

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 776 925**

51 Int. Cl.:

H04N 19/48 (2014.01)

H04N 19/467 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **20.01.2012 E 18159781 (6)**

97 Fecha y número de publicación de la concesión europea: **11.12.2019 EP 3399753**

54 Título: **Ocultación de múltiples bits de signo dentro de una unidad de transformación**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
03.08.2020

73 Titular/es:

**VELOS MEDIA INTERNATIONAL LIMITED
(100.0%)**

**Unit 32, the Hyde Building The Park,
Carrickmines
Dublin 18, IE**

72 Inventor/es:

**HE, DAKE;
WANG, JING y
YU, XIANG**

74 Agente/Representante:

MILTENYI , Peter

ES 2 776 925 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

Ocultación de múltiples bits de signo dentro de una unidad de transformación

5 CAMPO

La presente solicitud se refiere en general a compresión de datos y, en particular, a procedimientos y dispositivos para ocultación de bits de signo cuando se codifican y decodifican datos de vídeo residuales.

10 ANTECEDENTES

10 La compresión de datos se produce en diversos contextos. Se utiliza muy comúnmente en las comunicaciones y en las redes informáticas para almacenar, transmitir y reproducir información de manera eficiente. Encuentra aplicación particular en la codificación de imágenes, audio y vídeo. El vídeo representa un reto importante para la compresión de datos debido a la gran cantidad de datos que se requieren para cada fotograma de vídeo y la velocidad con la que a menudo es necesario codificar y decodificar. El estado actual de la técnica para la codificación de vídeo es el estándar de codificación de vídeo ITU-T H.264/AVC. Define un número de perfiles diferentes para diferentes aplicaciones, que incluyen el perfil principal, el perfil de línea base y otros. Actualmente se está desarrollando un estándar de codificación de vídeo de próxima generación a través de una iniciativa conjunta del estándar MPEG-ITU que se denomina Codificación de Vídeo de Alta Eficiencia (HEVC: High Efficiency Video Coding). La iniciativa puede eventualmente dar como resultado un estándar de codificación de vídeo comúnmente conocido como estándar MPEG-H.

Existen diversos estándares para la codificación/decodificación de imágenes y vídeos, que incluyen el estándar H.264, que utilizan procesos de codificación basados en bloques. En estos procesos, la imagen o fotograma se divide en bloques, normalmente de 4 x 4 o de 8 x 8, y los bloques se transforman espectralmente en coeficientes, se cuantifican y se codifican por entropía. En muchos casos, los datos que se transforman no son datos de los píxeles reales, sino que son datos residuales después de una operación de predicción. Las predicciones pueden ser intra fotograma, es decir, bloque a bloque dentro del fotograma/imagen, o inter fotograma, es decir, entre fotogramas (también denominada predicción de movimiento). Se espera que el estándar MPEG-H también tenga estas características.

30 Cuando se transforman espectralmente los datos residuales, muchos de estos estándares prescriben el uso de una transformación de coseno discreta (DCT: discrete cosine transform) o alguna variante de la misma. Los coeficientes de la DCT resultantes son entonces cuantificados usando un cuantificador para producir coeficientes de dominio de transformación cuantificados, o índices.

35 El bloque o matriz de coeficientes de dominio de transformación cuantificados (a veces denominados "unidad de transformación") es codificado entonces por entropía utilizando un modelo de contexto particular. En el estándar H.264/AVC y en los actuales trabajos de desarrollo del estándar MPEG-H, los coeficientes de transformación cuantificados son codificados mediante (a) codificación de una posición de último coeficiente significativo que indica la ubicación del último coeficiente distinto de cero en la unidad de transformación, (b) codificación de un mapa de significación que indica las posiciones en la unidad de transformación (distintas de la posición de último coeficiente significativo) que contienen coeficientes distintos de cero, (c) codificación de las magnitudes de los coeficientes distintos de cero y (d) codificación de los signos de los coeficientes distintos de cero. Esta codificación de los coeficientes de transformación cuantificados suele ocupar entre el 30 y el 80% de los datos codificados en el flujo de bits.

50 Las unidades de transformación son normalmente de N x N. Los tamaños comunes incluyen 4 x 4, 8 x 8, 16 x 16 y 32 x 32, aunque son posibles otros tamaños, que incluyen tamaños no cuadrados en algunas formas de realización, tales como 8 x 32 o 32 x 8. El signo de cada coeficiente distinto de cero en un bloque se codifica utilizando un bit de signo para cada coeficiente distinto de cero.

55 JEAN-MARC THIESSE ET AL, "Data hiding of motion information in chroma and luma samples for video compression", 2010 IEEE INTERNATIONAL WORKSHOP ON MULTIMEDIA SIGNAL PROCESSING (MMSP '10), SAINT MALO, FRANCIA, 4 - 6 OCT. 2010, IEEE, IEEE, PISCATAWAY, USA, (20101004), ISBN 978-1-4244-8110-1, páginas 217 - 221, propone reducir la información de señalización resultante de esta competencia de vectores, utilizando técnicas de ocultación de datos. Dado que la ocultación de datos y la compresión de vídeo tienen tradicionalmente objetivos contradictorios, se realiza primero un estudio de la ocultación de datos. Entonces, se propone una forma eficiente de utilizar la ocultación de datos para la compresión de vídeo. La idea principal es ocultar los índices en coeficientes de transformación de croma y de luma debidamente seleccionados. Para minimizar los errores de predicción, la modificación se realiza a través de una optimización de tasa-distorsión. Las mejoras objetivas (hasta un 2,3% de ahorro en tasa de bits) y la evaluación subjetiva de la pérdida de croma son reportadas y analizadas para diversas secuencias.

65 BREVE DESCRIPCIÓN DE LOS DIBUJOS

A continuación se hará referencia, a modo de ejemplo, a los dibujos adjuntos, en los que se muestran unas formas de realización de ejemplo de la presente solicitud:

La Figura 1 muestra, en forma de diagrama de bloques, un codificador para la codificación de vídeo;

La Figura 2 muestra, en forma de diagrama de bloques, un decodificador para la decodificación de vídeo;

La Figura 3 muestra un ejemplo de un orden de exploración multi nivel para una unidad de transformación de 16 x 16;

La Figura 4 muestra un ejemplo de unidad de transformación de 16 x 16 dividida en grupos de coeficientes numerados en un orden inverso de exploración a nivel de grupo;

La Figura 5 muestra un ejemplo de una unidad de transformación en la que se forman cuatro grupos de grupos de coeficientes para ocultación de bits de signo;

La Figura 6 ilustra otro ejemplo de una agrupación de grupos de coeficientes para ocultación de bits de signo;

La Figura 7 muestra otro ejemplo más de una agrupación de grupos de coeficientes para ocultación de bits de signo;

La Figura 8 ilustra un ejemplo de formación dinámica de conjuntos de coeficientes para ocultación de bits de signo;

La Figura 9 muestra, en forma de diagrama de flujo, un ejemplo de proceso para ocultación de bits de signo;

La Figura 10 muestra un diagrama de bloques simplificado de un ejemplo de forma de realización de un codificador; y

La Figura 11 muestra un diagrama de bloques simplificado de un ejemplo de forma de realización de un decodificador.

Es posible que se hayan utilizado números de referencia similares en diferentes figuras para denotar componentes similares.

DESCRIPCIÓN DE EJEMPLOS DE FORMAS DE REALIZACIÓN

La presente solicitud describe procedimientos y codificadores/decodificadores para codificar y decodificar datos de vídeo residuales utilizando ocultación de bits de signo. La invención se define en el conjunto de reivindicaciones anexas. Los demás ejemplos denominados formas de realización en la descripción son ejemplos ilustrativos. En algunas de las formas de realización, el codificador y el decodificador pueden utilizar mapas de significación multi nivel para codificar indicadores de coeficientes significativos. Un bit de signo de al menos un coeficiente para cada subconjunto de los coeficientes en la unidad de transformación se puede ocultar usando una técnica de paridad. En algunos casos, los subconjuntos de coeficientes corresponden a los grupos de coeficientes utilizados en los mapas multi nivel que se utilizan, por ejemplo, en la codificación y decodificación de los mapas de significación. En al menos un caso, los mapas multi nivel se utilizan con unidades de transformación más grandes, tales como las unidades de 16 x 16 y de 32 x 32. En algunos casos, los mapas multi nivel se utilizan con unidades de transformación de 8 x 8, unidades de transformación que no son cuadradas y otras unidades de transformación de tamaño diferente. La técnica de ocultación de bits de signo se puede utilizar para aquellos subconjuntos de coeficientes que contengan más de un número umbral de coeficientes distintos de cero. En algunas formas de realización, la técnica de ocultación de bits de signo basada en subconjuntos también se puede utilizar con unidades de transformación aunque no utilicen una codificación de mapas de significación multi nivel, particularmente si la codificación de coeficientes significativos de la unidad de transformación se implementa de forma modular para subconjuntos de los indicadores de coeficientes significativos.

En un aspecto, la presente solicitud describe un procedimiento de decodificación de un flujo de bits de vídeo codificado reconstruyendo coeficientes para una unidad de transformación, codificando el flujo de bits dos o más conjuntos de bits de signo para la unidad de transformación, correspondiendo cada conjunto a un respectivo conjunto de coeficientes sin solapamientos para la unidad de transformación, en el que cada bit de signo indica el signo de un correspondiente coeficiente distinto de cero dentro del respectivo conjunto. El procedimiento incluye, para cada uno de los dos o más conjuntos de bits de signo, sumar un valor absoluto de los coeficientes para el respectivo conjunto que corresponde a ese conjunto de bits de signo para obtener un valor de paridad; y asignar un signo a uno de los coeficientes dentro del respectivo conjunto en base a si el valor de paridad es par o impar.

En otro aspecto, la presente solicitud describe un procedimiento de codificación de un flujo de bits de vídeo codificando bits de signo para coeficientes de una unidad de transformación. El procedimiento incluye, para cada uno de dos o más conjuntos de coeficientes sin solapamientos para la unidad de transformación,

sumar un valor absoluto de los coeficientes de ese conjunto para obtener un valor de paridad; determinar que un signo de uno de los coeficientes en ese conjunto no corresponde al valor de paridad; y ajustar un nivel de un coeficiente en ese conjunto en uno para cambiar el valor de paridad para que corresponda al signo de uno de los coeficientes.

5

En otro aspecto, la presente solicitud describe codificadores y decodificadores configurados para implementar dichos procedimientos de codificación y decodificación.

10

En otro aspecto más, la presente solicitud describe medios no transitorios legibles por ordenador que almacenan instrucciones de programa ejecutables por ordenador que, cuando son ejecutadas, configuran un procesador para realizar los procedimientos de codificación y/o decodificación que se describen.

15

Otros aspectos y características de la presente solicitud serán entendidos por parte de los expertos en la materia a partir de una revisión de la siguiente descripción de ejemplos juntamente con las figuras adjuntas.

20

En la siguiente descripción se describen algunos ejemplos de formas de realización con referencia al estándar H.264 para la codificación de vídeo y/o el estándar MPEG-H que está en desarrollo. Los expertos en la materia entenderán que la presente solicitud no está limitada al estándar H.264/AVC o al estándar MPEG-H, sino que puede ser aplicable a otros estándares de codificación/decodificación de vídeo, que incluyen posibles estándares futuros, estándares de codificación multi vista, estándares de codificación de vídeo escalables y estándares de codificación de vídeo reconfigurables.

25

En la siguiente descripción, cuando se hace referencia a un vídeo o a imágenes, los términos fotograma (frame), imagen, corte (slice), pieza de mosaico (tile) y grupo de piezas de mosaico con forma rectangular se pueden utilizar de forma intercambiable. Los expertos en la materia apreciarán que, en el caso del estándar H.264, un fotograma puede contener uno o más cortes. También se apreciará que ciertas operaciones de codificación/decodificación se realizan fotograma a fotograma, algunas se realizan corte a corte, algunas imagen a imagen, algunas pieza de mosaico a pieza de mosaico y algunas por grupos de cortes con forma rectangular, en función de los requisitos particulares o de la terminología del estándar de codificación de imágenes o de vídeo aplicable. En cualquier forma de realización particular, el estándar de codificación de imágenes o de vídeo aplicable puede determinar si las operaciones que se describen a continuación se realizan en relación con fotogramas y/o cortes y/o imágenes y/o piezas de mosaico y/o grupos de cortes con forma rectangular, según sea el caso. Por consiguiente, los expertos en la materia entenderán, a la luz de la presente divulgación, si las operaciones o procesos particulares que se describen en el presente documento y las referencias particulares a fotogramas, cortes, fotogramas, piezas de mosaico, grupos de cortes con forma rectangular son aplicables a fotogramas, cortes, imágenes, piezas de mosaico, grupos de cortes con forma rectangular, o algunos o todos los de una determinada forma de realización. Esto también se aplica a unidades de transformación, unidades de codificación, grupos de unidades de codificación, etc. según se entenderá a la luz de la siguiente descripción.

30

35

La presente solicitud describe ejemplos de procesos y dispositivos para codificar y decodificar bits de signo para los coeficientes distintos de cero de una unidad de transformación. Los coeficientes distintos de cero se identifican mediante un mapa de significación. Un mapa de significación es un bloque, matriz, grupo o conjunto de indicadores que se mapea con o corresponde a una unidad de transformación o a una unidad definida de coeficientes (por ejemplo, diversas unidades de transformación, una parte de una unidad de transformación o una unidad de codificación). Cada indicador indica si la correspondiente posición en la unidad de transformación o en la unidad especificada contiene o no contiene un coeficiente distinto de cero. En los estándares existentes, estos indicadores se pueden denominar indicadores de coeficientes significativos. En los estándares existentes, hay un indicador por cada coeficiente desde el coeficiente de DC hasta el último coeficiente significativo en un orden de exploración, y el indicador es un bit que se establece igual a cero si el correspondiente coeficiente es igual a cero y se establece igual a uno si el correspondiente coeficiente es distinto de cero. El término "mapa de significancia" tal como se utiliza en el presente documento pretende referirse a una matriz o conjunto ordenado de indicadores de coeficientes significativos para una unidad de transformación, según se entenderá a partir de la siguiente descripción, o a una unidad definida de coeficientes, lo que quedará claro a partir del contexto de las aplicaciones.

40

45

Se entenderá, a la luz de la siguiente descripción, que la estructura de codificación y decodificación multi nivel se puede aplicar en ciertas situaciones, y esas situaciones se pueden determinar a partir de información complementaria como el tipo de contenido de vídeo (vídeo natural o gráficos según se identifica en las cabeceras de secuencia, imagen o corte). Por ejemplo, se pueden utilizar dos niveles para el vídeo natural y se pueden utilizar tres niveles para los gráficos (que normalmente son mucho más dispersos). Otra posibilidad es proporcionar un indicador en una de las cabeceras de secuencia, imagen o corte para indicar si la estructura tiene uno, dos o tres niveles, permitiendo con ello al codificador la flexibilidad de elegir la estructura más apropiada para el contenido actual. En otra forma de realización, el indicador puede representar un tipo de contenido, que estaría asociado con el número de niveles. Por ejemplo, un contenido de tipo "gráfico" puede tener tres niveles.

50

55

Ahora se hace referencia a la Figura 1, que muestra, en forma de diagrama de bloques, un codificador 10 para la codificación de vídeo. También se hace referencia a la Figura 2, que muestra un diagrama de bloques de un decodificador 50 para la decodificación de vídeo. Se apreciará que el codificador 10 y el decodificador 50 que se describen en el presente documento pueden ser implementados en un dispositivo informático de aplicación específica o de propósito general, que contenga uno o más elementos de procesamiento y una memoria. Las operaciones realizadas por el codificador 10 o el decodificador 50, según sea el caso, se pueden implementar mediante un circuito integrado específico de aplicación, por ejemplo, o mediante instrucciones de programa almacenadas que son ejecutables por un procesador de propósito general. El dispositivo puede incluir un software adicional, por ejemplo, un sistema operativo para controlar las funciones básicas del dispositivo. La gama de dispositivos y plataformas dentro de las cuales se pueden implementar el codificador 10 o el decodificador 50 será apreciada por parte de los expertos en la materia teniendo en cuenta la siguiente descripción.

El codificador 10 recibe una fuente de vídeo 12 y produce un flujo de bits codificado 14. El decodificador 50 recibe el flujo de bits codificado 14 y produce un fotograma de vídeo decodificado 16. El codificador 10 y el decodificador 50 pueden estar configurados para operar de acuerdo con una serie de estándares de compresión de vídeo. Por ejemplo, el codificador 10 y el decodificador 50 pueden ser compatibles con el estándar H.264/AVC. En otras formas de realización, el codificador 10 y el decodificador 50 pueden cumplir con otros estándares de compresión de vídeo, que incluyen las evoluciones del estándar H.264/AVC, como el estándar MPEG-H.

El codificador 10 incluye un predictor espacial 21, un selector de modo de codificación 20, un procesador de transformación 22, un cuantificador 24 y un codificador de entropía 26. Según apreciarán los expertos en la materia, el selector de modo de codificación 20 determina el modo de codificación apropiado para la fuente de vídeo, por ejemplo si el fotograma/corte en cuestión es de tipo I, P, o B, y si las unidades de codificación particulares (por ejemplo, macro bloques, unidades de codificación, etc.) dentro del fotograma/corte son inter o intra codificadas. El procesador de transformación 22 realiza una transformación sobre los datos de dominio espacial. En particular, el procesador de transformación 22 aplica una transformación basada en bloques para convertir los datos de dominio espacial en componentes espectrales. Por ejemplo, en muchas formas de realización se utiliza una transformación de coseno discreta (DCT: discrete cosine transform). En algunos casos se pueden utilizar otras transformaciones, tal como una transformación de seno discreta u otras. La transformación basada en bloques se realiza en base a una unidad de codificación, macro bloque o sub bloque, dependiendo del tamaño de los macro bloques o unidades de codificación. En el estándar H.264, por ejemplo, un macro bloque típico de 16 x 16 contiene dieciséis bloques de transformación de 4 x 4 y el proceso de DCT se realiza en los bloques de 4 x 4. En algunos casos, los bloques de transformación pueden ser de 8 x 8, lo que significa que hay cuatro bloques de transformación por cada macro bloque. En otros casos, los bloques de transformación pueden ser de otros tamaños. En algunos casos, un macro bloque de 16 x 16 puede incluir una combinación sin solapamientos de bloques de transformación de 4 x 4 y de 8 x 8.

Aplicando la transformación basada en bloques a un bloque de datos de píxeles se obtiene un conjunto de coeficientes de dominio de transformación. Un "conjunto" en este contexto es un conjunto ordenado en el que los coeficientes tienen posiciones de coeficiente. En algunos casos, el conjunto de coeficientes de dominio de transformación se puede considerar como un "bloque" o matriz de coeficientes. En la descripción de la presente descripción, las frases un "conjunto de coeficientes de dominio de transformación" o un "bloque de coeficientes de dominio de transformación" se utilizan indistintamente y pretenden indicar un conjunto ordenado de coeficientes de dominio de transformación.

El conjunto de coeficientes de dominio de transformación es cuantificado por el cuantificador 24. Los coeficientes cuantificados y la información asociada son codificados entonces por el codificador de entropía 26.

El bloque o matriz de coeficientes de dominio de transformación cuantificados se puede denominar en el presente documento "unidad de transformación" (UT). En algunos casos, la unidad de transformación puede ser no cuadrada, por ejemplo, una transformación de cuadratura no cuadrada (NSQT: non-square quadrature transform).

Los fotogramas/cortes intra codificados (es decir, de tipo I) se codifican sin referencia a otros fotogramas/cortes. En otras palabras, no utilizan una predicción temporal. Sin embargo, los fotogramas intra codificados dependen de una predicción espacial dentro del fotograma/corte, según se ilustra en la Figura 1 por el predictor espacial 21. Es decir, cuando se codifica un bloque concreto, los datos del bloque se pueden comparar con los datos de los píxeles cercanos dentro de los bloques ya codificados para ese fotograma o corte. Usando un algoritmo de predicción, se pueden convertir los datos de origen del bloque en datos residuales. El procesador de transformación 22 codifica entonces los datos residuales. El estándar H.264, por ejemplo, prescribe nueve modos de predicción espacial para bloques de transformación de 4 x 4. En algunas formas de realización, cada uno de los nueve modos se puede utilizar para procesar un

bloque de forma independiente y, a continuación, se utiliza una optimización de tasa-distorsión para seleccionar el mejor modo.

5 El estándar H.264 también prescribe el uso de una predicción/compensación de movimiento para obtener ventajas de la predicción temporal. Por consiguiente, el codificador 10 tiene un bucle de retroalimentación que incluye un cuantificador inverso (de-quantizer) 28, un procesador de transformación inversa 30 y un procesador de desbloqueo 32. El procesador de desbloqueo 32 puede incluir un procesador de desbloqueo y un procesador de filtrado. Estos elementos son un reflejo del proceso de decodificación implementado por el decodificador 50 para reproducir el fotograma/corte. Se utiliza un almacén de fotogramas 34 para almacenar los fotogramas reproducidos. De esta manera, la predicción de movimiento se basa en lo que serán los fotogramas reconstruidos en el decodificador 50 y no en los fotogramas originales, que pueden ser diferentes de los fotogramas reconstruidos debido a la compresión con pérdidas involucrada en la codificación/decodificación. Un predictor de movimiento 36 utiliza los fotogramas/cortes almacenados en el almacén de fotogramas 34 como fotogramas/cortes de origen para compararlos con un fotograma actual con el fin de identificar bloques similares. Por consiguiente, para los macro bloques o unidades de codificación a los que se aplica la predicción de movimiento, los "datos de origen" que codifica el procesador de transformación 22 son los datos residuales que produce el proceso de predicción de movimiento. Por ejemplo, puede incluir información sobre el fotograma de referencia, un desplazamiento espacial o un "vector de movimiento", y datos de píxeles residuales que representan las diferencias (si las hay) entre el bloque de referencia y el bloque actual. La información relativa al fotograma de referencia y/o al vector de movimiento puede no ser procesada por el procesador de transformación 22 y/o el cuantificador 24, sino que, en lugar de ello, puede ser suministrada al codificador de entropía 26 para su codificación como parte del flujo de bits junto con los coeficientes cuantificados.

25 Los expertos en la materia apreciarán los detalles y las posibles variaciones para implementar codificadores de vídeo.

30 El decodificador 50 incluye un decodificador de entropía 52, un cuantificador inverso (dequantizer) 54, un procesador de transformación inversa 56, un compensador espacial 57 y un procesador de desbloqueo 60. El procesador de desbloqueo 60 puede incluir procesadores de desbloqueo y de filtrado. Un búfer de fotogramas 58 suministra fotogramas reconstruidos para que los utilice un compensador de movimiento 62 en la aplicación de compensación de movimiento. El compensador espacial 57 representa la operación de recuperación de los datos de vídeo para un bloque intra codificado particular a partir de un bloque decodificado previamente.

35 El flujo de bits 14 es recibido y decodificado por el decodificador de entropía 52 para recuperar los coeficientes cuantificados. También se puede recuperar información complementaria durante el proceso de decodificación de entropía, parte de la cual puede ser suministrada al bucle de compensación de movimiento para su uso en la compensación de movimiento, si procede. Por ejemplo, el decodificador de entropía 52 puede recuperar vectores de movimiento y/o información de fotograma de referencia para macro bloques inter codificados.

40 Los coeficientes cuantificados son entonces de-cuantificados por el cuantificador inverso 54 para producir los coeficientes de dominio de transformación, los cuales son sometidos entonces a una transformación inversa por parte del procesador de transformación inversa 56 para recrear los "datos de vídeo". Se apreciará que, en algunos casos, tales como en el caso de un macro bloque o unidad de codificación intra codificada, los "datos de vídeo" recreados son los datos residuales para uso en la compensación espacial relativa a un bloque decodificado previamente dentro del fotograma. El compensador espacial 57 genera los datos de vídeo a partir de los datos residuales y datos de píxeles de un bloque decodificado previamente. En otros casos, como en el caso de macro bloques o unidades de codificación inter codificadas, los "datos de vídeo" recreados procedentes del procesador de transformación inversa 56 son los datos residuales para uso en la compensación de movimiento relativa a un bloque de referencia de un fotograma diferente. Tanto la compensación espacial como la de movimiento se pueden denominar en el presente documento como "operaciones de predicción".

55 El compensador de movimiento 62 localiza un bloque de referencia dentro del búfer de fotogramas 58 especificado para un macro bloque o unidad de codificación inter codificada en particular. Esto lo realiza en base a la información del fotograma de referencia y del vector de movimiento especificados para el macro bloque o unidad de codificación inter codificada. A continuación, suministra los datos de píxeles del bloque de referencia para su combinación con los datos residuales para llegar a los datos de vídeo reconstruidos para esa unidad de codificación/macro bloque.

60 A continuación se puede aplicar un proceso de desbloqueo/filtrado a un fotograma/corte reconstruido, según lo indicado por el procesador de desbloqueo 60. Después del desbloqueo/filtrado, se produce el fotograma/corte como el fotograma de vídeo decodificado 16, por ejemplo para su visualización en un dispositivo de visualización. Se entenderá que la máquina de reproducción de vídeo, tal como un ordenador,

65

decodificador de contenidos multimedia, reproductor de DVD o Blu-Ray, y/o dispositivo móvil de mano, puede almacenar en un búfer los fotogramas decodificados antes de mostrarlos en un dispositivo de salida.

5 Se espera que los codificadores y decodificadores que cumplen con el estándar MPEG-H tendrán muchas de estas características iguales o similares.

Codificación y Decodificación de Coeficientes de Dominio de Transformación Cuantificados

10 Según se ha observado anteriormente, la codificación de entropía de un bloque o conjunto de coeficientes de dominio de transformación cuantificados incluye la codificación del mapa de significación (por ejemplo, un conjunto de indicadores de coeficientes significativos) para ese bloque o conjunto de coeficientes de dominio de transformación cuantificados. El mapa de significación es un mapeo binario del bloque que indica en qué posiciones (desde la posición de DC hasta la posición del último coeficiente significativo) aparecen coeficientes distintos de cero. El mapa de significación se puede convertir en un vector de acuerdo con el orden de exploración (que puede ser vertical, horizontal, en diagonal, en zig-zag, o cualquier otro orden de exploración prescrito por el estándar de codificación aplicable). La exploración se realiza normalmente en orden "inverso", es decir, comenzando con el último coeficiente significativo y procesando hacia atrás a través del mapa de significación en dirección inversa hasta que se alcanza el indicador de coeficiente significativo en la esquina superior izquierda en [0,0]. En la presente descripción, se pretende que el término "orden de exploración" se refiera al orden en el que se procesan los indicadores, coeficientes o grupos, según sea el caso, y puede incluir órdenes que se denominan coloquialmente "orden de exploración inverso".

25 Entonces, se codifica por entropía cada indicador (flag) de coeficiente significativo usando el esquema de codificación adaptativa al contexto que sea aplicable. Por ejemplo, en muchas aplicaciones se puede utilizar un esquema de codificación aritmética binaria adaptativa al contexto (CABAC: context-adaptive binary arithmetic coding).

30 En mapas de significación de 16 x 16 y de 32 x 32, el contexto para un indicador de coeficiente significativo se basa (en la mayoría de los casos) en valores de indicadores de coeficientes significativos vecinos. Entre los contextos usados para mapas de significación de 16 x 16 y de 32 x 32, hay ciertos contextos dedicados a la posición de bit en [0,0] y (en algunas implementaciones de ejemplo) a posiciones de bits vecinos, pero la mayoría de los indicadores de coeficientes significativos toman uno de los cuatro o cinco contextos que dependen de los valores acumulativos de indicadores de coeficientes significativos vecinos. En estos casos, la determinación del contexto correcto para un indicador de coeficiente significativo depende de la determinación y la suma de los valores de los indicadores de coeficientes significativos en ubicaciones vecinas (normalmente cinco ubicaciones, pero podrían ser más o menos en algunos casos).

40 Entonces, se pueden codificar los niveles de coeficientes significativos para esos coeficientes distintos de cero. En una implementación de ejemplo, los niveles se pueden codificar codificando primero un mapa de los coeficientes distintos de cero que tengan un nivel de valor absoluto superior a uno. Se puede codificar entonces otro mapa de los coeficientes distintos de cero que tengan un nivel superior a dos. Entonces se codifica el valor o nivel de cualquiera de los coeficientes que tengan un valor absoluto superior a dos. En algunos casos, el valor codificado puede ser el valor real menos tres.

45 También se codifica el signo de los coeficientes distintos de cero. Cada coeficiente distinto de cero tiene un bit de signo que indica si el nivel de ese coeficiente distinto de cero es negativo o positivo. Se ha propuesto ocultar el bit de signo para el primer coeficiente en la unidad de transformación: Clare, Gordon, et al., "Sign Data Hiding", JCTVC-G271, 7ª reunión, Ginebra, 21-30 de noviembre de 2011. En esta propuesta, se codifica el signo del primer coeficiente en la unidad de transformación a través de la paridad de la suma de coeficientes cuantificados en la unidad de transformación. En el caso de que la paridad no corresponda al signo real del primer coeficiente, entonces el codificador debe ajustar el nivel de uno de los coeficientes incrementándolo o decrementándolo en 1 para ajustar la paridad. Se debe utilizar la cuantificación optimizada de tasa-distorsión (RDOQ: Rate-Distortion Optimized Quantization) para determinar qué coeficiente se debe ajustar y en qué dirección.

55 Algunos trabajos anteriores se han centrado en el uso de mapas de significación multi nivel. Ahora se hace referencia a la Figura 3, que muestra una unidad de transformación de 16 x 16 100 con un orden de exploración en diagonal multi nivel ilustrado. La unidad de transformación 100 se divide en dieciséis grupos de coeficientes de 4 x 4 o "conjuntos de indicadores de coeficientes significativos" contiguos. Dentro de cada grupo de coeficientes, se aplica un orden de exploración en diagonal dentro del grupo, en lugar de a través de toda la unidad de transformación 100. Los conjuntos o grupos de coeficientes se procesan en un orden de exploración, que en este ejemplo de implementación es también un orden de exploración en diagonal. Se observará que el orden de exploración en este ejemplo se ilustra en orden de exploración "inverso"; es decir, el orden de exploración se muestra progresando desde el grupo de coeficientes inferior derecho en una dirección en diagonal hacia abajo a la izquierda hacia el grupo de coeficientes superior izquierdo. En algunas implementaciones, se puede definir el mismo orden de exploración en la otra

dirección; esto es, progresando en una dirección en diagonal hacia arriba a la derecha y cuando se aplica durante la codificación o decodificación se puede aplicar en un orden de exploración "inverso".

5 El uso de mapas de significación multi nivel implica la codificación de un mapa de significación de nivel L1 o de nivel superior que indica qué grupos de coeficientes se puede esperar que contengan indicadores de coeficientes significativos distintos de cero, y qué grupos de coeficientes contienen todos los indicadores de coeficientes significativos que son cero. Los grupos de coeficientes de los que se puede esperar que contengan indicadores de coeficientes significativos distintos de cero tienen sus indicadores de coeficientes significativos codificados, mientras que los grupos de coeficientes que contienen todos los indicadores de coeficientes significativos que son cero no están codificados (a menos que sean grupos que están codificados debido a una excepción de caso especial porque se presume que contienen al menos un indicador de coeficiente significativo distinto de cero). Cada grupo de coeficientes tiene un indicador de grupo de coeficientes significativos (a menos que se aplique un caso especial en el que ese grupo de coeficientes tenga un indicador de un valor presumido, tal como el grupo que contiene el último coeficiente significativo, el grupo superior izquierdo, etc.).

La utilización de mapas de significación multi nivel facilita el procesamiento modular de datos residuales para la codificación y decodificación.

20 Las unidades de transformación más grandes presentan una oportunidad de ocultar múltiples bits de signo. La unidad de transformación se puede dividir o particionar en conjuntos de coeficientes distintos de cero y se puede ocultar un bit de signo para cada conjunto de coeficientes distintos de cero utilizando la paridad de la suma de los coeficientes distintos de cero en ese conjunto. En una forma de realización, el conjunto de coeficientes distintos de cero se puede construir para que corresponda a los grupos de coeficientes definidos para mapas de significación multi nivel.

30 Se puede utilizar un solo umbral para determinar si se debe ocultar un bit de signo para un determinado conjunto de coeficientes distintos de cero, con independencia del tipo de datos. En un ejemplo, el chequeo de umbral se basa en el número de coeficientes entre el primer coeficiente distinto de cero y el último coeficiente distinto de cero del conjunto. Es decir, si hay al menos un número umbral de coeficientes entre el primer y el último coeficiente distinto de cero en el conjunto. En otro ejemplo, el chequeo se puede basar en que haya al menos un número umbral de coeficientes distintos de cero en el conjunto. En otra forma de realización más, el chequeo se puede basar en la suma del valor absoluto de los coeficientes distintos de cero del conjunto por encima de un umbral. En otra forma de realización más, se puede aplicar una combinación de estas pruebas; es decir, debe haber al menos un número mínimo de coeficientes en el conjunto y el valor absoluto acumulativo de los coeficientes debe superar un valor de umbral. También se pueden utilizar variaciones en estas pruebas de umbral.

40 Ahora se hace referencia a la Figura 4, que muestra un ejemplo de unidad de transformación de 16 x 16 120. La unidad de transformación 120 se divide en grupos de coeficientes de 4 x 4, es decir, dieciséis conjuntos de coeficientes. Los grupos de coeficientes se numeran 1, 2, 3,... 16 en el orden en el que se procesan, por ejemplo, en orden de exploración diagonal inverso.

45 En una primera forma de realización, cada grupo de coeficientes es un conjunto de coeficientes con el fin de ocultar bits de signo. Es decir, cada grupo de coeficientes es chequeado en relación a un umbral para determinar si el grupo de coeficientes es adecuado para ocultación de bits de signo. Según se ha indicado anteriormente, el chequeo puede ser que el grupo de coeficientes contenga al menos un número mínimo de coeficientes entre el primer coeficiente distinto de cero y el último coeficiente distinto de cero dentro de ese grupo de coeficientes.

50 En una segunda forma de realización, los conjuntos de coeficientes para ocultación de bits de signo se forman agrupando grupos de coeficientes. La Figura 5 muestra una unidad de transformación de 16 x 16 140 en la que se ilustra un ejemplo de agrupación de grupos de coeficientes en cuatro conjuntos de coeficientes. En este ejemplo, cada conjunto de coeficientes para ocultación de bits de signo contiene cuatro grupos de coeficientes. Los cuatro grupos de coeficientes en cada conjunto son grupos consecutivos en el orden de exploración. Por ejemplo, el primer conjunto de coeficientes 142 contiene los grupos de coeficientes 16, 15, 14 y 13. El segundo conjunto de coeficientes 144 contiene los grupos de coeficientes 12, 11, 10 y 9. El tercer conjunto de coeficientes 146 contiene los grupos de coeficientes 8, 7, 6 y 5. Finalmente, el cuarto conjunto de coeficientes 148 contiene los grupos de coeficientes 4, 3, 2, y 1. En esta forma de realización, se puede ocultar un bit de signo para cada conjunto de coeficientes. Es decir, se pueden ocultar hasta cuatro bits de signo por cada unidad de transformación 140.

65 Para cada conjunto de coeficientes 142, 144, 146, 148, el número de coeficientes entre el primer y el último coeficiente distinto de cero (o el número de coeficientes distintos de cero, o el valor total acumulativo de esos coeficientes) es chequeado con respecto al umbral para determinar si se debe ocultar un bit de signo para ese conjunto. La paridad de la suma de valores absolutos de esos coeficientes en el conjunto es el

mecanismo a través del cual se oculta el bit de signo. Si la paridad no corresponde al signo que se debe ocultar, entonces se ajusta la paridad ajustando el nivel de uno de los coeficientes del conjunto.

5 La Figura 6 ilustra una tercera forma de realización de conjuntos de coeficientes para ocultación de bits de signo con una unidad de transformación de 16×16 150. En esta forma de realización, los conjuntos se forman otra vez en base a grupos de coeficientes, pero los conjuntos no contienen necesariamente el mismo número de coeficientes o grupos de coeficientes. Por ejemplo, en esta ilustración se definen cinco conjuntos de coeficientes. El primer conjunto 152 contiene los grupos de coeficientes del 1 al 6. El segundo conjunto 154 contiene cuatro grupos de coeficientes: 7, 8, 9 y 10. El tercer conjunto 156 contiene los grupos de coeficientes 11, 12 y 13. El cuarto conjunto 158 contiene los grupos de coeficientes 14 y 15. El quinto conjunto 159 contiene sólo el grupo de coeficientes superior izquierdo 16. Se apreciará que esta forma de realización proporciona unos conjuntos más grandes de coeficientes en las áreas de la unidad de transformación 150 para las que es probable que haya menos coeficientes distintos de cero, y conjuntos más pequeños de coeficientes en las áreas de la unidad de transformación 150 en las que los coeficientes distintos de cero son más comunes. Nótese que las formas de realización anteriores se podrían aplicar a tamaños de unidad de transformación de 32×32 o más grandes, así como a un tamaño de unidad de transformación de 8×8 , siempre y cuando se aplique una estructura de grupo de coeficientes a esas unidades de transformación.

20 La Figura 7 ilustra una cuarta forma de realización, en la que se forman conjuntos de coeficientes para ocultación de bits de signo dentro de una unidad de transformación de 8×8 160 sin satisfacer la estructura de grupos de coeficientes. La unidad de transformación de 8×8 puede o no tener división de grupos de coeficientes con el fin de codificar el mapa de significancia. En cualquier caso, en esta forma de realización se utiliza una exploración en diagonal basada en unidades de transformación para procesar los coeficientes para codificación y ocultación de bits de signo. En este caso, los conjuntos de coeficientes se forman para agrupar coeficientes consecutivos en el orden de exploración. Por ejemplo, en esta ilustración la unidad de transformación 160 se agrupa en cuatro conjuntos de coeficientes, cada uno de los cuales contiene 16 coeficientes consecutivos en el orden de exploración. Los grupos se etiquetan con 162, 164, 166 y 168 en la Figura 7.

30 En otra forma de realización, los conjuntos de coeficientes pueden no satisfacer el orden de exploración. Es decir, cada conjunto puede incluir algunos coeficientes de posiciones con mayor frecuencia en la unidad de transformación y algunos coeficientes de posiciones con menor frecuencia en la unidad de transformación. Todos los coeficientes de estos conjuntos no tienen por qué ser adyacentes en el orden de exploración.

40 La Figura 8 muestra una quinta forma de realización, en la que los conjuntos de coeficientes para ocultación de bits de signo dentro de una unidad de transformación de 16×16 170 se forman dinámicamente, usando la estructura de grupos de coeficientes y el orden de exploración. En esta forma de realización, en lugar de un conjunto fijo de coeficientes predefinidos en base al tamaño de la unidad de transformación y al orden de exploración, el codificador y el decodificador forman los conjuntos siguiendo el orden de exploración y rastreando cualquier cantidad que se mida en relación a un valor de umbral hasta que se alcance el valor de umbral. Una vez que se ha alcanzado el umbral, se oculta un bit de signo para el grupo de coeficientes que el codificador o decodificador está procesando.

45 Como ejemplo, la Figura 8 ilustra un último coeficiente significativo dentro del grupo de coeficientes [2, 2]. En el orden de exploración, el codificador y el decodificador se mueven entonces a los grupos de coeficientes [1, 3], [3, 0], y [2, 1]. Mientras se procesan los coeficientes en el grupo de coeficientes [2, 1], se alcanza el umbral. Por consiguiente, se oculta el bit de signo para el último coeficiente distinto de cero a procesar en orden de exploración inverso en el grupo de coeficientes [2, 1] (el coeficiente distinto de cero superior izquierdo en el grupo) en la paridad del valor absoluto acumulativo de los coeficientes desde el último coeficiente significativo hasta e incluyendo todos los coeficientes del grupo de coeficientes actual [2, 1]. El chequeo de umbral en este ejemplo se puede basar en la existencia de un número mínimo de coeficientes distintos de cero, o en el valor absoluto de los coeficientes que superen algún valor de umbral. El número de referencia 174 indica una operación de ocultación de bits de signo con respecto al 'último' coeficiente o coeficiente superior izquierdo de un grupo de coeficientes en particular.

60 En una sexta forma de realización, la ocultación de bits de signo se realiza en base a grupos de coeficientes, y los criterios utilizados para determinar si un grupo de coeficientes es adecuado para ocultación de bits de signo son ajustados dinámicamente de acuerdo con los grupos de coeficientes decodificados previamente. Como ejemplo, si el grupo de coeficientes que está inmediatamente a su derecha o el grupo de coeficientes que está inmediatamente debajo tiene un coeficiente distinto de cero, se determina que el grupo de coeficientes actual es adecuado para ocultación de bits de signo siempre que contenga un mínimo de dos coeficientes distintos de cero. El grupo de coeficientes también se puede determinar que es adecuado si contiene al menos un número mínimo de coeficientes entre el primer coeficiente distinto de cero y el último coeficiente distinto de cero dentro de ese grupo de coeficientes, según se describe en algunas de las formas de realización anteriores.

Se apreciará que en algunas de las formas de realización anteriores, un bit de signo se puede ocultar en un grupo de coeficientes en base a un valor de paridad que depende de coeficientes de otro grupo de coeficientes. En otras palabras, el valor de signo de un coeficiente en un grupo de coeficientes se puede ocultar en la paridad por medio de un cambio de nivel a un coeficiente en otro grupo de coeficientes.

Además, se apreciará que en algunas de las formas de realización anteriores, un bit de signo ocultado en un conjunto de coeficientes puede proceder de un elemento de sintaxis diferente como un indicador de diferencia en vector de movimiento (por ejemplo, el indicador `mvd_sign_flag`).

En el lado del codificador se decide qué coeficiente ajustar para ocultar un bit de signo en caso de que el valor de paridad no corresponda al signo. En caso de que se necesite ajustar el valor de paridad, se debe incrementar o decrementar un nivel de coeficientes en 1 para cambiar la paridad.

En una forma de realización, la primera etapa en el proceso de ajuste de un nivel de coeficiente es determinar un rango de búsqueda, es decir, una posición inicial y una posición final en el orden de exploración. Los coeficientes dentro de este rango son evaluados y se selecciona uno para su modificación. En una forma de realización de ejemplo, el rango de búsqueda puede ser desde el primer coeficiente distinto de cero hasta el último coeficiente en el orden de exploración.

Con el uso de mapas de significación multi nivel, se puede cambiar la posición final de un rango de búsqueda para un subconjunto para utilizar la información a nivel de bloque. En concreto, si un subconjunto contiene el último coeficiente distinto de cero en toda la unidad de transformación, (el denominado último coeficiente significativo o último coeficiente significativo global), el rango de búsqueda se puede establecer como el primer coeficiente distinto de cero hasta el último coeficiente distinto de cero. Para otros subconjuntos, se puede extender el rango de búsqueda a un rango desde el primer coeficiente distinto de cero hasta el final del sub bloque actual.

En una forma de realización, se puede extender la posición inicial para incluir condicionalmente coeficientes no cuantificados por delante del primer coeficiente distinto de cero cuantificado. En particular, considerar todos los coeficientes antes de la cuantificación. Los coeficientes no cuantificados que tengan el mismo signo que el signo a ocultar serán incluidos en la búsqueda. Para los coeficientes no cuantificados desde la posición cero hasta la posición del primer coeficiente distinto de cero cuantificado, se evaluará y chequeará en la búsqueda el coste de cambiar un coeficiente cuantificado de cero a uno.

Otra cuestión en el proceso de ajuste de un nivel de coeficiente es definir el cálculo de coste utilizado para evaluar el impacto de un ajuste. Cuando la complejidad de cálculo es una preocupación, el coste se puede basar en la distorsión y se puede no tener en cuenta la tasa, en cuyo caso la búsqueda es para minimizar la distorsión. Por otra parte, cuando la complejidad de cálculo no es una preocupación primordial, el coste puede incluir tanto la tasa como la distorsión para minimizar el coste de tasa-distorsión.

Si la cuantificación optimizada de tasa-distorsión (RDOQ: rate-distortion optimized quantization) está habilitada, entonces se puede utilizar la RDOQ para ajustar el nivel. Sin embargo, en muchos casos, la complejidad de cálculo de la RDOQ puede ser indeseable y la RDOQ puede no estar habilitada. Por consiguiente, en algunas formas de realización se puede aplicar un análisis simplificado de tasa-distorsión en el codificador para implementar ocultación de bit de signo.

Cada coeficiente entre el primer coeficiente distinto de cero del conjunto y el último coeficiente distinto de cero del conjunto puede ser chequeado calculando aproximadamente la distorsión que resulta de incrementar el coeficiente en 1 y decrementar el coeficiente en 1. En términos generales, el valor de coeficiente de u tiene un valor real de $u + \delta$. La distorsión viene dada por $(\delta q)^2$. Si ese coeficiente u es ajustado incrementándolo en 1 hasta $u + 1$, entonces se puede estimar la distorsión resultante de la siguiente manera:

$$q^2 (1 - 2\delta)$$

Si el coeficiente u es ajustado decrementándolo en 1 hasta $u - 1$, entonces se puede estimar la distorsión resultante de la siguiente manera:

$$q^2 (1 + 2\delta)$$

Se reconocerá que para el caso inter codificado, la distorsión de cuantificación δ está en el rango $[-1/6$ a $+5/6]$ cuando la RDOQ está desactivada. En el caso de bloques intra codificados, la distorsión de cuantificación δ está en el rango $[-1/3$ a $+2/3]$ cuando la RDOQ está desactivada. Cuando la RDOQ está activada, el rango de δ variará. Sin embargo, el cálculo anterior del incremento de la distorsión sigue siendo válido, con independencia del rango de δ .

El codificador también puede hacer estimaciones aproximadas del coste de tasa para los distintos

ES 2 776 925 T3

coeficientes utilizando un conjunto de reglas lógicas, es decir, una métrica de coste de tasa predefinida. Por ejemplo, la métrica de coste de tasa predefinida, en una forma de realización, puede incluir:

$u+1$ ($u \neq 0$ y $u \neq -1$) \rightarrow 0,5 bits

$u-1$ ($u \neq 0$ y $u \neq +1$) \rightarrow -0,5 bits

5 $u = 1$ o -1 y cambiado a 0 \rightarrow -1-0,5-0,5 bits

$u=0$ y cambiado a 1 o -1 \rightarrow 1+0,5+0,5 bits

en el que el coste de un indicador de signo se estima que es 1 bit, el coste de un indicador de coeficiente significativo se estima que es 0,5 bits, y el incremento del coste desde u hasta $u+1$ se estima que es 0,5 bits.

10

Se pueden utilizar otras reglas o estimaciones en otras formas de realización.

Ahora se hace referencia a la Figura 9, que muestra un ejemplo de proceso 200 para decodificar datos de vídeo con ocultación de bits de signo basada en grupos de coeficientes. El proceso 200 se basa en la segunda forma de realización descrita anteriormente. Después de revisar la descripción, los expertos en la materia apreciarán alteraciones y modificaciones del proceso 200 para implementar otras formas de realización descritas.

15

En la operación 202 se establece un valor de umbral. En algunas formas de realización, este valor de umbral puede ser predeterminado o preconfigurado dentro del decodificador. En otras formas de realización, este valor se puede extraer del flujo de bits de los datos de vídeo codificados. Por ejemplo, el valor de umbral puede estar en la cabecera de imagen o en otra ubicación dentro del flujo de bits.

20

En la operación 204, el decodificador identifica la primera posición distinta de cero en el grupo de coeficientes actual, es decir, el conjunto de coeficientes, y la última posición distinta de cero en el grupo de coeficientes actual, en orden de exploración. A continuación, determina el número de coeficientes, en orden de exploración, entre el primer y el último coeficiente distinto de cero en el grupo de coeficientes.

25

En la operación 206, el decodificador decodifica bits de signo del flujo de bits. Decodifica un bit de signo para cada coeficiente distinto de cero en el grupo de coeficientes, excepto para el coeficiente distinto de cero superior de más a la izquierda del grupo de coeficientes (el último coeficiente distinto de cero en orden de exploración inverso). Los bits de signo son aplicados a sus respectivos coeficientes distintos de cero. Por ejemplo, si la convención aplicable es que un bit de signo igual a cero es positivo y un bit de signo igual a uno es negativo, entonces se establece para todos los bits de signo iguales a uno que el correspondiente nivel de coeficiente es negativo.

30

35

En la operación 208, el decodificador evalúa si el número de coeficientes entre el primer coeficiente distinto de cero y el último coeficiente distinto de cero en orden de exploración en el grupo de coeficientes supera el umbral. Si no, entonces no se ha utilizado ocultación de bits de signo en el codificador, por lo que en la operación 210 el decodificador decodifica el bit de signo para el coeficiente distinto de cero superior de más a la izquierda (último en orden de exploración inverso) y lo aplica al nivel de coeficiente. Si el número de coeficientes alcanza el umbral, entonces en la operación 212 el decodificador evalúa si el valor absoluto de la suma de los coeficientes del grupo de coeficientes es par o impar, es decir, su paridad. Si es par, entonces el signo del coeficiente distinto de cero superior de más a la izquierda es positivo y el decodificador no necesita ajustarlo. Si es impar, entonces el signo del coeficiente distinto de cero superior de más a la izquierda es negativo, por lo que en la operación 214 hace que ese coeficiente sea negativo.

40

45

En la operación 216, el decodificador determina si ha terminado de procesar los grupos de coeficientes. Si es así, el proceso 200 termina. De lo contrario, procede al siguiente grupo de coeficientes en el orden de exploración de grupos en la operación 218 y vuelve a la operación 204.

50

En otra forma de realización, el tamaño del conjunto de coeficientes se puede reducir a un solo coeficiente. Es decir, la ocultación de bits de signo puede ser una ocultación de signo basada en un solo coeficiente. En esta forma de realización, se chequea cada coeficiente para ver si se debe ocultar la información de signo. Un ejemplo de chequeo es comparar la magnitud del nivel del coeficiente con un umbral determinado. Los coeficientes que tienen un nivel superior al umbral tienen sus bits de signo ocultos; de lo contrario, se utiliza una codificación/decodificación de bits de signo convencional.

55

Para aplicar la ocultación de bits de signo en el caso de un solo coeficiente, la información de signo se compara con la paridad del nivel de coeficiente. Por ejemplo, una paridad par puede corresponder a un signo positivo y una paridad impar puede corresponder a un signo negativo. El codificador ajusta entonces el nivel si el nivel no corresponde al signo. Se apreciará que esta técnica implica que por encima del valor del umbral todos los niveles negativos son impares y todos los niveles positivos son pares. En cierto sentido, esto se puede considerar, en efecto, una modificación del tamaño de etapa de cuantificación para los coeficientes que tienen una magnitud superior al umbral.

60

65

A continuación se presenta un ejemplo de sintaxis para implementar ocultación de bits de signo. La sintaxis

de este ejemplo no es más que una posible implementación. En este ejemplo, la ocultación de bits de signo se aplica en base a grupos de coeficientes, y el chequeo de umbral se basa en los coeficientes numéricos desde el primer coeficiente distinto de cero en el grupo de coeficientes hasta el último coeficiente distinto de cero en el grupo de coeficientes. En la cabecera de la imagen se envía un indicador denominado **sign_data_hiding** para indicar si está activada la ocultación de bits de signo. Si está habilitada, entonces la cabecera también contiene el parámetro **tsig**, que es el valor de umbral. A continuación se presenta un ejemplo de sintaxis:

5

pic_parameter_set_rbsp() {	Descriptor
pic parameter set id	ue(v)
seq parameter set id	ue(v)
sign data hiding	u(1)
si(sign_data_hiding) {	
Tsig	u(4)
}	
entropy coding synchro	u(v)
cabac istate reset flag	u(1)
si(entropy_coding_synchro)	
num_substreams_minus1	ue(v)
...	

10

El siguiente pseudocódigo ilustra un ejemplo de implementación de ocultación de bits de signo basada en grupos de coeficientes:

codificación_residual_cabac(x0, y0, log2TrafoWidth, log2TrafoHeight, scanIdx, cIdx) {	Descriptor
last significant coeff x prefix	ae(v)
last significant coeff y prefix	ae(v)
si(last significant coeff x prefix > 3)	
last significant coeff x suffix	ae(v)
si(last significant coeff y prefix > 3)	
last significant coeff y suffix	ae(v)
numCoeff = 0	
hacer {	
xC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][numCoeff][0]	
yC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][numCoeff][1]	
numCoeff++	
} mientras((xC != LastSignificantCoeffX) (yC != LastSignificantCoeffY))	
numLastSubset = (numCoeff - 1) >> 4	
para(i = numLastSubset; i >= 0; i--) {	
offset = i << 4	
si(max(log2TrafoWidth, log2TrafoHeight) > 3) {	
..... omitida	
} sino {	
..... omitida	
}	
firstNZPosInCG = 1 << 4 ;	
lastNZPosInCG = - 1 ;	
para(n = 15; n >= 0; n--) {	
xC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][n + offset][0]	
yC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][n + offset][1]	
si(significant_coeff_flag[xC][yC]) {	
coeff abs level greater1 flag[n]	ae(v)
si(lastNZPosInCG == -1)	
lastNZPosInCG = n	
firstNZPosInCG = n	
}	
}	
signHidden = lastNZPosInCG - firstNZPosInCG + 1 >= tsig	
para(n = 15; n >= 0; n--) {	
si(coeff_abs_level_greater1_flag[n])	
coeff abs level greater2 flag[n]	ae(v)
}	
para(n = 15; n >= 0; n--) {	
xC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][n + offset][0]	

yC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][n + offset][1]	
si(significant_coeff_flag[xC][yC]) {	
si(!sign_data_hiding !signHidden n != firstNZPosInCG)	
coeff_sign_flag[n]	ae(v)
sino	
coeff_sign_flag[n] = 0 ;	
}	
}	
sumAbs = 0 ;	
para(n = 15; n >= firstNZPosInCG; n--) {	
si(coeff_abs_level_greater2_flag[n])	
coeff_abs_level_minus3[n]	ae(v)
xC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][n + offset][0]	
yC = ScanOrder[log2TrafoWidth][log2TrafoHeight][scanIdx][n + offset][1]	
si(significant_coeff_flag[xC][yC]) {	
transCoeffLevel[x0][y0][cIdx][xC][yC] =	
(coeff_abs_level_minus3[n] + 3) * (1 - 2 * coeff_sign_flag[n])	
si(sign_data_hiding && signHidden)	
sumAbs += coeff_abs_level_minus3[n] + 3	
} sino	
transCoeffLevel[x0][y][cIdx][xC][yC] = 0	
}	
si(sign_data_hiding && signHidden && (sumAbs%2 == 1))	
transCoeffLevel[x0][y0][cIdx][xC][yC] = - transCoeffLevel[x0][y0][cIdx][xC][yC]	
}	
}	

Ahora se hace referencia a la Figura 10, que muestra un diagrama de bloques simplificado de un ejemplo de forma de realización de un codificador 900. El codificador 900 incluye un procesador 902, una memoria 904 y una aplicación de codificación 906. La aplicación de codificación 906 puede incluir un programa informático o una aplicación informática almacenada en la memoria 904 y que contiene instrucciones para configurar el procesador 902 para realizar operaciones como las que se describen en el presente documento. Por ejemplo, la aplicación de codificación 906 puede codificar y suministrar flujos de bits codificados de acuerdo con los procesos que se describen en el presente documento. Se entenderá que la aplicación de codificación 906 se puede almacenar en un medio legible por ordenador, tal como un disco compacto, un dispositivo de memoria flash, una memoria de acceso aleatorio, un disco duro, etc.

Ahora también se hace referencia a la Figura 11, que muestra un diagrama de bloques simplificado de un ejemplo de forma de realización de un decodificador 1000. El decodificador 1000 incluye un procesador 1002, una memoria 1004 y una aplicación de decodificación 1006. La aplicación de decodificación 1006 puede incluir un programa informático o una aplicación informática almacenada en la memoria 1004 y que contiene instrucciones para configurar el procesador 1002 para realizar operaciones como las que se describen en el presente documento. La aplicación de decodificación 1006 puede incluir un decodificador de entropía configurado para reconstruir datos residuales basándose en, al menos en parte, reconstruir indicadores de coeficientes significativos, según se describe en el presente documento. Se entenderá que la aplicación de decodificación 1006 se puede almacenar en un medio legible por ordenador, tal como un disco compacto, un dispositivo de memoria flash, una memoria de acceso aleatorio, un disco duro, etc.

Se apreciará que el decodificador y/o codificador según la presente solicitud se puede implementar en un número de dispositivos informáticos, incluyendo, sin limitación, servidores, ordenadores de propósito general programados de forma adecuada, dispositivos de codificación y reproducción de audio/vídeo, decodificadores de televisión, equipos de difusión de televisión y dispositivos móviles. El decodificador o codificador puede ser implementado por un software que contiene instrucciones para configurar un procesador para llevar a cabo las funciones descritas en el presente documento. Las instrucciones de software se pueden almacenar en cualquier memoria no transitoria adecuada y legible por ordenador, incluyendo CD, RAM, ROM, memoria Flash, etc.

Se entenderá que el codificador descrito en el presente documento y el módulo, rutina, proceso, hilo u otro componente de software que implemente el procedimiento/proceso que se describe para configurar el codificador se puede realizar utilizando técnicas informáticas y lenguajes de programación informáticos estándar. La presente solicitud no se limita a procesadores particulares, lenguajes informáticos, convenciones de programación informáticas, estructuras de datos y otros detalles de implementación de este tipo. Los expertos en la materia reconocerán que los procesos que se describen pueden ser implementados como parte de un código informático ejecutable almacenado en una memoria volátil o no

ES 2 776 925 T3

volátil, como parte de un chip integrado específico de aplicación (ASIC: Application-Specific Integrated Chip), etc.

- 5 Se pueden hacer ciertas adaptaciones y modificaciones de las formas de realización que se describen. Por lo tanto, se considera que las formas de realización mencionadas anteriormente son ilustrativas y no restrictivas.

REIVINDICACIONES

- 5 1. Un procedimiento de decodificar un flujo de bits de vídeo codificado reconstruyendo coeficientes para una unidad de transformación, en el que la unidad de transformación se divide en grupos de coeficientes sin solapamientos, cada uno de los cuales contiene un respectivo conjunto de coeficientes, en el que cada uno de los respectivos conjuntos de coeficientes comprende dos o más coeficientes, comprendiendo el procedimiento: determinar, para cada grupo de coeficientes en la unidad de transformación, si un respectivo signo ha sido codificado en el grupo de coeficientes utilizando ocultación de bits de signo;
- 10 para cada grupo de coeficientes en la unidad de transformación para la cual se ha codificado el respectivo signo en el grupo de coeficientes utilizando ocultación de bits de signo, sumar los valores absolutos de los coeficientes en ese grupo de coeficientes para obtener una suma resultante para ese grupo de coeficientes; y en función de si la paridad de la suma resultante para ese grupo de coeficientes es par o impar, asignar un respectivo valor de bit de signo que representa al respectivo signo.
- 15 2. El procedimiento reivindicado en la reivindicación 1, en el que cada respectivo valor de bit de signo es asignado a un respectivo elemento de sintaxis que no es un bit de signo de cualquiera de los coeficientes de ese grupo de coeficientes.
- 20 3. El procedimiento reivindicado en la reivindicación 1, en el que cada respectivo valor de bit de signo es un signo de un respectivo coeficiente en otro grupo de coeficientes en la unidad de transformación.
- 25 4. El procedimiento reivindicado en la reivindicación 2, en el que cada respectivo elemento de sintaxis es un respectivo elemento de sintaxis de vector de movimiento asociado con la unidad de transformación.
- 30 5. El procedimiento de la reivindicación 1, en el que el respectivo conjunto de coeficientes de cada grupo de coeficientes es un bloque de coeficientes de 4×4 .
- 35 6. El procedimiento reivindicado en la reivindicación 1, en el que la determinación comprende determinar que hay al menos un número umbral de coeficientes dentro de ese grupo de coeficientes entre un primer coeficiente distinto de cero en ese grupo de coeficientes y un último coeficiente distinto de cero en ese grupo de coeficientes, en un orden de exploración dentro de ese grupo de coeficientes.
- 40 7. El procedimiento reivindicado en la reivindicación 1, en el que la determinación se basa en una característica de los coeficientes de ese grupo y en el que la característica incluye que la suma resultante para ese grupo de coeficientes supera un valor de umbral.
- 45 8. El procedimiento reivindicado en la reivindicación 1, en el que la determinación se basa en una característica de los coeficientes de ese grupo de coeficientes y en el que la característica incluye que un conteo de coeficientes distintos de cero en ese grupo de coeficientes supera un umbral.
9. El procedimiento reivindicado en la reivindicación 1, en el que la determinación comprende determinar si un respectivo grupo de coeficientes adyacentes contiene al menos un coeficiente distinto de cero.
- 50 10. El procedimiento reivindicado en la reivindicación 9, en el que el respectivo grupo de coeficientes adyacentes comprende cualquiera de entre un grupo de coeficientes a la derecha y un grupo de coeficientes debajo de ese grupo de coeficientes.
- 55 11. El procedimiento de la reivindicación 1, en el que la unidad de transformación es cuadrada y tiene un tamaño que es uno de entre 4×4 , 8×8 , 16×16 y 32×32 .
12. El procedimiento reivindicado en la reivindicación 1, en el que la unidad de transformación no es cuadrada.
- 60 13. El procedimiento de la reivindicación 12, en el que la unidad de transformación tiene un tamaño que es uno de entre 8×32 y 32×8 .
- 65 14. Un decodificador para decodificar un flujo de bits de vídeo codificado, comprendiendo el decodificador (1000): un procesador (1002); una memoria (1004); y una aplicación de decodificación (1006) almacenada en la memoria (1004) y que contiene instrucciones que, cuando son ejecutadas, hacen que el procesador (1002) realice el procedimiento de cualquiera de las reivindicaciones 1 a 13.

15. Un medio no transitorio y legible por un procesador que almacena instrucciones ejecutables por un procesador que, cuando son ejecutadas, configura uno o más procesadores para decodificar un flujo de bits de vídeo codificado realizando el procedimiento de cualquiera de las reivindicaciones 1 a 13.

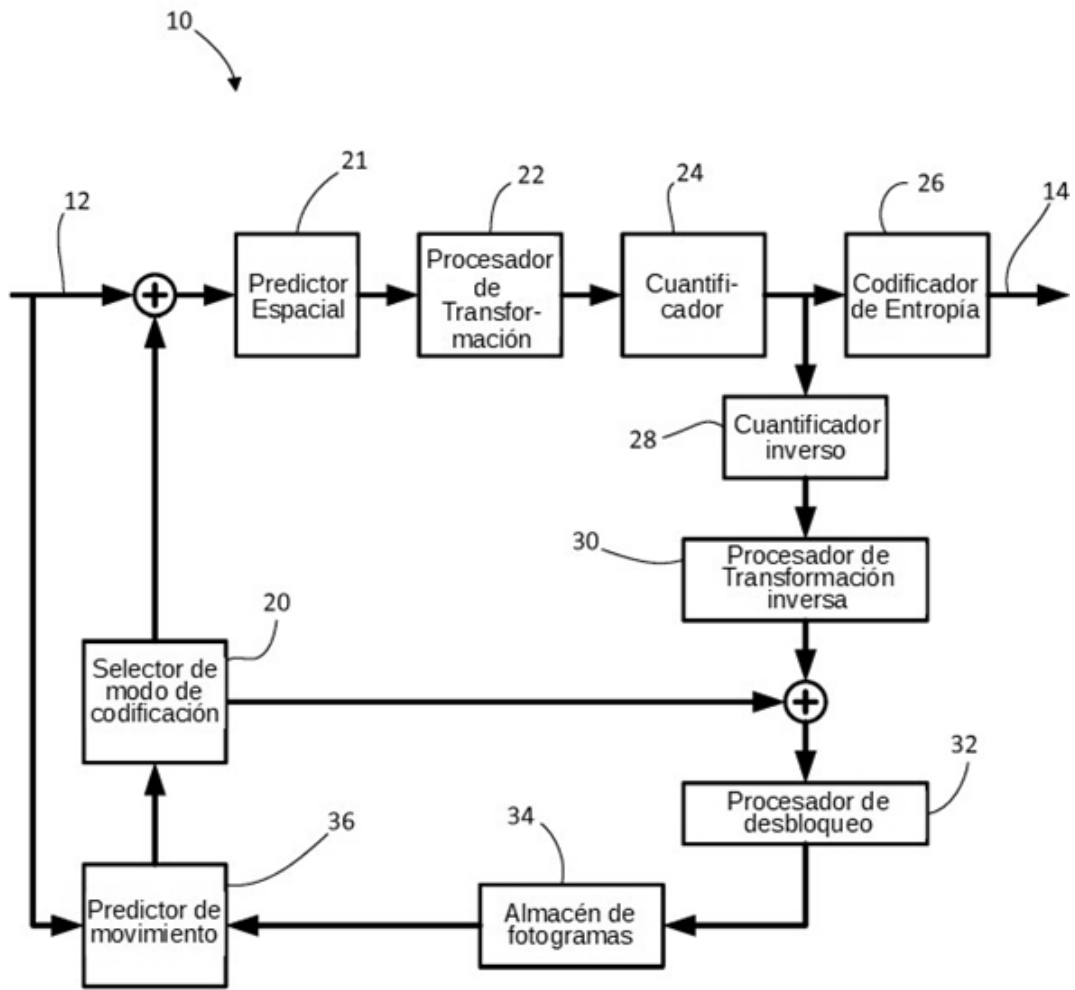


FIG. 1

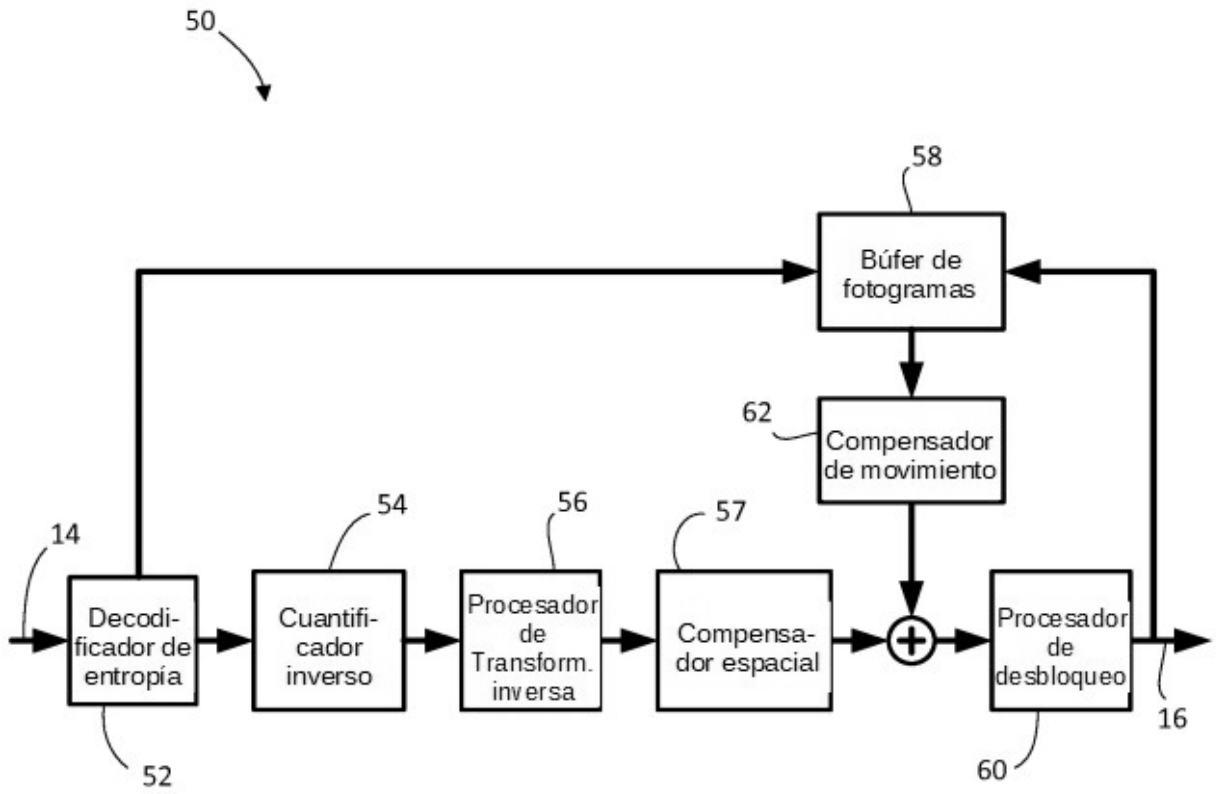


FIG. 2

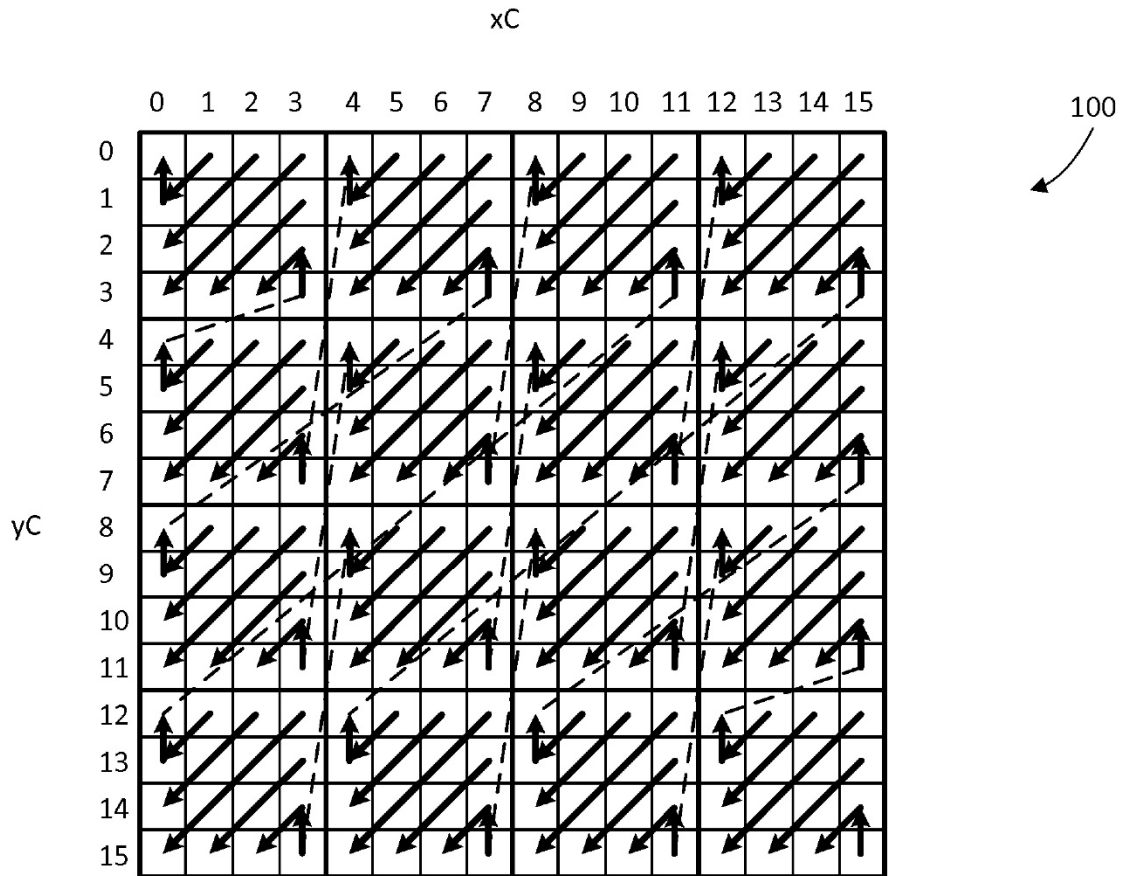


FIG. 3

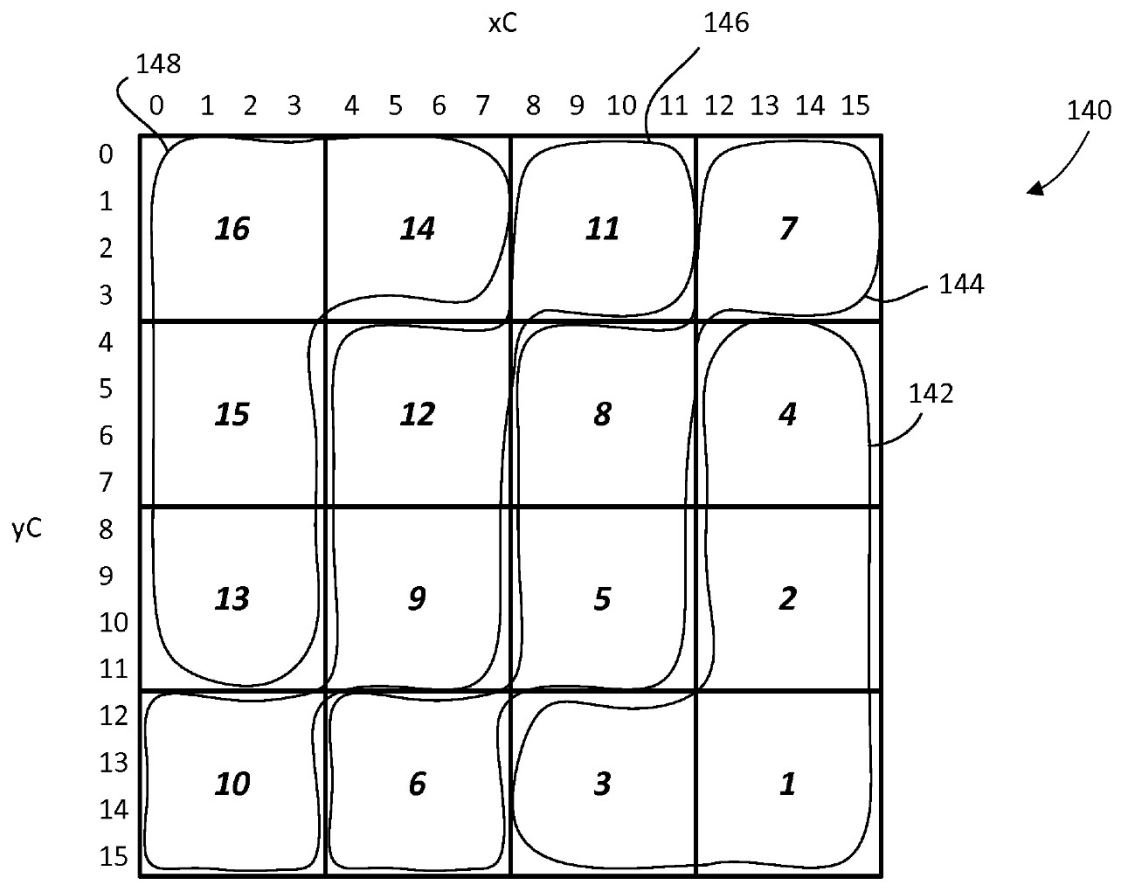


FIG. 5

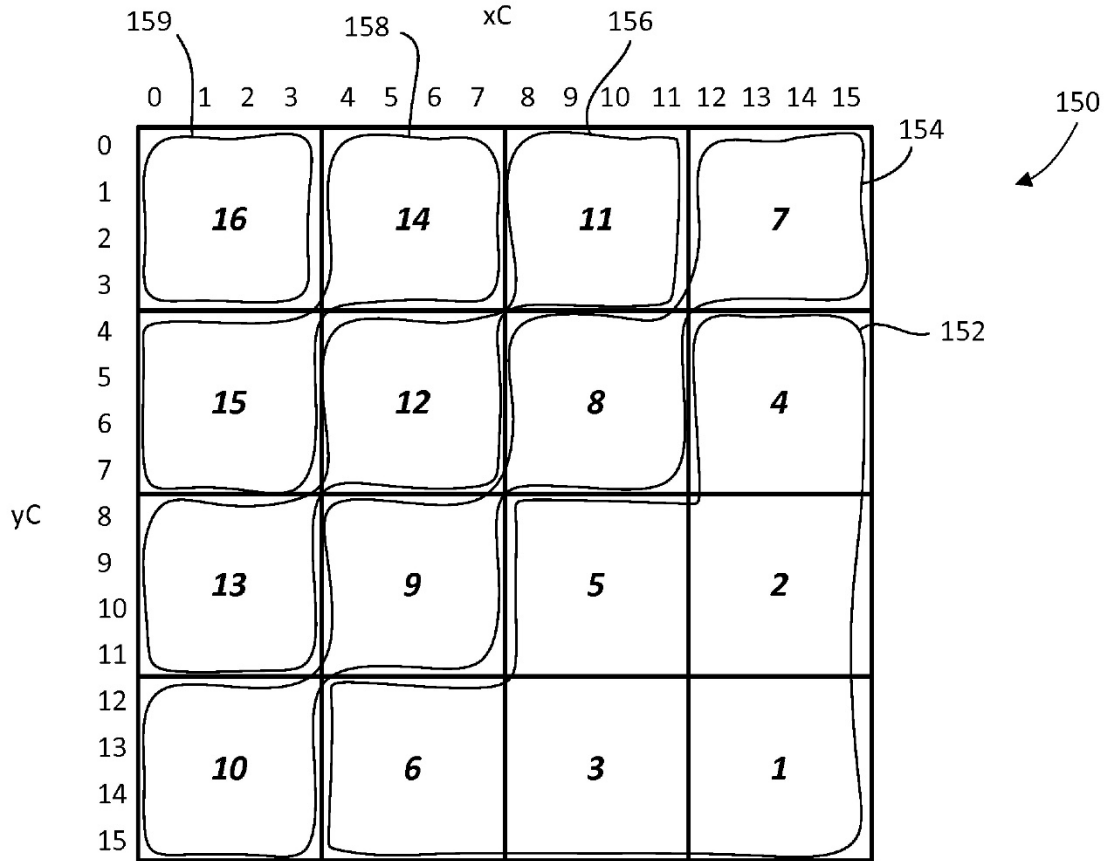


FIG. 6

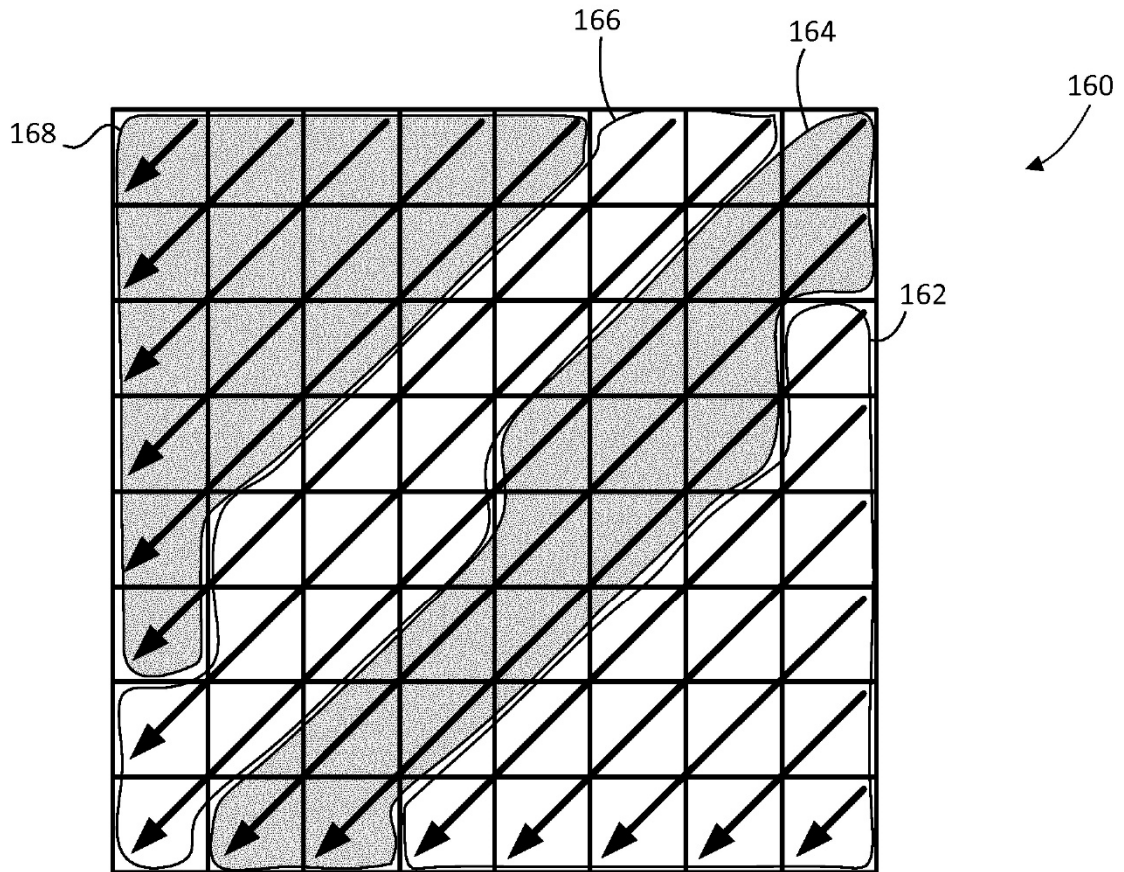


FIG. 7

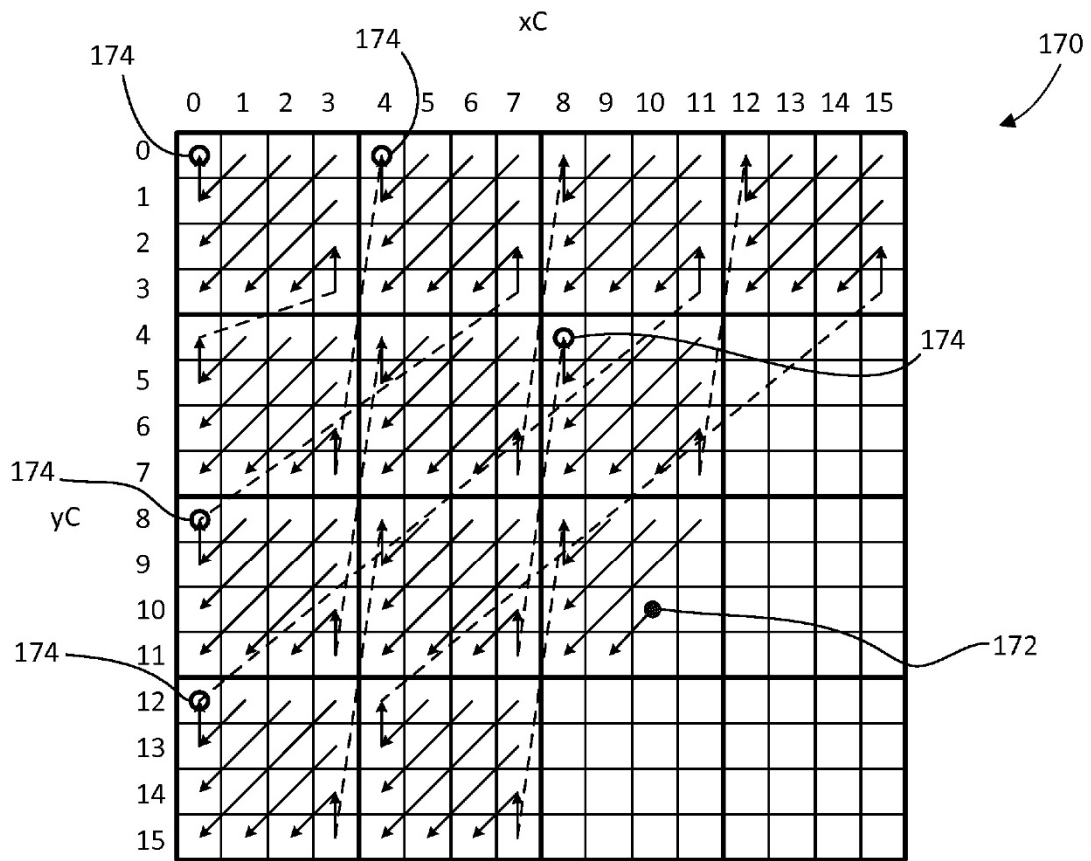


FIG. 8

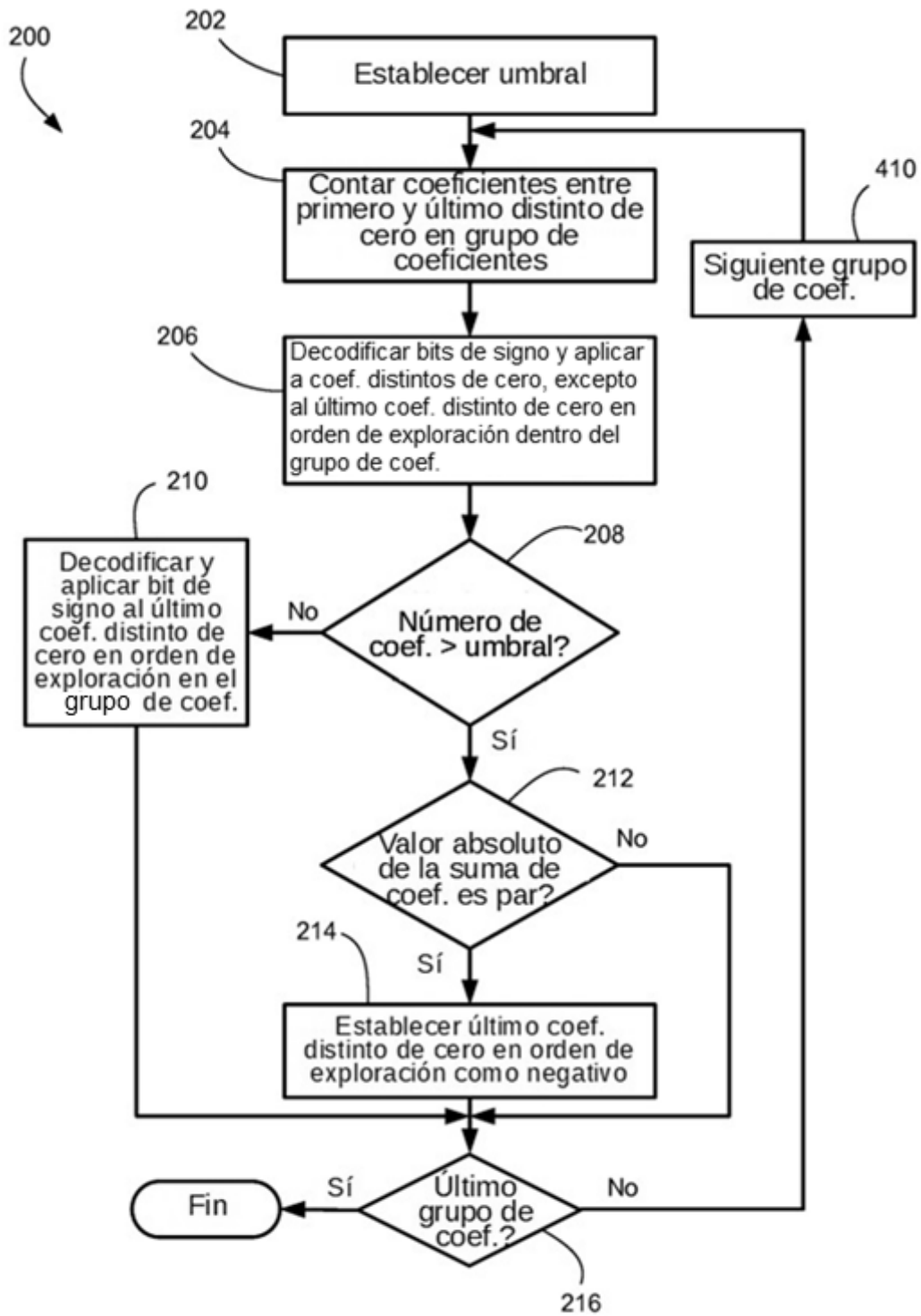


FIG. 9

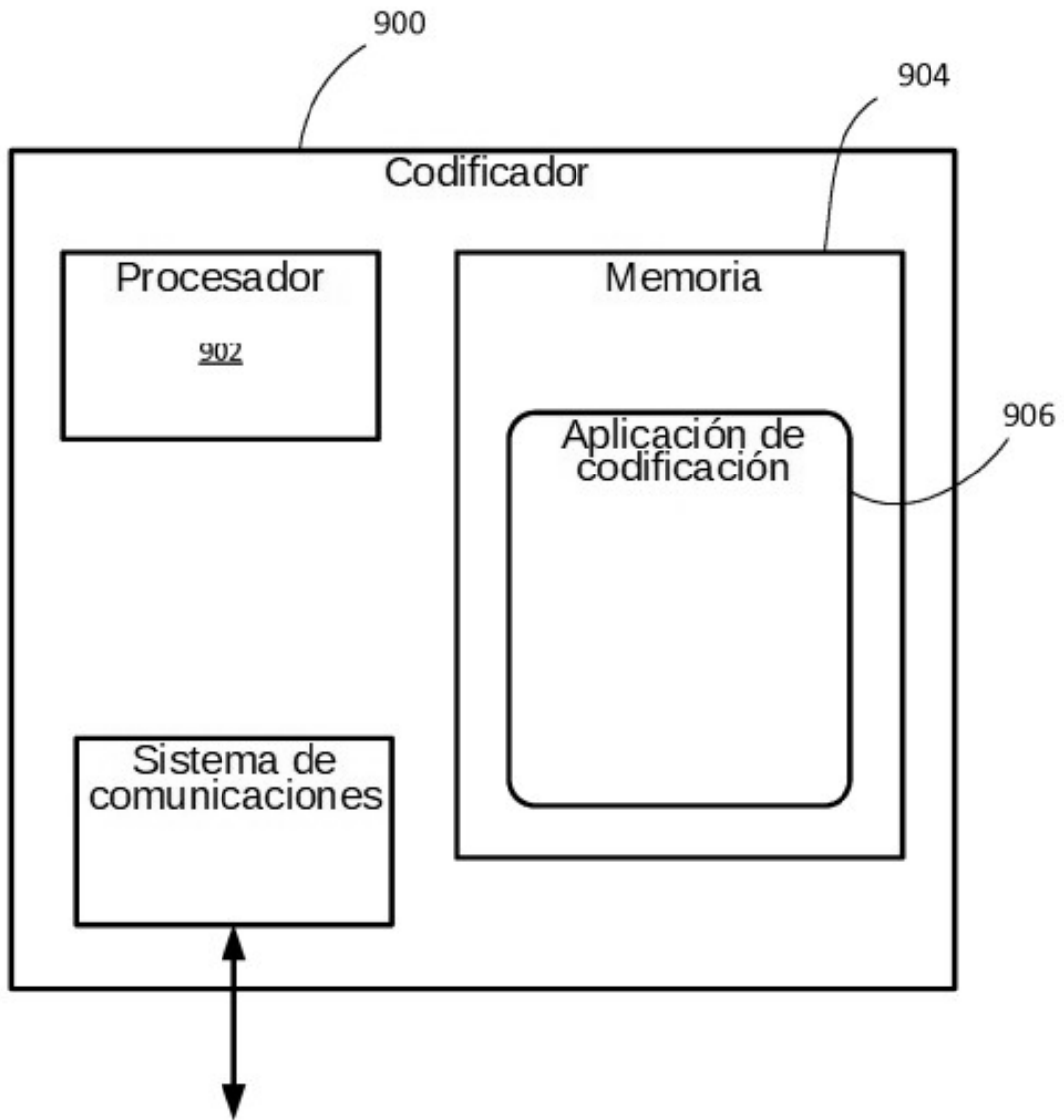


FIG. 10

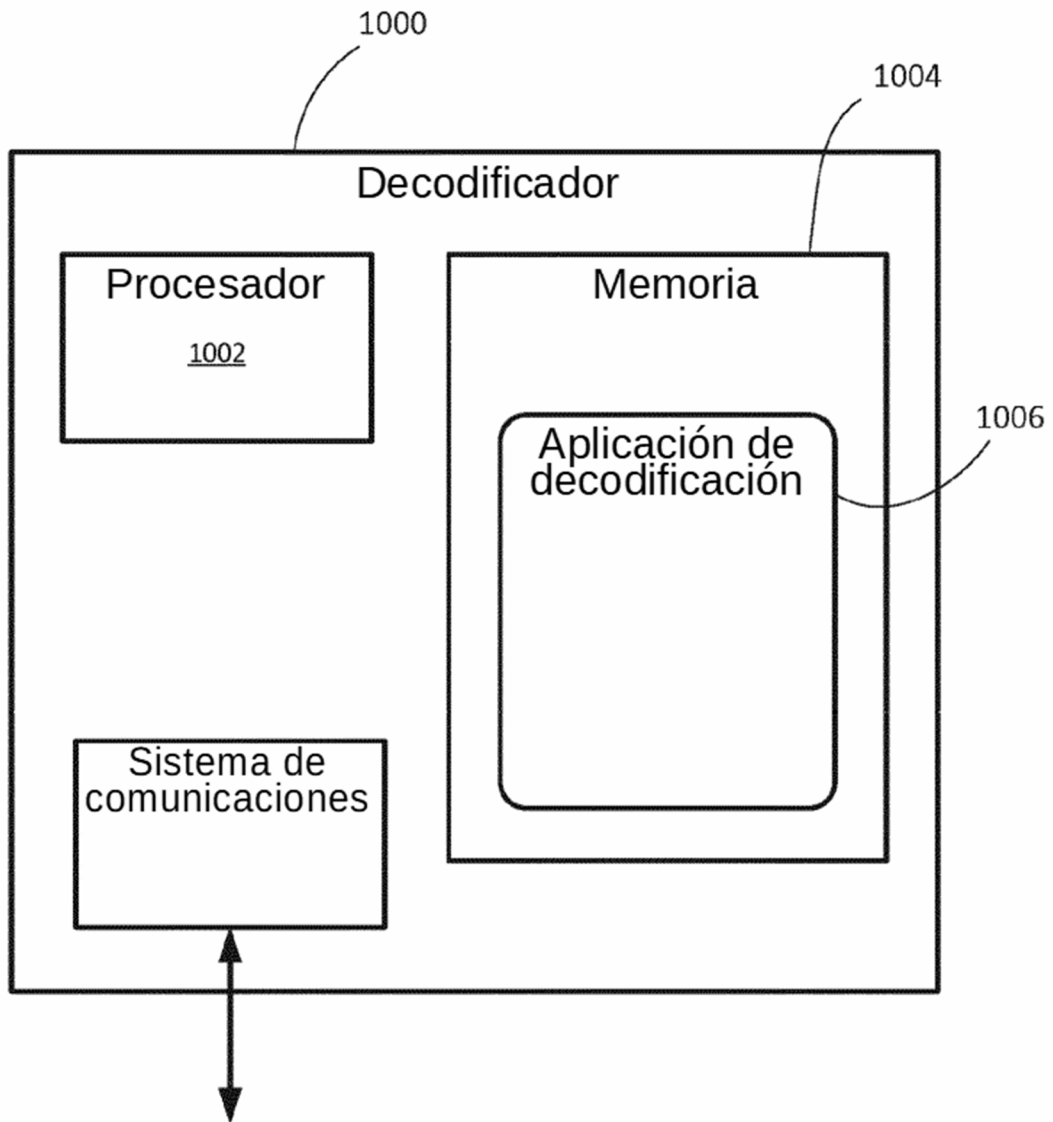


FIG. 11