo contenido, pero de diferente calidad. En otras palabras, determinando hasta qué capa decodificar a base de factores internos, tales como la capacidad de procesamiento en el lado de decodificador, y factores externos, tales como ancho de banda de comunicación, el lado de decodificador puede comutar libremente entre contenido de baja resolución y contenido de alta resolución mientras se decodifica. Por ejemplo, en un caso en el que el usuario quiere continuar viendo, en casa en un dispositivo, tal como una TV conectada a la internet, un vídeo que él o ella ha estado viendo anteriormente en el teléfono ex115 inteligente mientras se está moviendo, el dispositivo puede decodificar simplemente el mismo flujo hasta una capa diferente, lo que reduce la carga de lado de servidor.

15 20 Adicionalmente, además de la configuración descrita anteriormente en la que se consigue escalabilidad como resultado de las instantáneas que se codifican por capa y la capa de mejora está por encima de la capa base, la capa de mejora puede incluir metadatos a base de, por ejemplo, información estadística sobre la imagen, y el lado de decodificador puede generar contenido de calidad de imagen alta realizando formación de imágenes de superresolución en una instantánea en la capa base a base de los metadatos. La formación de imágenes 25 de superresolución puede ser mejorando la relación de SN mientras se mantiene la resolución y/o se aumenta la resolución. Los metadatos incluyen información para identificar un coeficiente lineal y uno no lineal usados en el procesamiento de superresolución o información que identifica un valor de parámetro en procesamiento de filtro, aprendizaje automático o procedimiento de mínimos cuadrados usado en procesamiento de superresolución.

30 35 Como alternativa, también es aceptable una configuración en la que una instantánea se divide en, por ejemplo, losas de acuerdo con el significado de, por ejemplo, un objeto en la imagen y, en el lado de decodificador, únicamente se decodifica una región parcial seleccionando una losa a decodificar. Además, almacenando un atributo acerca del objeto (persona, coche, balón, etc.) y una posición del objeto en el vídeo (coordenadas en imágenes idénticas) como metadatos, el lado de decodificador puede identificar la posición de un objeto deseado a base de los metadatos y determinar que losa o losas incluyen ese objeto. Por ejemplo, como se ilustra en la Figura 24, los metadatos se almacenan usando una estructura de almacenamiento de datos diferente desde datos de píxel, tales como un mensaje de SEI en HEVC. Estos metadatos indican, por ejemplo, la posición, tamaño o color del objeto principal.

40 Además, los metadatos pueden almacenarse en unidades de una pluralidad de instantáneas, tales como flujo, secuencia o unidades de acceso aleatorias. Con esto, el lado de decodificador puede obtener, por ejemplo, el momento en el que una persona específica aparece en el vídeo, y encajando eso con la información de unidad de instantánea, puede identificar una instantánea en la que el objeto está presente y la posición del objeto en la instantánea.

45 [Optimización de página web]

50 55 La Figura 25 ilustra un ejemplo de una pantalla de visualización de una página web en, por ejemplo, el ordenador ex111. La Figura 26 ilustra un ejemplo de una pantalla de visualización de una página web en, por ejemplo, el teléfono ex115 inteligente. Como se ilustra en la Figura 25 y la Figura 26, una página web puede incluir una pluralidad de enlaces de imagen que son enlaces a contenido de imagen y la apariencia de la página web difiere dependiendo del dispositivo usado para visualizar la página web. Cuando se pueden ver una pluralidad de enlaces de imagen 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 todo el enlace de imagen encaja en la pantalla, el aparato de visualización (decodificador) visualiza, como los enlaces de imagen, imágenes fijas incluidas en el contenido o instantáneas de tipo I, visualiza vídeo, tal como un gif animado que usa una pluralidad de imágenes fijas o instantáneas de tipo I, por ejemplo, o recibe únicamente la capa base y decodifica y visualiza el vídeo.

Cuando el usuario selecciona un enlace de imagen, el aparato de visualización decodifica dando la prioridad más alta a la capa base. Obsérvese que si existe información en el código HTML de la página web que indica que el contenido es escalable, el aparato de visualización puede decodificar hasta la capa de mejora. Además, para garantizar la reproducción en tiempo real, antes de que se haga una selección o cuando el ancho de

banda está gravemente limitado, el aparato de visualización puede reducir el retardo entre el punto en el tiempo en el que se decodifica la instantánea principal y el punto en el tiempo en el que se visualiza la instantánea decodificada (es decir, el retardo entre el inicio de la decodificación del contenido hasta la visualización del contenido) decodificando y visualizando únicamente instantáneas de referencia directas (instantáneas de tipo I, instantáneas de tipo P, instantáneas de tipo B de referencia directa). Además, el aparato de visualización puede ignorar deliberadamente la relación de referencia entre instantáneas y decodificar de forma gruesa todas las instantáneas de tipo B y P como instantáneas de referencia directas y, a continuación, realizar decodificación normal a medida que aumenta el número de instantáneas recibidas con el paso del tiempo.

[Conducción autónoma]

10 Cuando se transmite y recibe una imagen fija o datos de vídeo, tal información de mapa bi o tridimensional para la conducción autónoma o conducción asistida de un automóvil, el terminal de recepción puede recibir, además de datos de imagen que pertenecen a una o más capas, información sobre, por ejemplo, el tiempo meteorológico o construcción de carretera como metadatos, y asociar los metadatos con los datos de imagen tras la decodificación. Obsérvese que los metadatos pueden asignarse por capa y, como alternativa, puede multiplexarse simplemente con los datos de imagen.

15 En un caso de este tipo, dado que el automóvil, dron, avión, etc., incluyendo el terminal de recepción es móvil, el terminal de recepción puede ser receptor y decodificar sin discontinuidades mientras cambia entre estaciones base entre las estaciones ex106 a ex110 base transmitiendo información que indica la posición del terminal de recepción tras la petición de recepción. Además, de acuerdo con la selección hecha por el usuario, la situación 20 del usuario, o el ancho de banda de la conexión, el terminal de recepción puede seleccionar dinámicamente hasta qué punto se reciben los metadatos o hasta qué punto se actualiza la información de mapa, por ejemplo.

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

[Difusión en continuo de contenido individual]

25 En el sistema ex100 de suministro de contenido, además de calidad de imagen alta, contenido largo distribuido por una entidad de distribución de vídeo, difusión en continuo de unidifusión o multidifusión de calidad de imagen baja, también es posible contenido corto desde un individuo. Además, tal contenido desde individuos es probable que aumente adicionalmente en popularidad. El servidor puede realizar primero un procesamiento de edición en el contenido antes del procesamiento de codificación para refinar el contenido individual. Esto 30 puede conseguirse con, por ejemplo, la siguiente configuración.

En tiempo real, mientras captura contenido de vídeo o imagen o después de que se haya capturado y acumulado el contenido, el servidor realiza un procesamiento de reconocimiento a base de datos sin procesar o codificados, tal como procesamiento de errores de captura, procesamiento de búsqueda de escena, análisis de significado y/o procesamiento de detección de objeto. A continuación, a base del resultado del procesamiento de reconocimiento, el servidor -o bien cuando se solicita o bien automáticamente- edita el contenido, ejemplos del cual incluyen: corrección tal como enfoque y/o corrección de desenfoque de movimiento; eliminar escenas de baja prioridad, tales como escenas que son bajas en iluminación en comparación con otras instantáneas o fuera de enfoque; ajuste de borde de objeto; y ajuste de tono de color. El servidor codifica los datos editados a base del resultado de la edición. Se conoce que los vídeos 40 excesivamente largos tienden a recibir menos vistas. Por consiguiente, para mantener el contenido dentro de una longitud específica que escala con la longitud del vídeo original, el servidor puede, además de las escenas de baja prioridad descritas anteriormente, cortar automáticamente escenas con movimiento bajo a base de un resultado de procesamiento de imagen. Como alternativa, el servidor puede generar y codificar un resumen de vídeo a base de un resultado de un análisis del significado de una escena.

45 Obsérvese que existen casos en los que el contenido individual puede incluir contenido que infringe un derecho de autor, derecho moral, derechos de retrato, etc. Un caso de este tipo puede conducir a una situación desfavorable para el creador, tal como cuando el contenido se comparte más allá del ámbito previsto por el creador. Por consiguiente, antes de codificar, el servidor puede editar, por ejemplo, imágenes para desenfocar las caras de las personas en la periferia de la pantalla o desenfocar el interior de una casa, por ejemplo. Además, el servidor puede configurarse para reconocer las caras de las personas distintas de una persona registrada en imágenes que hay que codificar y, cuando tales caras aparecen en una imagen, por ejemplo, aplicar un filtro de mosaico a la cara de la persona. Como alternativa, como procesamiento previo o posterior de codificación, el usuario puede especificar, por razones de derechos de autor, una región de una imagen que incluye una persona o una región del segundo plano que hay que procesar, y el servidor puede procesar la 50 región especificada, por ejemplo, sustituyendo la región con una imagen diferente o desenfocando la región. Si la región incluye una persona, la persona puede seguirse en la instantánea en movimiento, y la región de la cabeza puede sustituirse por otra imagen a medida que la persona se mueve.

Además, dado que existe una demanda para visualización de tiempo real de contenido producido por

individuos, que tiende a ser pequeña en tamaño de datos, el decodificador recibe primero la capa base como la prioridad más alta y realiza decodificación y reproducción, aunque esto puede diferir dependiendo del ancho de banda. Cuando el contenido se reproduce dos o más veces, tales como cuando el decodificador recibe la capa de mejora durante la decodificación y la reproducción de la capa base y enlaza en bucle la reproducción, el decodificador puede reproducir un vídeo de calidad de imagen alta que incluye la capa de mejora. Si el flujo se codifica usando tal codificación escalable, el vídeo puede ser de baja calidad cuando está en un estado no seleccionado o en el inicio del vídeo, pero puede ofrecer una experiencia en la que la calidad de imagen del flujo aumenta de forma progresiva de una manera inteligente. Esto no se limita a solo codificación escalable; puede ofrecerse la misma experiencia configurando un único flujo de un flujo de baja calidad reproducido para el primer tiempo y un segundo flujo codificado usando el primer flujo como una referencia.

[Otros ejemplos de uso]

La codificación y decodificación puede realizarse por el LSI ex500, que se incluye habitualmente en cada terminal. El LSI ex500 puede configurarse de un único chip o una pluralidad de chips. El software de codificación y decodificación instantáneas en movimiento puede integrarse en algún tipo de un medio de grabación (tal como un CD-ROM, un disco flexible o un disco duro) que es legible por, por ejemplo, el ordenador ex111, y la codificación y decodificación puede realizarse usando el software. Adicionalmente, cuando el teléfono ex115 inteligente está equipado con una cámara, pueden transmitirse los datos de vídeo obtenidos por la cámara. En este caso, los datos de vídeo se codifican por el LSI ex500 incluido en el teléfono ex115 inteligente.

Obsérvese que el LSI ex500 puede configurarse para descargar y activar una aplicación. En un caso de este tipo, el terminal determina primero si es compatible con el esquema usado 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 software de aplicación y, a continuación, obtiene y reproduce el contenido.

A parte del ejemplo del sistema ex100 de suministro de contenido que usa la internet ex101, al menos el codificador de instantáneas en movimiento (codificador de imágenes) o el decodificador de instantáneas en movimiento (decodificador de imágenes) descrito en las realizaciones anteriores y variaciones puede implementarse en un sistema de difusión digital. El mismo procesamiento de codificación y procesamiento de decodificación pueden aplicarse para transmitir y recibir ondas de radio difundidas superpuestas con datos de audio y vídeo multiplexados usando, por ejemplo, un satélite, aunque esto se adapta hacia multidifusión mientras que unidifusión es más fácil con el sistema ex100 de suministro de contenido.

[Configuración de hardware]

La Figura 27 ilustra el teléfono ex115 inteligente. La Figura 28 ilustra un ejemplo de configuración del teléfono ex115 inteligente. El teléfono ex115 inteligente incluye la antena ex450 para transmitir y recibir ondas de radio a y desde la estación ex110 base, la cámara ex465 con capacidad de capturar vídeo e imágenes fijas y el visualizador ex458 que visualiza datos decodificados, tales como vídeo capturado por la cámara ex465 y vídeo recibido por la antena ex450. El teléfono ex115 inteligente incluye adicionalmente la interfaz ex466 de usuario, tal como un panel táctil, la unidad ex457 de salida de audio, tal como un altavoz para emitir voz u otro audio, la unidad ex456 de entrada de audio, tal como un micrófono para entrada de audio, la memoria ex467 con capacidad de almacenamiento de datos decodificados, tales como vídeo o imágenes fijas capturadas, audio grabado, vídeo o imágenes fijas recibidas y correo, así como datos decodificados, y la ranura ex464 que es una interfaz para el SIM ex468 para autorizar el acceso a una red y diversos datos. Obsérvese que puede usarse una memoria externa en lugar de la memoria ex467.

Además, el controlador ex460 principal que controla exhaustivamente el visualizador ex458 y la interfaz ex466 de usuario, el circuito ex461 de fuente de alimentación, el controlador ex462 de entrada de interfaz de usuario, el procesador ex455 de señales de vídeo, la interfaz ex463 de cámara, el controlador ex459 de visualización, el modulador/demodulador ex452, el multiplexor/demultiplexor ex453, el procesador ex454 de señales de audio, la ranura ex464 y la memoria ex467 se conectan a través del bus ex470.

Cuando el usuario enciende el botón de potencia del circuito ex461 de fuente de alimentación, el teléfono ex115 inteligente se activa en un estado operable siendo cada componente suministrado con potencia desde un conjunto de baterías.

El teléfono ex115 inteligente realiza un procesamiento, por ejemplo, de llamada y transmisión de datos, a base de control realizado por el controlador ex460 principal, que incluye una CPU, ROM y RAM. Cuando se hacen llamadas, una señal de audio grabada por la unidad ex456 de entrada de audio se convierte a una señal de audio digital por el procesador ex454 de señales de audio, y esto se aplica con procesamiento de espectro ensanchado por el modulador/demodulador ex452 y procesamiento de conversión digital-analógica y de conversión de frecuencia por el transmisor/receptor ex451 y, a continuación, se transmite a través de la antena ex450. Los datos recibidos se amplifican, convierten en frecuencia y convierten en analógico-digital, procesan por espectro ensanchado inverso por el modulador/demodulador ex452, convierten a una señal de audio

analógica por el procesador ex454 de señales de audio y, a continuación, emiten desde la unidad ex457 de salida de audio. En modo de transmisión de datos, texto, imagen fija o datos de vídeo se transmiten por el controlador ex460 principal a través del controlador ex462 de entrada de interfaz de usuario como resultado de operación de, por ejemplo, la interfaz ex466 de usuario del cuerpo principal, y se realiza un procesamiento de transmisión y recepción similar. En modo de transmisión de datos, cuando se envía un vídeo, imagen fija o vídeo y audio, la compresión del procesador ex455 de señales de vídeo codifica, a través del procedimiento de codificación de instantáneas en movimiento descrito en las realizaciones anteriores y variaciones, una señal de vídeo almacenada en la memoria ex467 o una señal de vídeo introducida desde la cámara ex465, y transmite los datos de vídeo codificados al multiplexor/demultiplexor ex453. Además, el procesador ex454 de señales de audio codifica una señal de audio grabada por la unidad ex456 de entrada de audio mientras la cámara ex465 está capturando, por ejemplo, un vídeo o imagen fija, y transmite los datos de audio codificados al multiplexor/demultiplexor ex453. El multiplexor/demultiplexor ex453 multiplexa los datos de vídeo codificados y datos de audio codificados usando un esquema predeterminado, modula y convierte los datos usando el modulador/demodulador ex452 (circuito de modulador/demodulador) y el transmisor/receptor ex451, y transmite el resultado a través de la antena ex450.

Cuando se recibe vídeo adjunto en un correo electrónico o un chat, o un vídeo 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 vídeo y un flujo de bits de datos de audio, suministra los datos de vídeo codificados al procesador ex455 de señales de vídeo a través de bus síncrono ex470, y suministra los datos de audio codificados al procesador ex454 de señales de audio a través del bus síncrono ex470. El procesador ex455 de señales de vídeo decodifica la señal de vídeo usando un procedimiento de decodificación de instantáneas en movimiento que corresponde al procedimiento de codificación de instantáneas en movimiento descrito en las realizaciones anteriores y variaciones, y vídeo o una imagen fija incluida en el archivo de instantánea en movimiento enlazado se visualiza en el visualizador ex458 a través del controlador ex459 de visualización. Además, el procesador ex454 de señales de audio decodifica la señal de audio y emite audio desde la unidad ex457 de salida de audio. Obsérvese que dado que la difusión en continuo en tiempo real se está volviendo cada vez más popular, existen casos en los que la reproducción del audio puede ser socialmente inapropiado dependiendo del entorno del usuario. Por consiguiente, como un valor inicial, es preferible una configuración en la que únicamente se reproducen los datos de vídeo, es decir, la señal de audio no se reproduce. El audio puede sincronizarse y reproducirse únicamente cuando se recibe una entrada, tal como cuando el usuario clica en datos de vídeo.

Aunque se usó el teléfono ex115 inteligente en el ejemplo anterior, son concebibles tres implementaciones: un transceptor terminal que incluye tanto un codificador como un decodificador; un transmisor terminal que incluye únicamente un codificador; y un receptor terminal que incluye únicamente un decodificador. Además, en la descripción del sistema de difusión digital, se proporciona un ejemplo en el que los datos multiplexados obtenidos como resultado de datos de vídeo que se multiplexan con, por ejemplo, datos de audio, se reciben o transmiten, pero los datos multiplexados pueden ser datos de vídeo multiplexados con datos distintos de datos de audio, tales como datos de texto relacionados con el vídeo. Además, pueden recibirse o transmitirse los propios datos de vídeo en lugar de los datos multiplexados.

Aunque el controlador ex460 principal que incluye una CPU se describe como que controla los procedimientos de codificación o decodificación, los terminales a menudo incluyen una GPU. Por consiguiente, una configuración es aceptable en la que una gran área se procesa de una vez haciendo uso de la capacidad de rendimiento de la GPU a través de memoria compartida por la CPU y GPU o memoria que incluye una dirección que se gestiona para permitir un uso común por la CPU y GPU. Esto hace posible acortar el tiempo de codificación, mantener la naturaleza en tiempo real del flujo y reducir el retardo. En particular, un procedimiento relacionado con una estimación de movimiento, filtrado de desbloqueo, compensación adaptativa de muestra (SAO) y transformación/cuantificación puede efectuarse de forma efectiva por la GPU en lugar de la CPU en unidades de, por ejemplo, instantáneas, todas de una vez.

50 Aplicabilidad industrial

La presente divulgación es aplicable a, por ejemplo, receptores de televisión, grabadores de vídeo digital, sistemas de navegación de coche, teléfonos móviles, cámaras digitales y cámaras de vídeo digitales, etc.

Marcas de referencia en los dibujos

- 100 codificador
- 102 divisor
- 104 restador
- 106 transformador

| | |
|----------|----------------------------|
| 108 | cuantificador |
| 110 | codificador por entropía |
| 112, 204 | cuantificador inverso |
| 114, 206 | transformador inverso |
| 116, 208 | sumador |
| 118, 210 | memoria de bloque |
| 120, 212 | filtro de bucle |
| 122, 214 | memoria de fotograma |
| 124, 216 | intra predictor |
| 126, 218 | inter predictor |
| 128, 220 | controlador de predicción |
| 200 | decodificador |
| 202 | decodificador por entropía |

REIVINDICACIONES

1. Un codificador (100) que codifica un primer bloque actual y un segundo bloque actual en una instantánea, comprendiendo el codificador:

5 circuitería (106, 108); y
 memoria (118, 122), en el que
 usando la memoria, la circuitería está configurada para:

- determinar una primera base de transformación utilizada para una primera transformación;

el codificador **se caracteriza por**:

10 - (i) realizar la primera transformación en una primera señal residual del primer bloque actual utilizando la primera base de transformación para generar los primeros coeficientes de transformación; realizar una segunda transformación en los primeros coeficientes de transformación utilizando una segunda base de transformación para generar los segundos coeficientes de transformación y cuantificar los segundos coeficientes de transformación, cuando la primera base de transformación determinada es la misma que una base de transformación predeterminada;

15 - (ii) realizar la primera transformación en una segunda señal residual del segundo bloque actual utilizando la primera base de transformación para generar los primeros coeficientes de transformación; y cuantificar los primeros coeficientes de transformación sin realizar una segunda transformación, cuando la primera base de transformación determinada es diferente de la base de transformación predeterminada; y

20 - la primera señal residual y la segunda señal residual son señales luma.

2. Un decodificador (200) que decodifica un primer bloque actual y un segundo bloque actual en una instantánea, comprendiendo el decodificador:

25 circuitería (204, 206); y
 memoria (210, 214); en la que
 utilizando la memoria, la circuitería está configurada para:

- determinar una primera base de transformación inversa utilizada para una primera transformación inversa;

el decodificador **se caracteriza por**:

30 - (i) cuantificar de forma inversa los coeficientes cuantificados del primer bloque actual para generar segundos coeficientes de transformación; realizar una segunda transformación inversa sobre los segundos coeficientes de transformación utilizando una segunda base de transformación inversa para generar primeros coeficientes de transformación y realizar la primera transformación inversa sobre los primeros coeficientes de transformación utilizando la primera base de transformación inversa para generar una primera señal residual del primer bloque actual, cuando la primera base de transformación inversa determinada es la misma que una base de transformación inversa predeterminada; y

35 - (ii) cuantificar de forma inversa los coeficientes cuantificados del segundo bloque actual para generar primeros coeficientes de transformación; y realizar la primera transformación inversa sobre los primeros coeficientes de transformación utilizando la primera base de transformación inversa para generar una segunda señal residual del segundo bloque actual sin realizar una segunda transformación inversa, cuando la primera base de transformación inversa determinada es diferente de la base de transformación inversa predeterminada, y

40 - la primera señal residual y la segunda señal residual son señales luma.

3. Un procedimiento de codificación para codificar un primer bloque actual y un segundo bloque actual en una instantánea, comprendiendo el procedimiento de codificación:

- 45 - determinar una primera base de transformación utilizada para una primera transformación;

el procedimiento de codificación **se caracteriza por**:

50 - (i) realizar la primera transformación en una primera señal residual del primer bloque actual utilizando la primera base de transformación para generar primeros coeficientes de transformación; realizar una segunda transformación en los primeros coeficientes de transformación utilizando una segunda base de transformación para generar segundos coeficientes de transformación y cuantificar los segundos coeficientes de transformación, cuando la primera base de transformación determinada es la misma que

una base de transformación predeterminada;

- (ii) realizar la primera transformación en una segunda señal residual del segundo bloque actual utilizando la primera base de transformación para generar primeros coeficientes de transformación; y cuantificar unos primeros coeficientes de transformación sin realizar una segunda transformación, cuando la primera base de transformación determinada es diferente de la base de transformación predeterminada; y en la que

- la primera señal residual y la segunda señal residual son señales luma.

4. Un procedimiento de decodificación para decodificar un primer bloque actual y un segundo bloque actual en una instantánea, comprendiendo el método de decodificación:

10 - determinar una primera base de transformación inversa utilizada para una primera transformación inversa;

el método de decodificación **se caracteriza por**:

- (i) cuantificar de forma inversa los coeficientes cuantificados del primer bloque actual para generar segundos coeficientes de transformación; realizar una segunda transformación inversa sobre los segundos coeficientes de transformación utilizando una segunda base de transformación inversa para generar primeros coeficientes de transformación y realizar la primera transformación inversa sobre los primeros coeficientes de transformación utilizando la primera base de transformación inversa para generar una primera señal residual del primer bloque actual, cuando la primera base de transformación inversa determinada es la misma que una base de transformación inversa predeterminada; y

15 - (ii) cuantificar de forma inversa los coeficientes cuantificados del segundo bloque actual para generar primeros coeficientes de transformación; y realizar la primera transformación inversa sobre los primeros coeficientes de transformación utilizando la primera base de transformación inversa para generar una segunda señal residual del segundo bloque actual sin realizar la segunda transformación inversa, cuando la primera base de transformación inversa determinada es diferente de la base de transformación inversa predeterminada, y en la que

20 - la primera señal residual y la segunda señal residual son señales luma.

.

FIG. 1

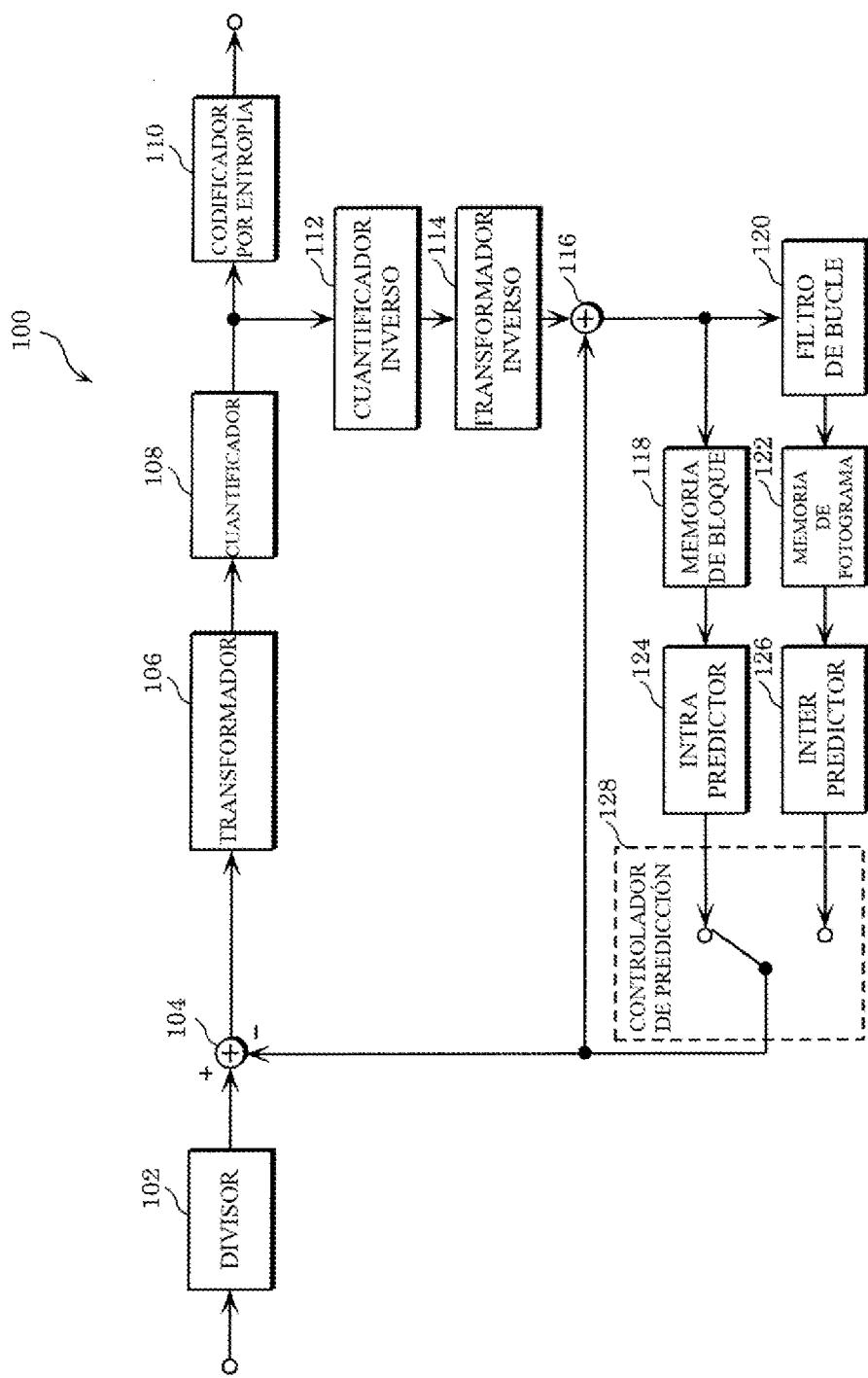


FIG. 2

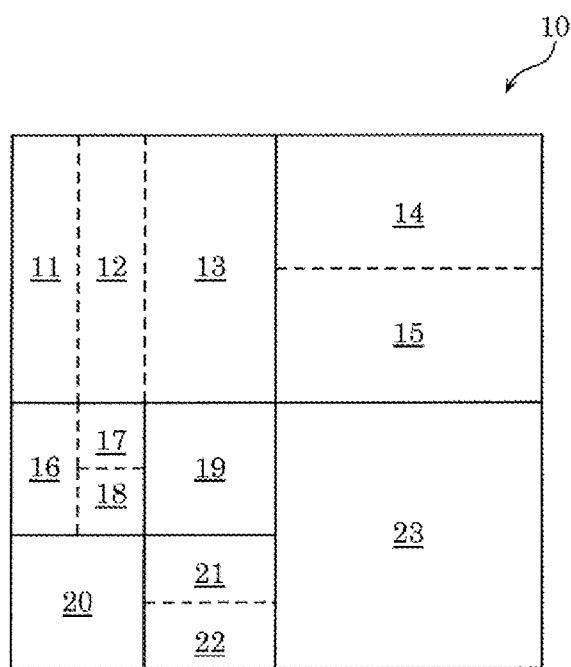


FIG. 3

| TIPO DE TRANSFORMADA | FUNCIÓN DE 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 style="text-align: center;">DONDE $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$</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 style="text-align: center;">DONDE $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$, $\omega_1 = \begin{cases} \sqrt{\frac{2}{N}} & j = 0 \\ 1 & j \neq 0 \end{cases}$</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

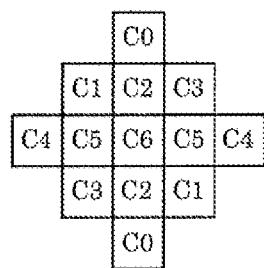


FIG. 4B

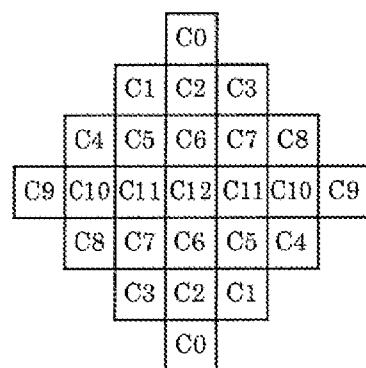


FIG. 4C

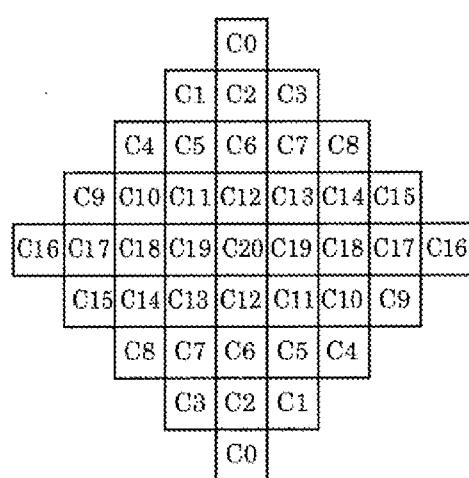


FIG. 5A

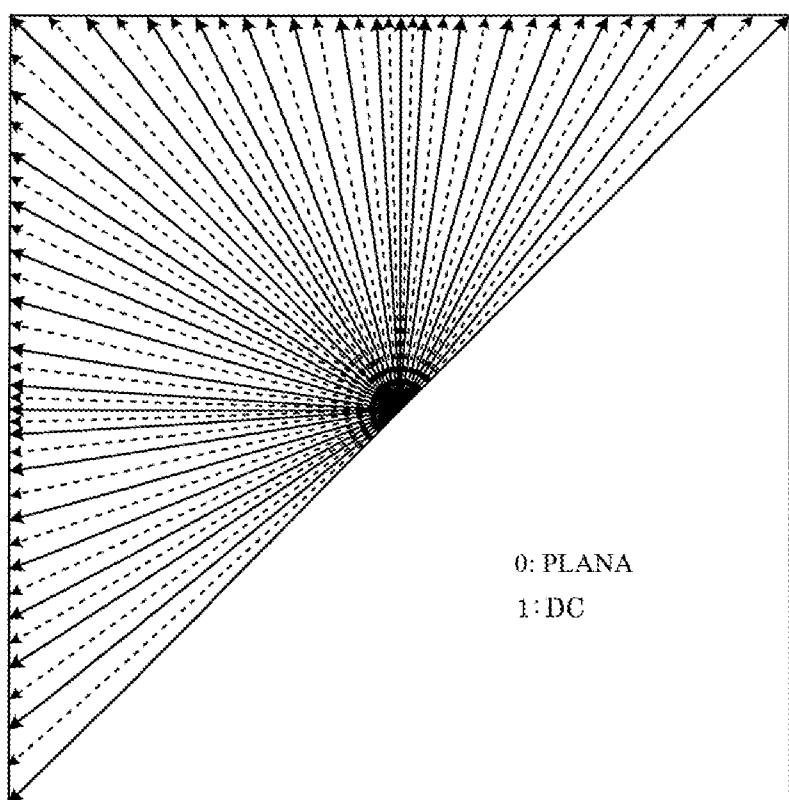


FIG. 5B

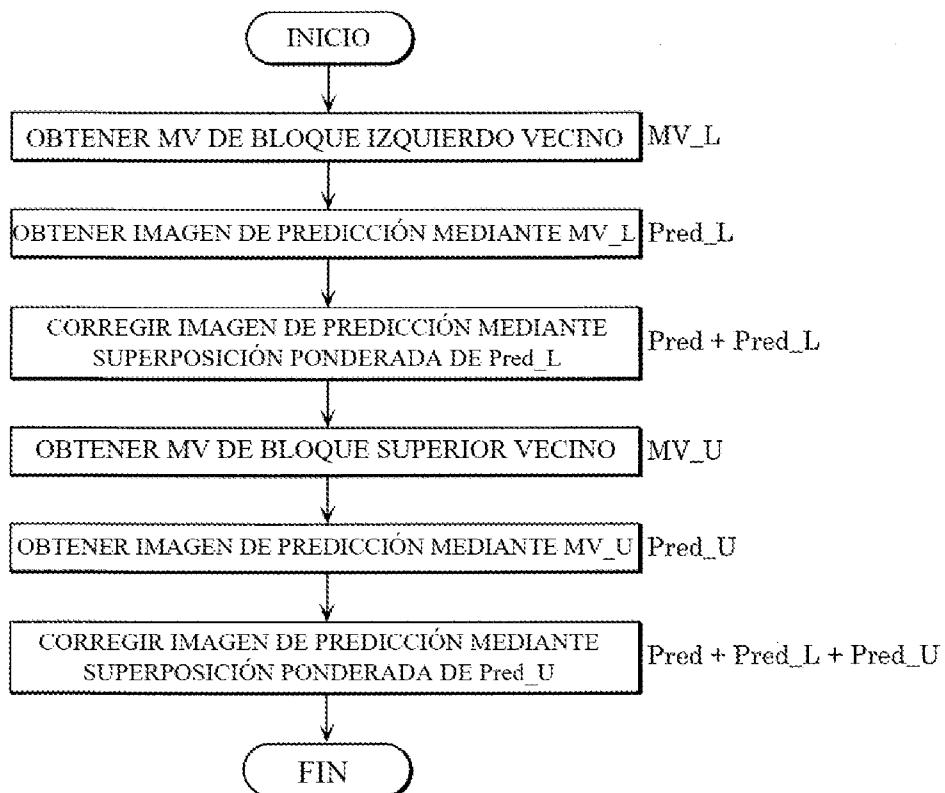


FIG. 5C

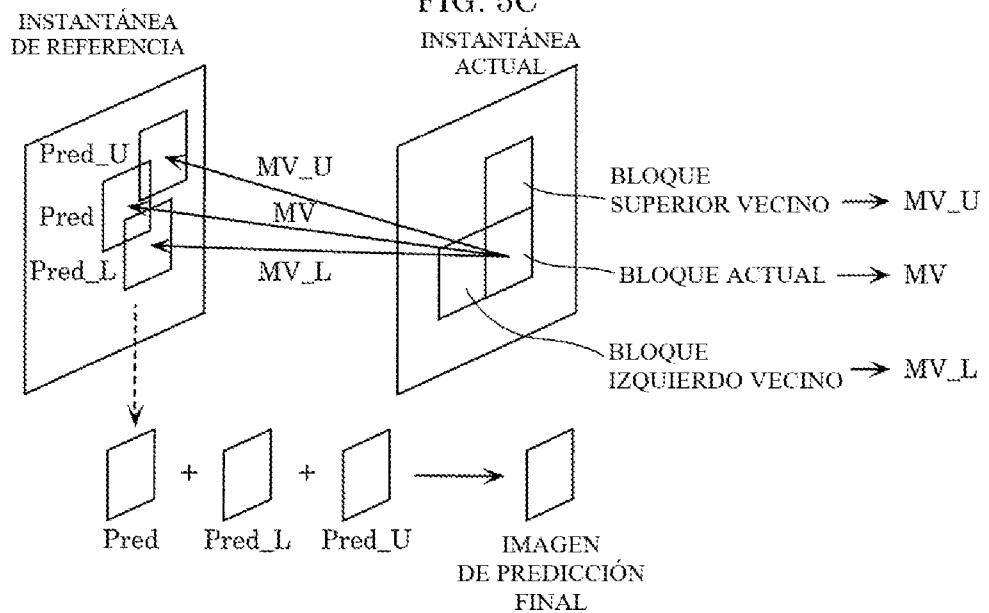


FIG. 5D

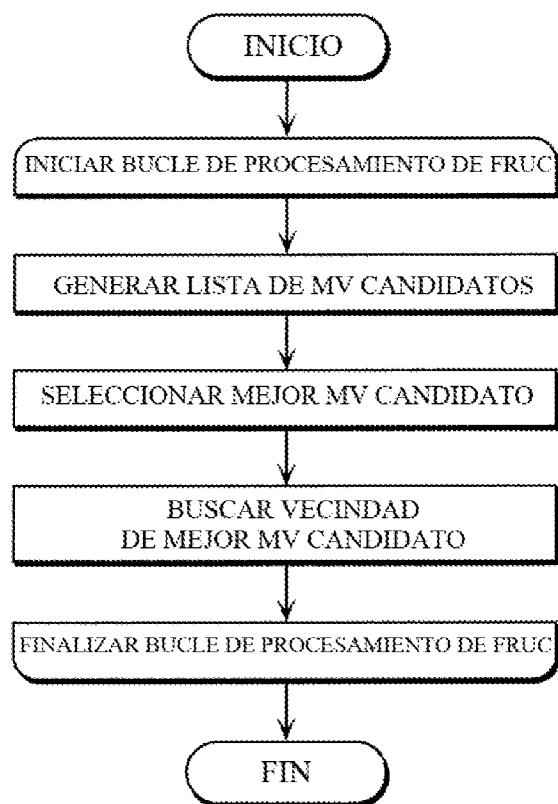


FIG. 6

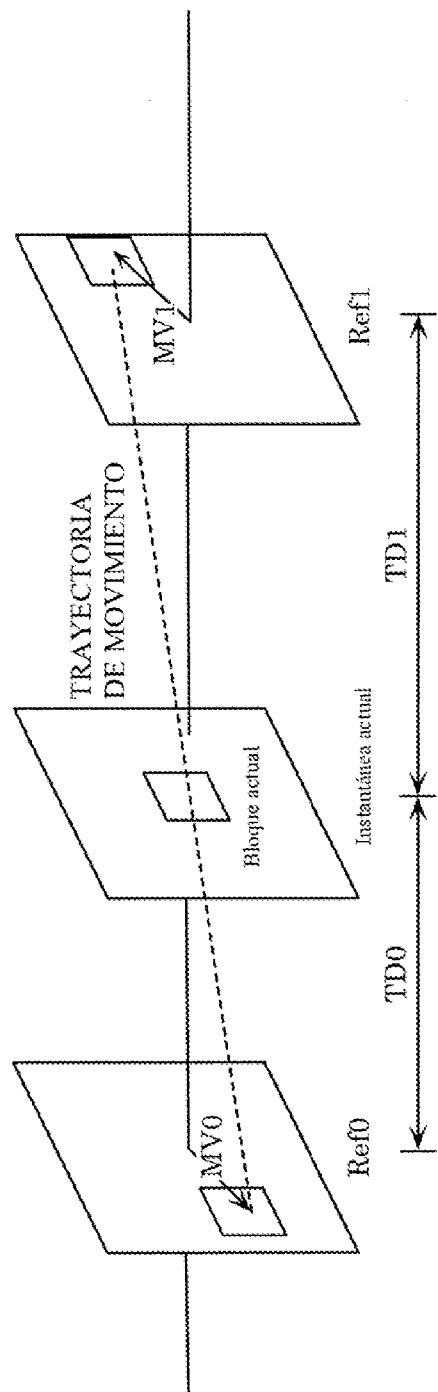


FIG. 7

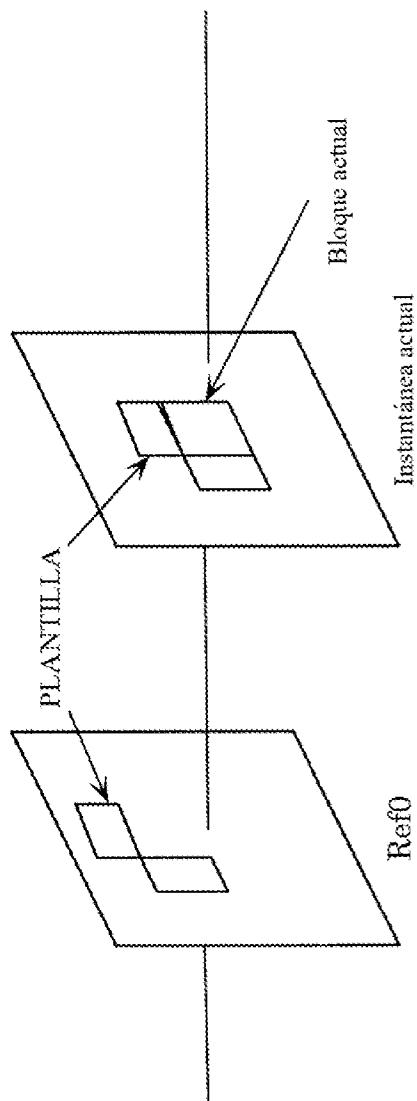


FIG. 8

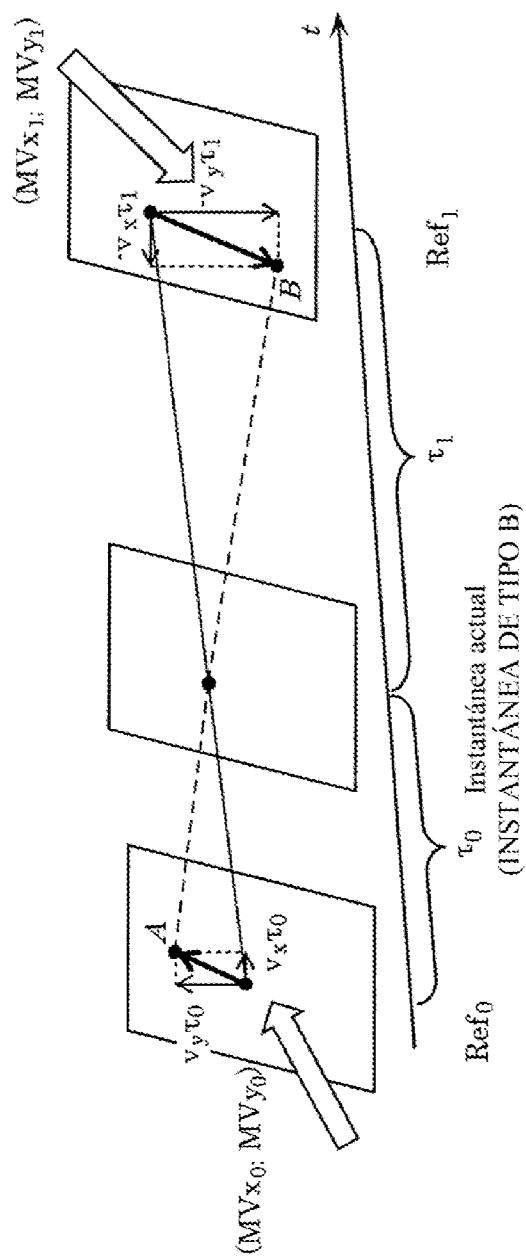


FIG. 9A

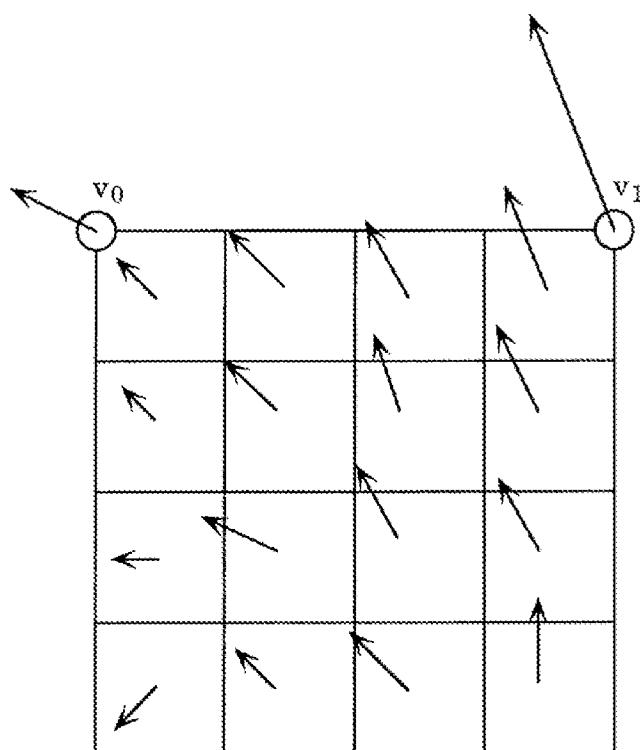


FIG. 9B

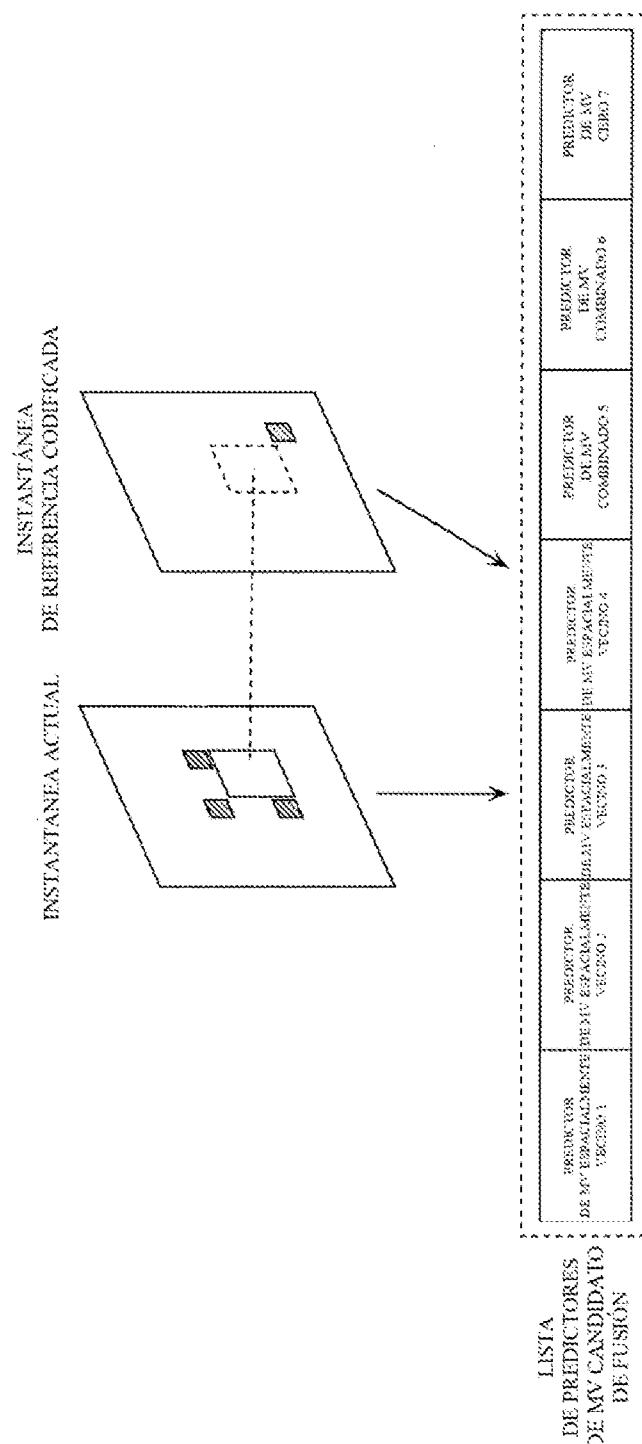


FIG. 9C

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

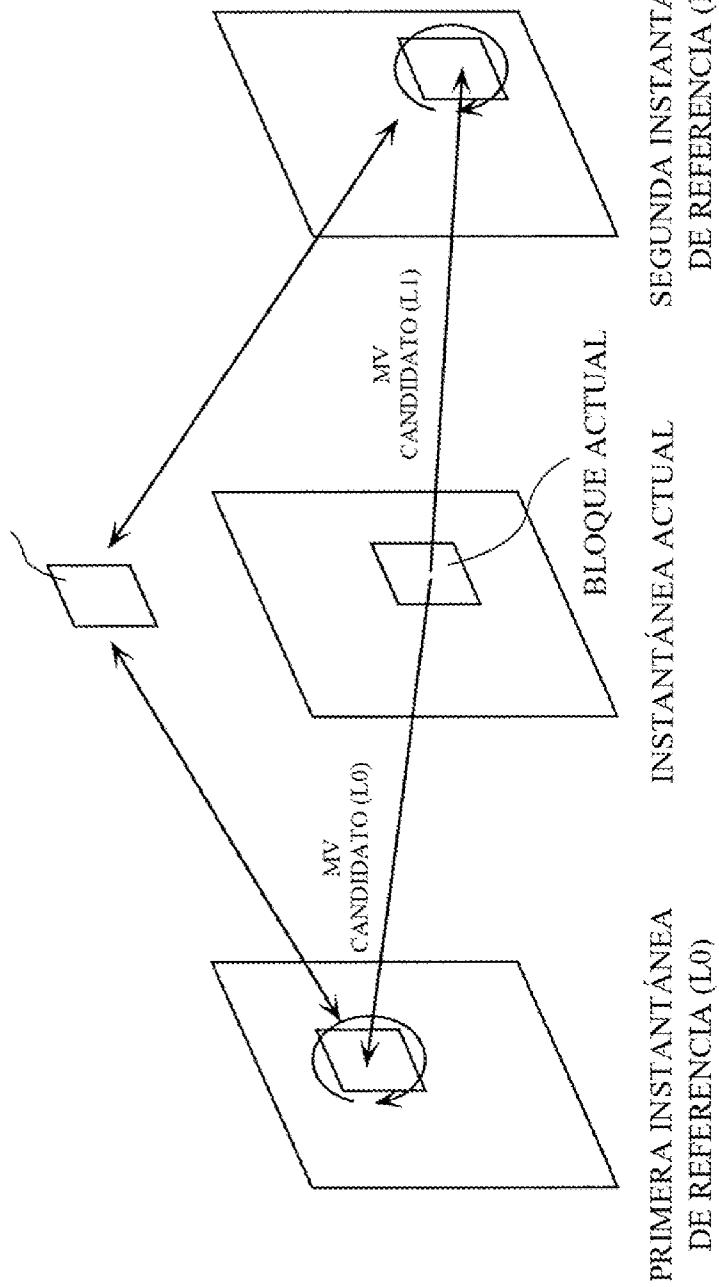


FIG. 9D

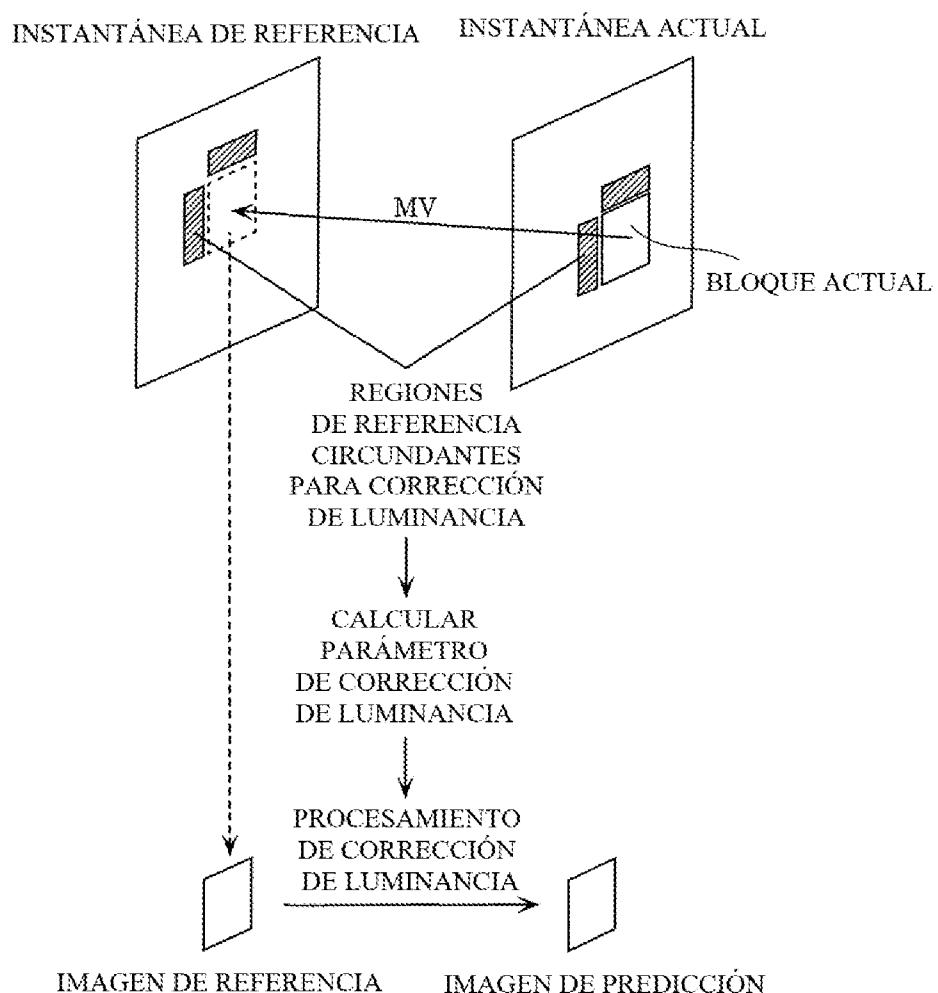


FIG. 10

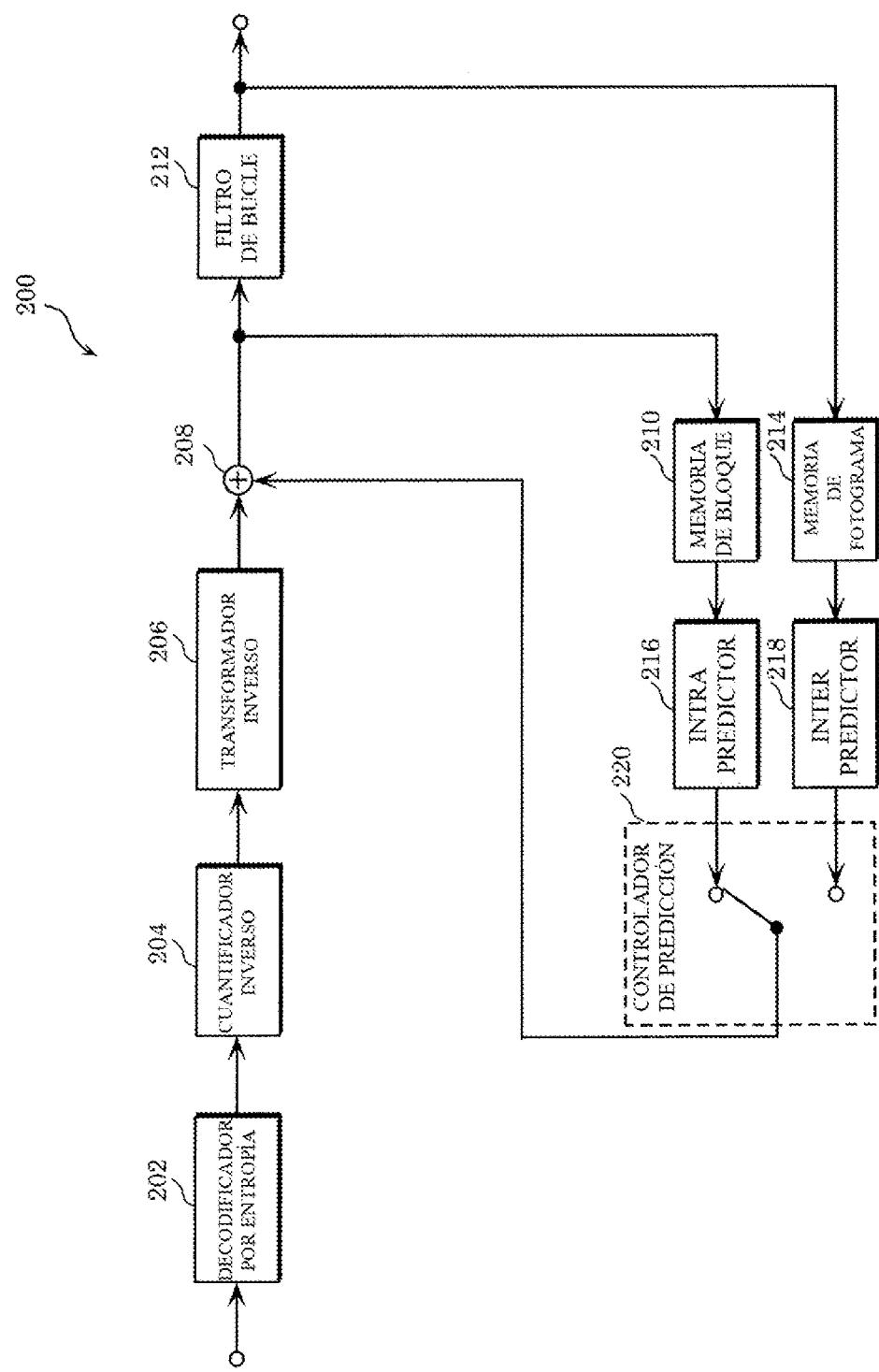


FIG. 11

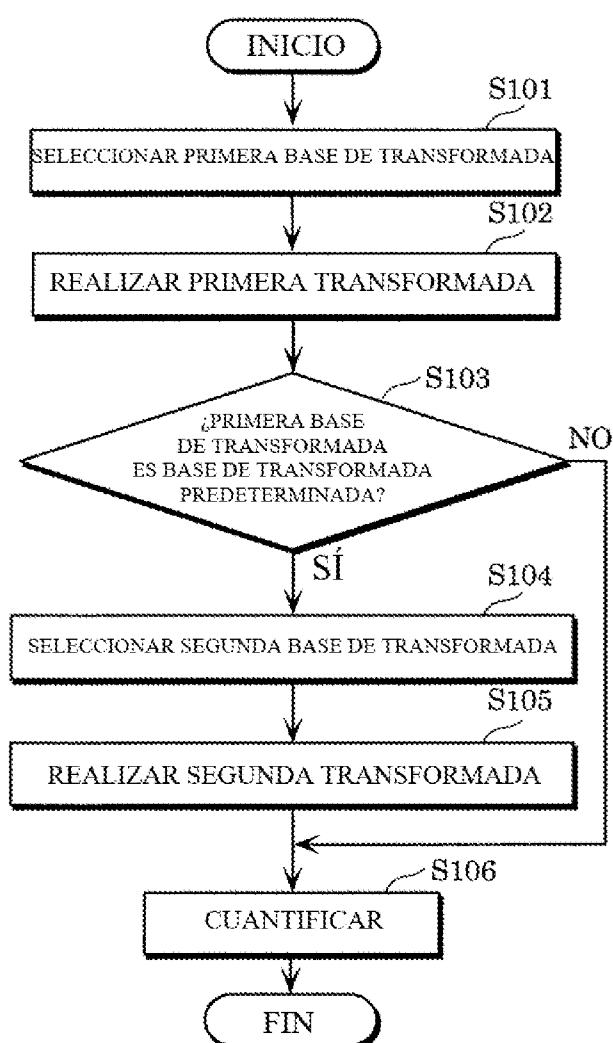


FIG. 12

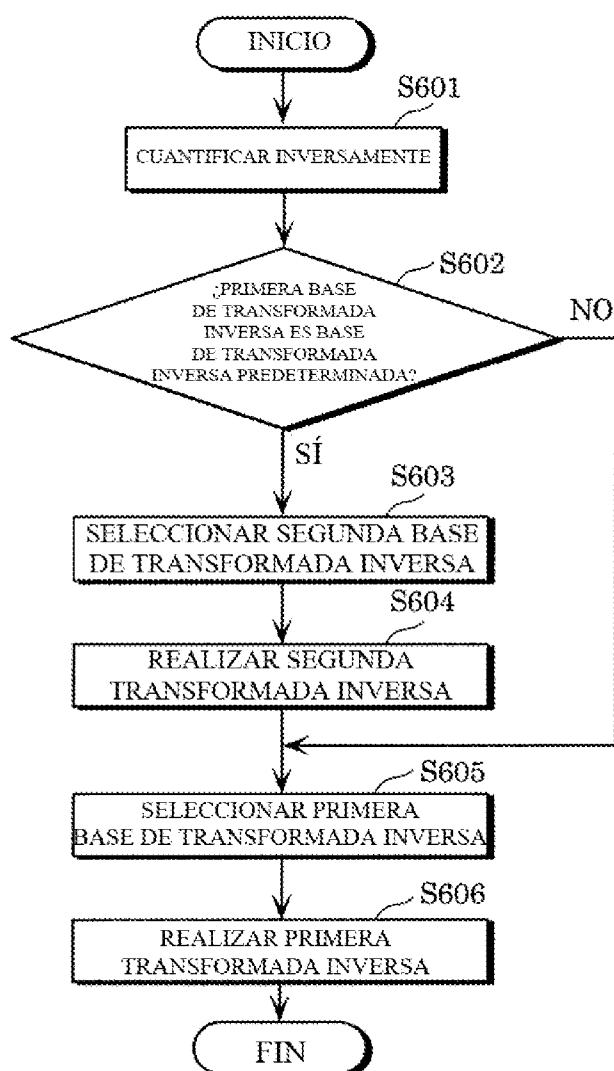


FIG. 13

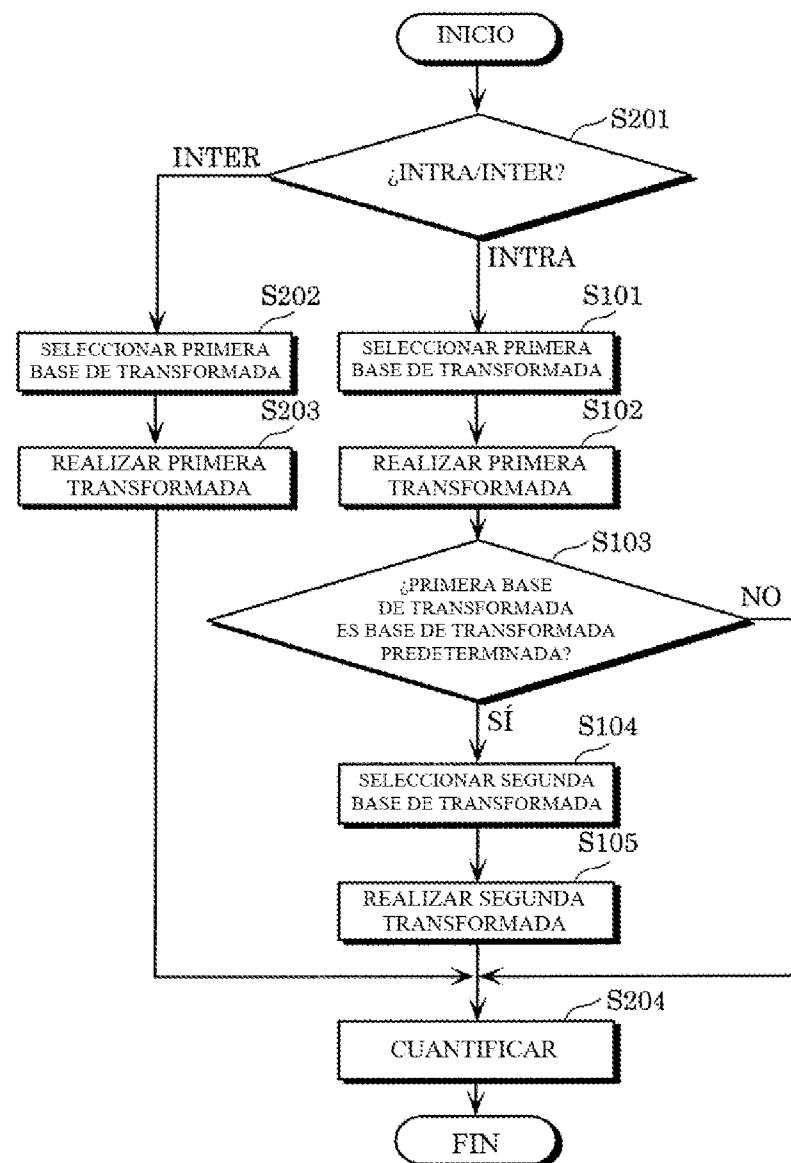


FIG. 14

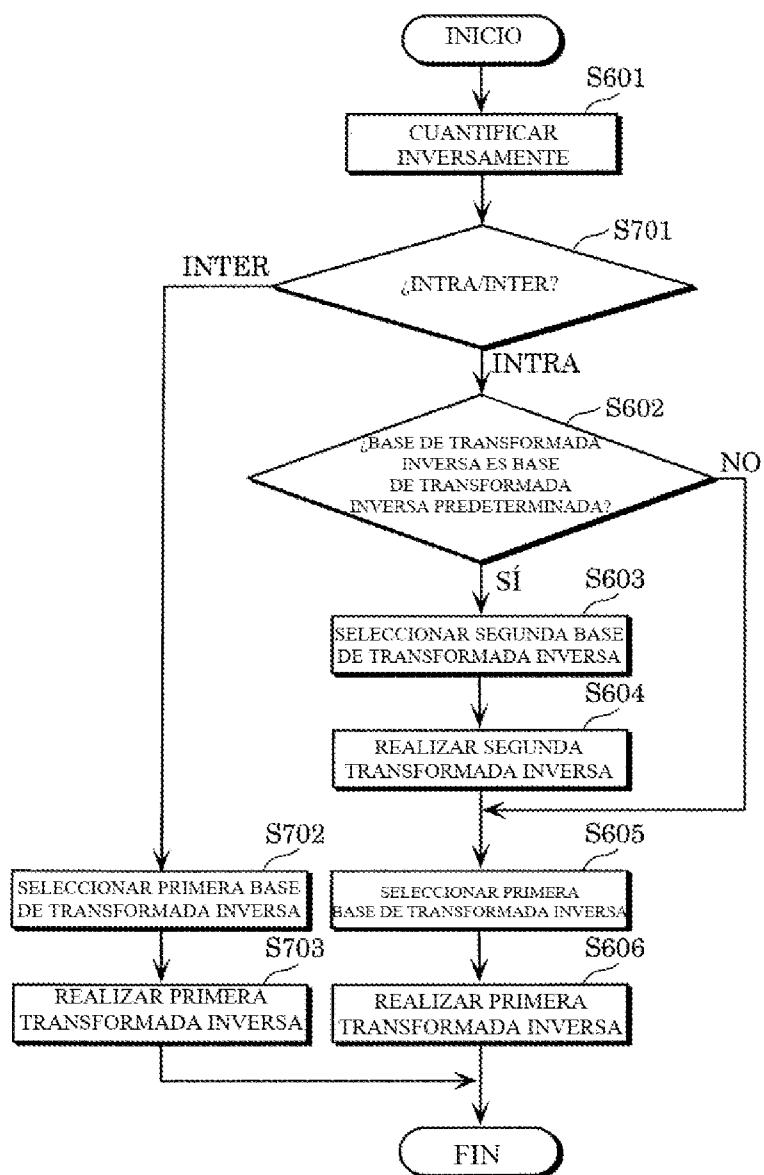


FIG. 15

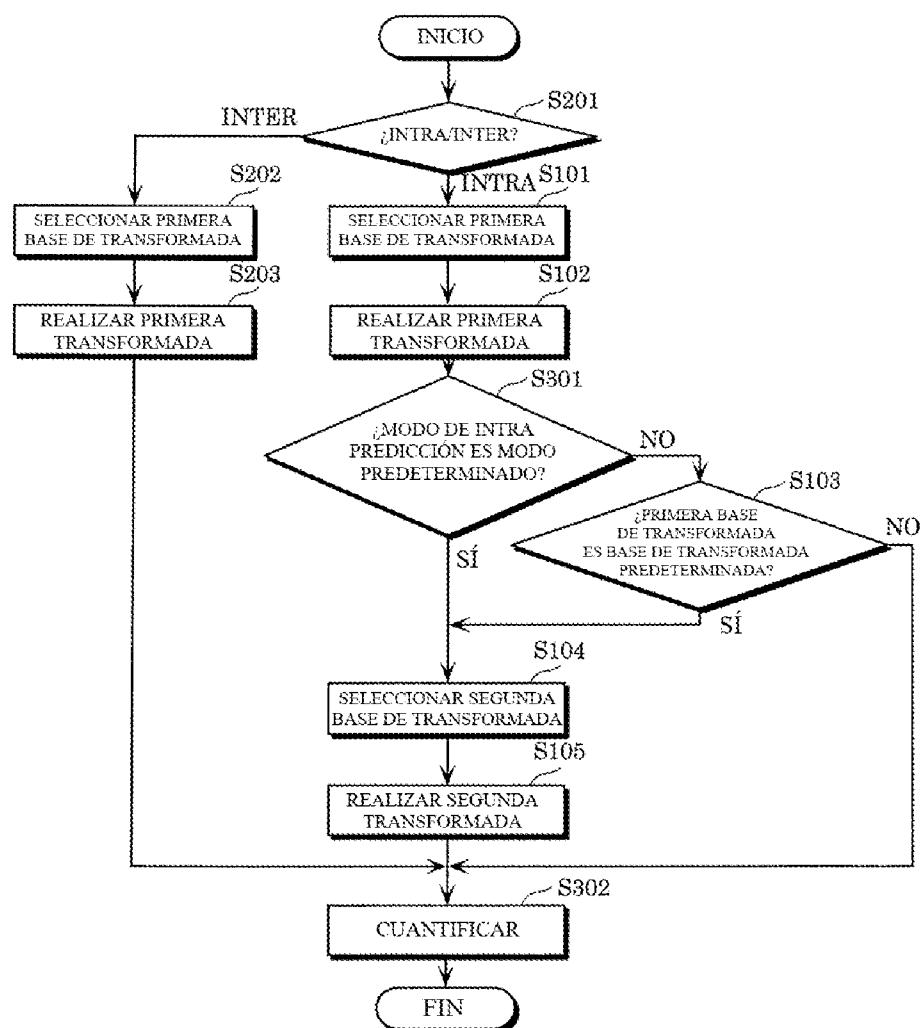


FIG. 16

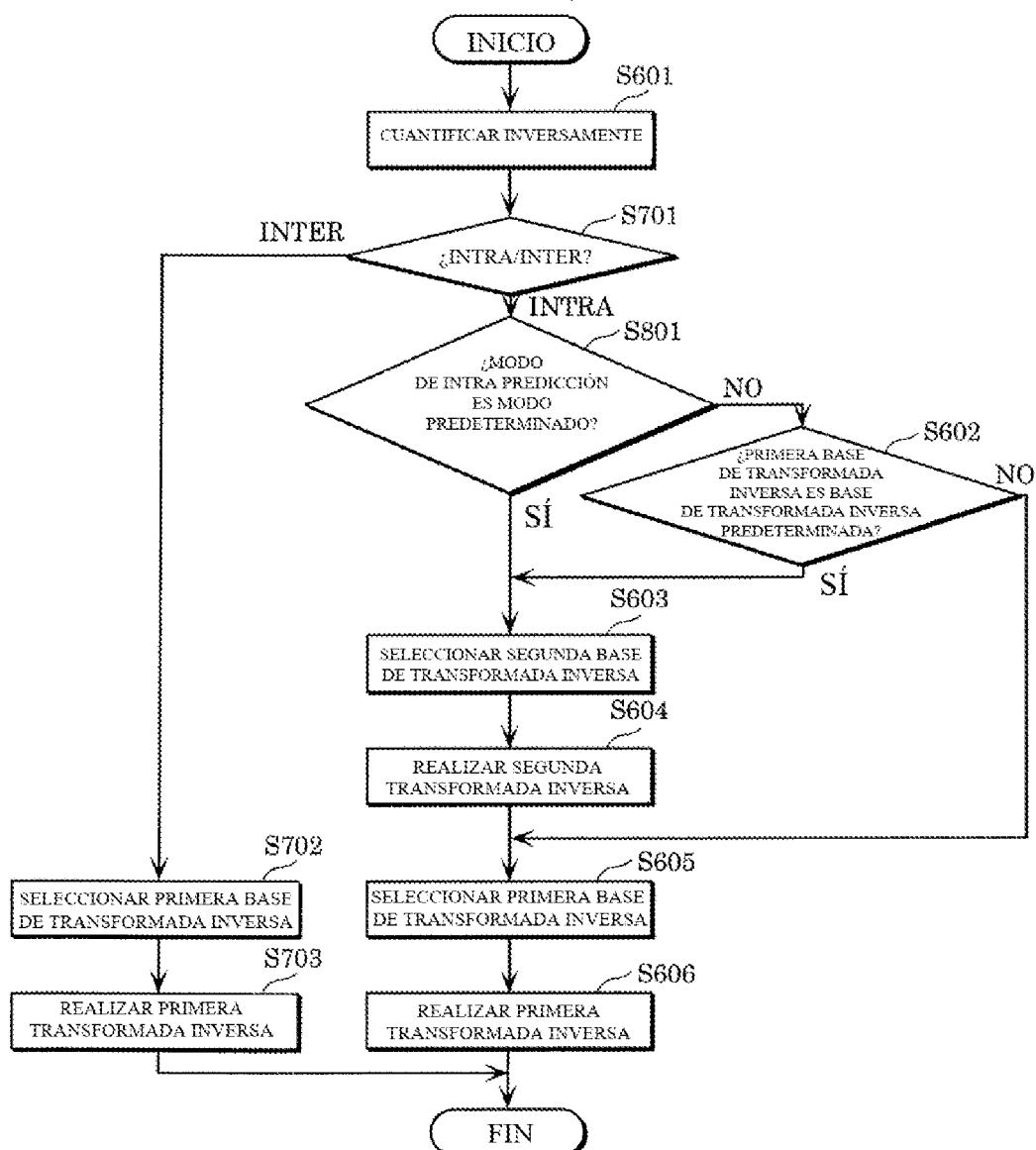


FIG. 17

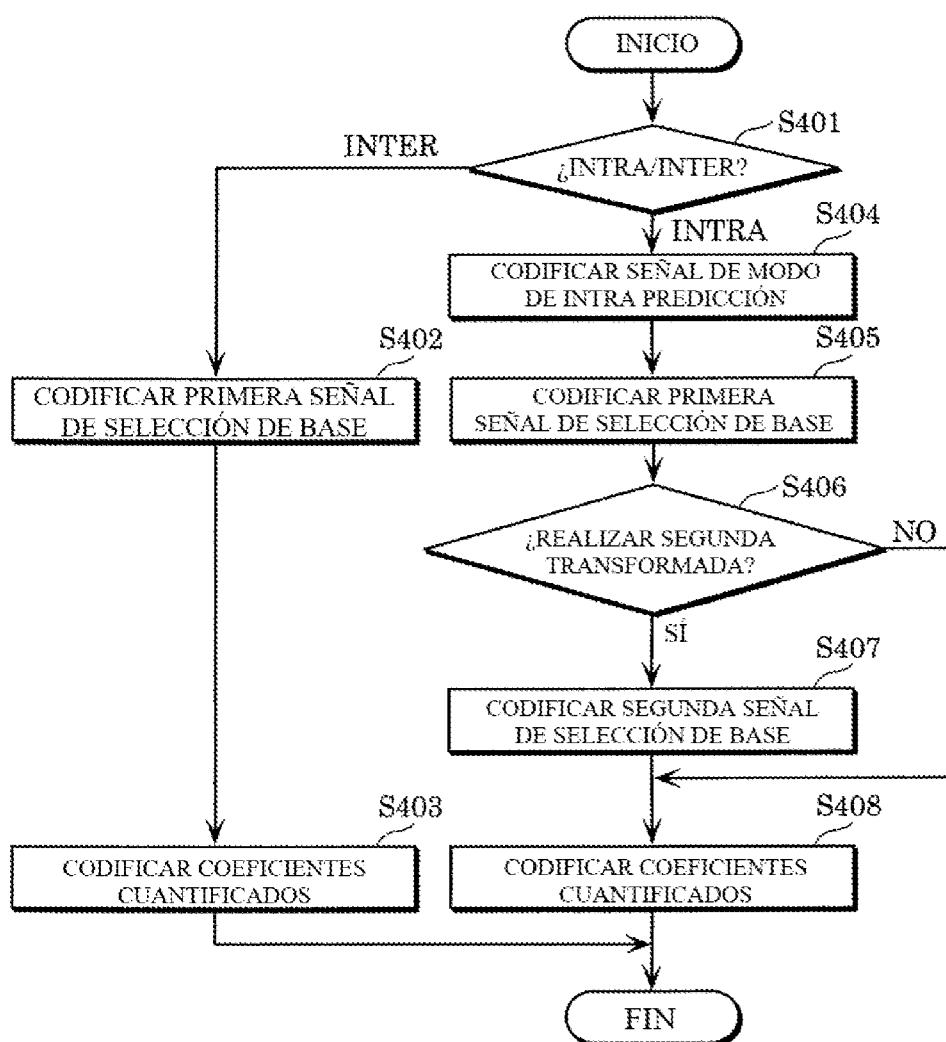


FIG. 18

```

coding_unit()
{
    pred_mode          // INTRA o INTER
    ;
    pred_mode_dir      // Modo de intra predicción
    ;
    emit_mode          // EMT en (DCT-5/DCT-8/DST-1/DTS-7) o off(DCT-2)
    if(emit_mode)
    {
        primary_transform_type // Tipo de transformada primario
    }
    ;
    if(pred_mode == MODE_INTRA)
    {
        if(!emit_mode) || (pred_mode_dir == 2) || (pred_mode_dir == 34) || (pred_mode_dir == 66))
        {
            secondary_transform_type
        } else {
            // Sin serialización
        }
    }
}

```

FIG. 19

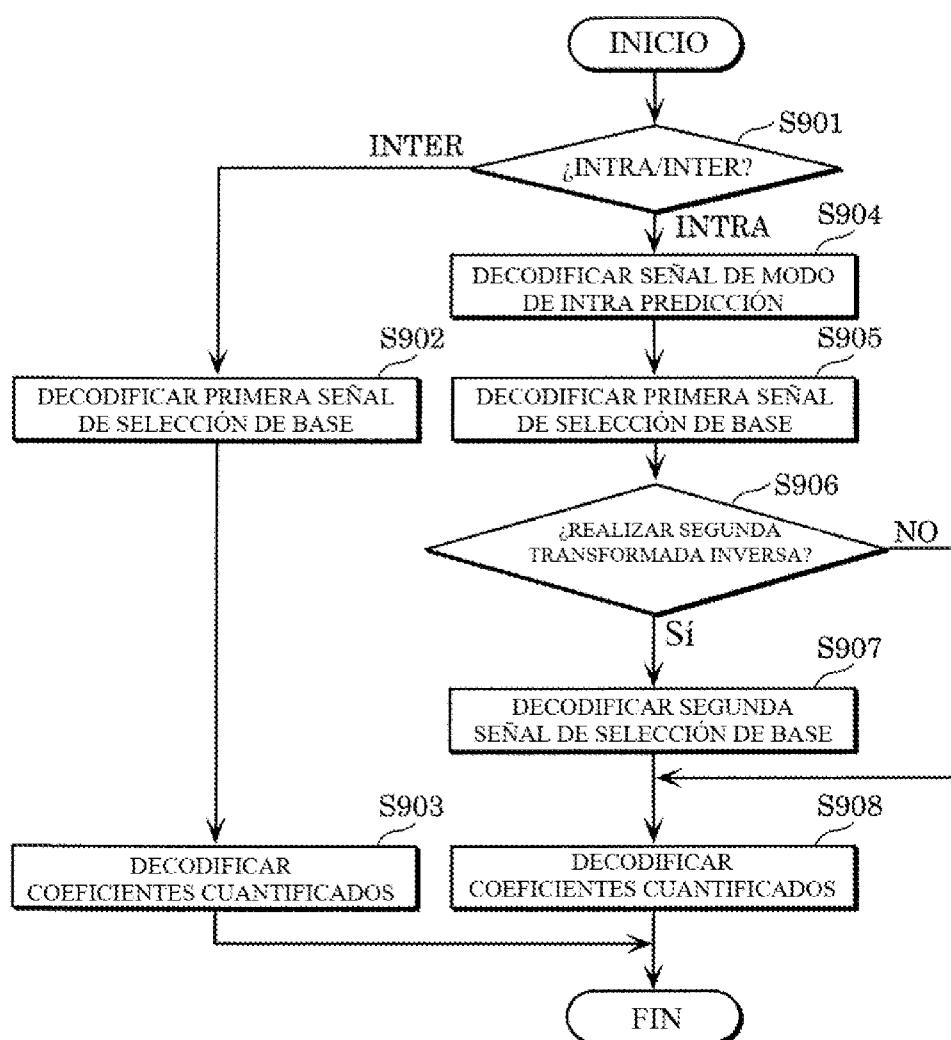


FIG. 20

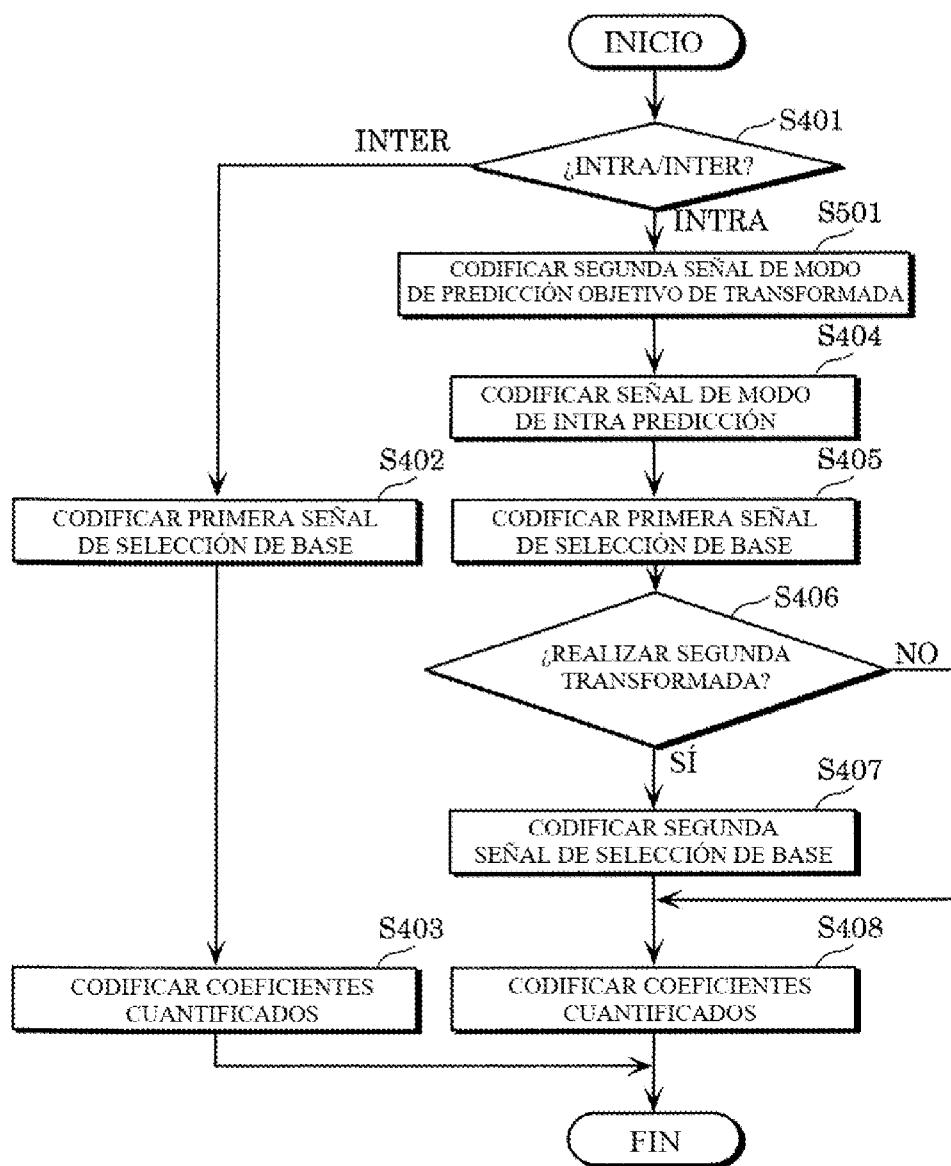


FIG. 21

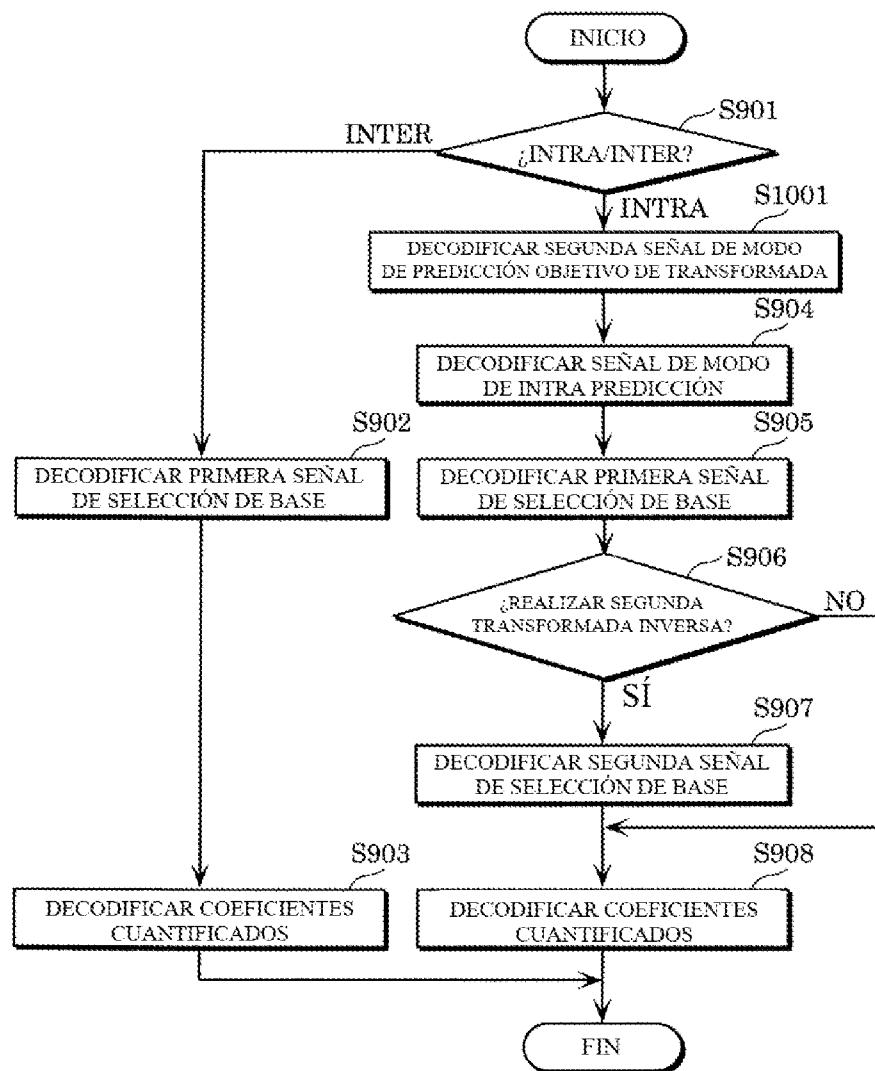


FIG. 22

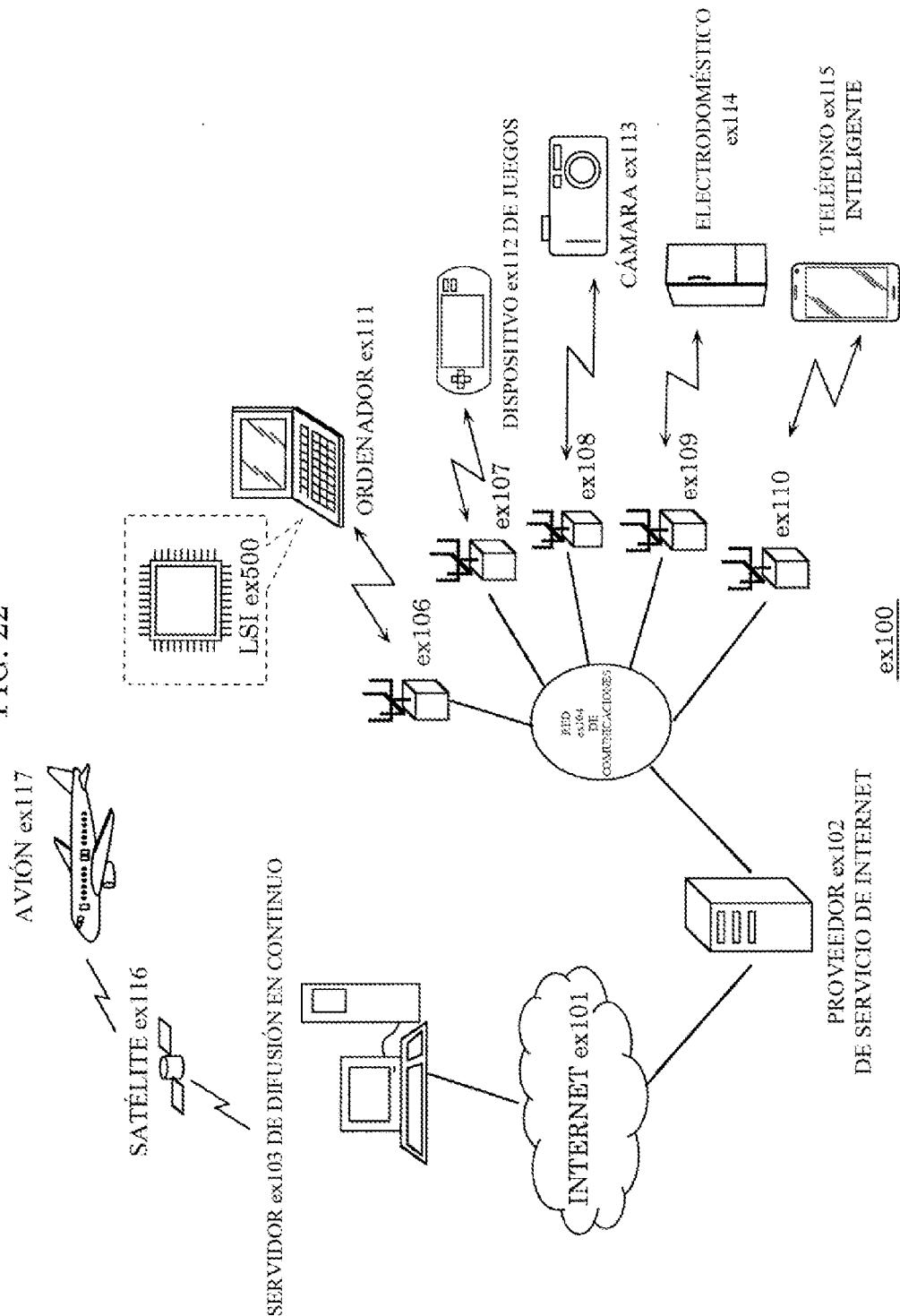


FIG. 23

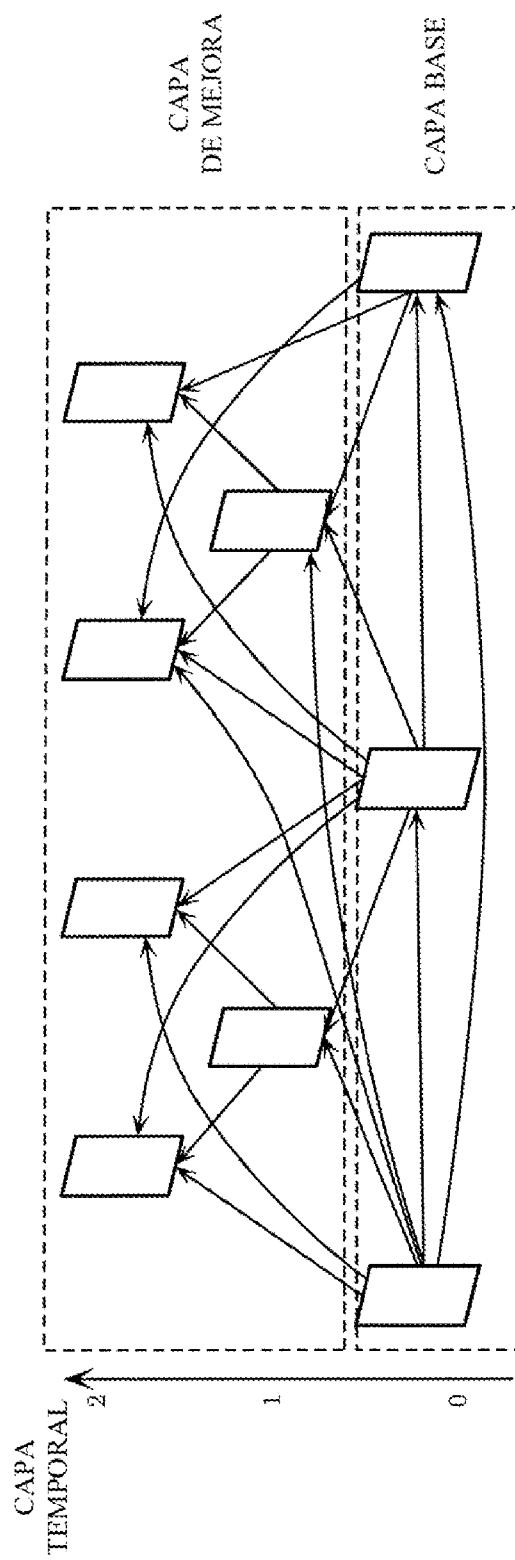


FIG. 24

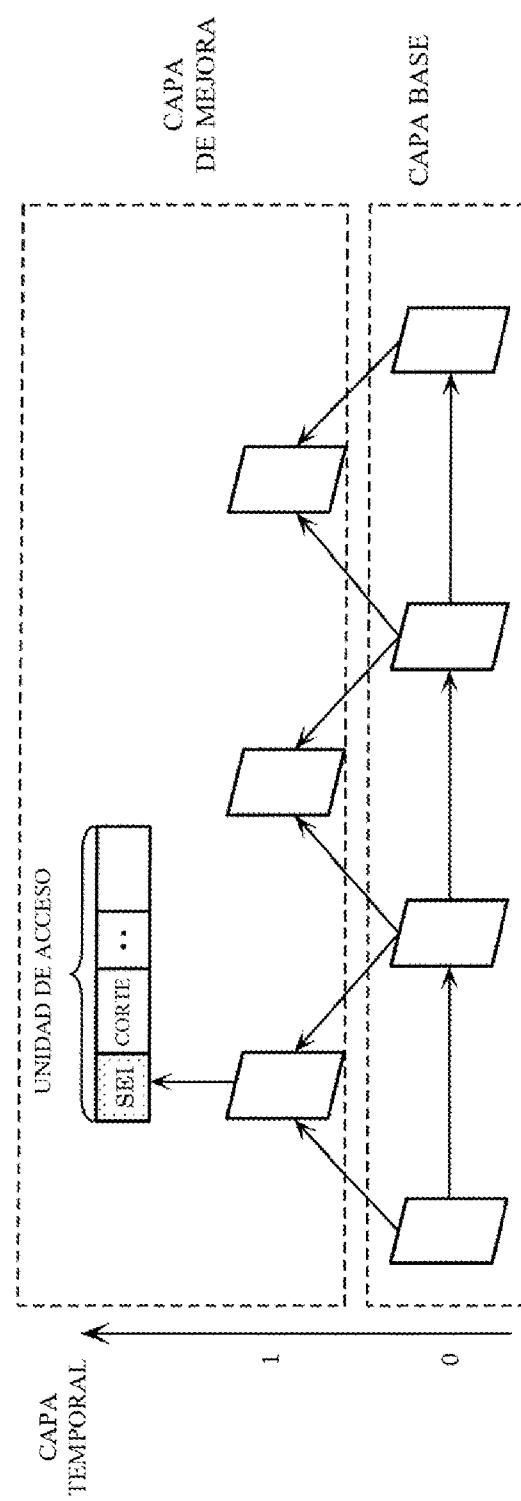


FIG. 25

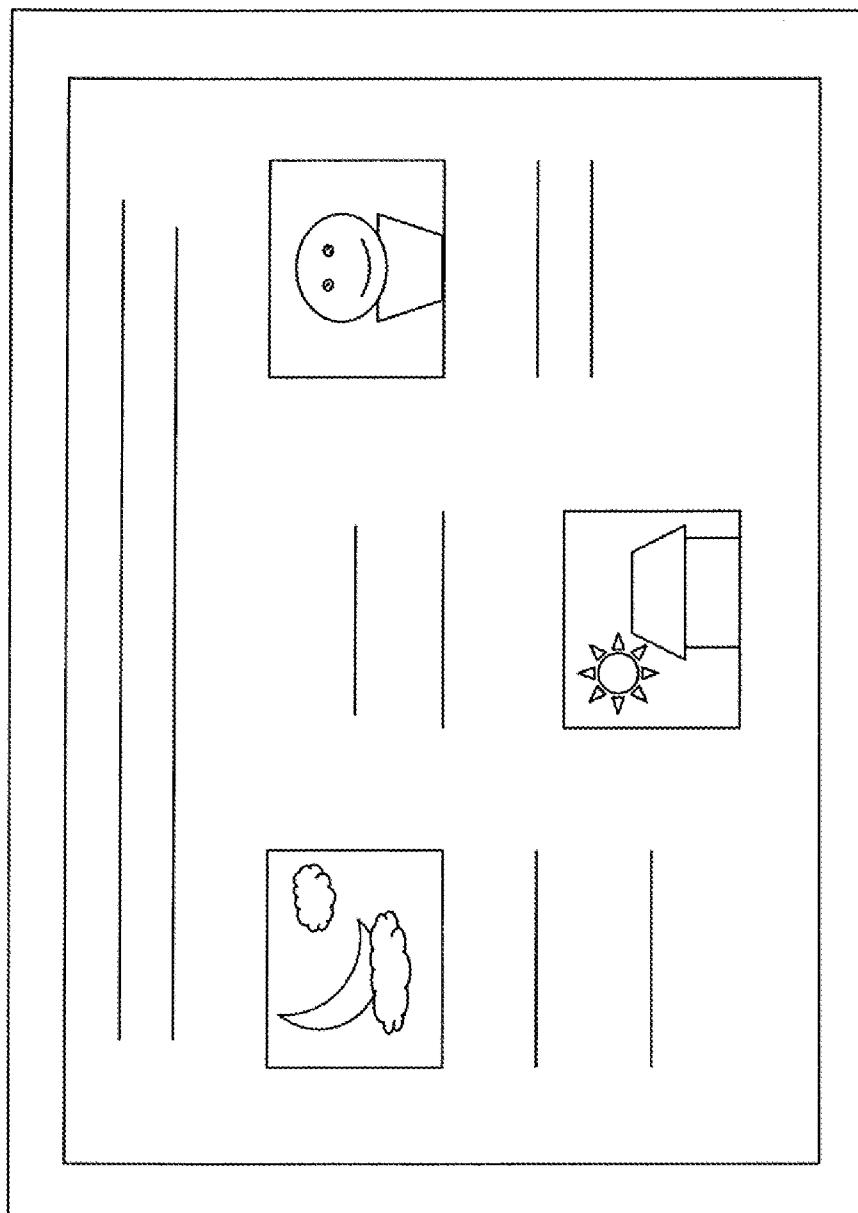


FIG. 26

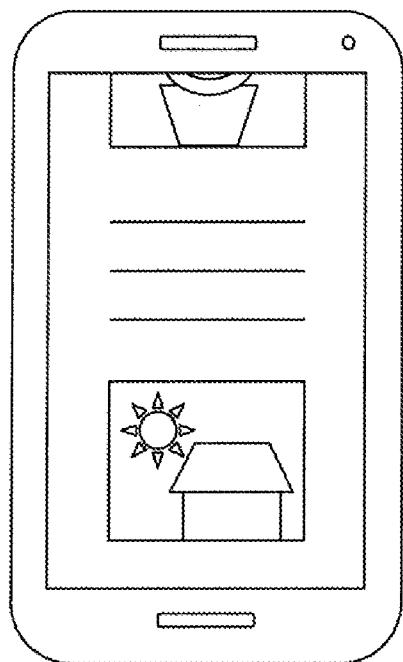


FIG. 27

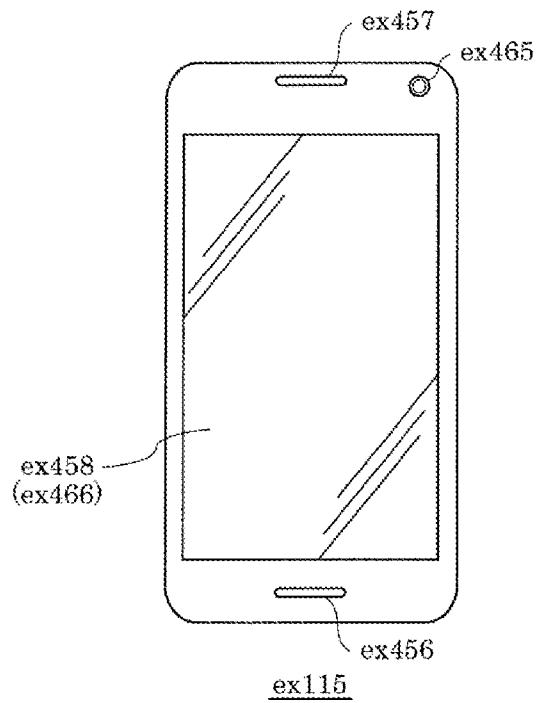


FIG. 28

