(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 496 365**

(51) Int. CI.:

H04N 19/186 (2014.01)
H04N 19/196 (2014.01)
H04N 19/13 (2014.01)
H04N 19/18 (2014.01)
H04N 19/463 (2014.01)
H04N 19/129 (2014.01)
H04N 19/176 (2014.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **24.10.2011 E 11186392 (4)**

(97) Fecha y número de publicación de la concesión europea: **16.07.2014 EP 2587802**

(54) Título: **Codificación y descodificación de mapas significativos usando selección de la partición**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
18.09.2014

(73) Titular/es:

BLACKBERRY LIMITED (100.0%)
2200 University Avenue East
Waterloo, ON N2K 0A7, CA

(72) Inventor/es:

KORODI, GERGELY FERENC;
ZAN, JINWEN y
HE, DAKE

(74) Agente/Representante:

DE ELZABURU MÁRQUEZ, Alberto

DESCRIPCIÓN

Codificación y descodificación de mapas significativos usando selección de la partición

AVISO DE DERECHOS DE AUTOR

CAMPO

La presente solicitud se refiere en general a la compresión de datos y, en particular, a los métodos y dispositivos para la codificación y descodificación de mapas significativos de vídeo utilizando selección de la partición.

ANTECEDENTES

La compresión de datos se realiza en gran número de contextos. Se utiliza muy comúnmente en comunicaciones y en redes de ordenadores para almacenar, transmitir y reproducir la información de manera eficaz. Encuentra particular aplicación en la codificación de imágenes, audio y video. El vídeo presenta un desafío significativo para la compresión de datos debido a la gran cantidad de datos necesarios para cada cuadro de video y a la velocidad con la que a menudo tiene que realizarse la codificación y la descodificación. El estado actual de la técnica para la codificación de vídeo es la norma de codificación de vídeo ITU-T H.264/AVC. En ella se define una serie de diferentes perfiles para diferentes aplicaciones, incluyendo el perfil Principal, el perfil de línea Base y otros. Una norma de codificación de vídeo de próxima generación se encuentra actualmente en fase de desarrollo por medio de una iniciativa conjunta de MPEG-ITU: Codificación de Vídeo de Alto Rendimiento (HEVC).

Hay una serie de normas para la codificación/descodificación de imágenes y videos, incluyendo la H.264, que utilizan procesos de codificación basados en bloques. En estos procesos, la imagen o el cuadro se dividen en bloques, normalmente de 4x4 u 8x8 y los bloques se transforman espectralmente en coeficientes cuantificados y codificados entrópicamente. En muchos casos, los datos que están siendo transformados no son los datos reales de pixeles, sino que son datos residuales tras una operación de predicción. Las predicciones puede ser intra-cuadro, es decir, de bloque a bloque dentro del cuadro/imagen o inter-cuadro, es decir, entre cuadros (también llamada predicción de movimiento). Se espera que la norma HEVC (también llamada H.265) tenga también estas características.

Al transformar espectralmente los datos residuales, muchas de estas normas prescriben el uso de una transformada discreta del coseno (DCT) o alguna variante de la misma. Los coeficientes DCT resultantes se cuantifican entonces usando un cuantificador para producir coeficientes cuantificados de dominios transformados o índices.

El bloque o matriz de coeficientes cuantificados de dominio transformados (a veces referidos como una "unidad de transformación") se codifica entrópicamente utilizando un modelo de contexto particular. En la norma H.264/AVC y en el trabajo en desarrollo actualmente para HEVC, los coeficientes cuantificados transformados se codifican (a) codificando una posición del último coeficiente significativo que indica la posición del último coeficiente distinto de cero en el bloque, (b) codificando un mapa significativo que indica las posiciones en el bloque (excepto la posición del último coeficiente significativo) que contienen coeficientes distintos de cero, (c) codificando las magnitudes de los coeficientes distintos de cero y (d) codificando los signos de los coeficientes distintos de cero. Esta codificación de los coeficientes cuantificados transformados ocupa a menudo el 30-80% de los datos codificados en el flujo de bits.

La codificación entrópica de los símbolos en el mapa significativo se basa en un modelo de contexto. En el caso de un bloque de luminancia o de crominancia de 4x4 o de unidad transformada (TU), se asocia un contexto separado con cada posición del coeficiente en la TU. Es decir, el codificador y el descodificador rastrean un total de 30 (excluyendo las posiciones del ángulo inferior derecho) contextos separados para las TUs de luminancia y crominancia de 4x4. Las TUs de 8x8 se particionan (conceptualmente con el fin de la asociación de contexto) en bloques de 2x2 de tal manera que un contexto distinto se asocia con cada bloque de 2x2 en la TU de 8x8. En consecuencia, el codificador y el descodificador rastrean un total de 16+16=32 contextos para las TUs de luminancia y crominancia de 8x8. Esto significa que el codificador y el descodificador mantienen el rastreo y buscan hasta 62 contextos diferentes durante la codificación y descodificación del mapa significativo. Cuando se tienen en cuenta las TUs de 16x16 y las TUs de 32x32, el número total de distintos contextos implicados es 88. Esta operación también está destinada a ser llevada a cabo a alta velocidad de cálculo.

La publicación PCT 2011/128303A2 publicada el 20 de octubre de 2011 describe un proceso de codificación de mapa significativo. Un mapa significativo puede ser particionado en sub-bloques, estando asociado cada sub-bloque a un contexto distinto.

BREVE SUMARIO

La presente solicitud describe métodos y codificadores/descodificadores para codificar y descodificar mapas significativos con codificación o descodificación adaptables al contexto. El codificador y el descodificador disponen de un particionado no espacialmente uniforme del mapa en partes, en el que las posiciones de los bits dentro de cada parte están asociadas a un contexto dado. Se describen a continuación conjuntos y procedimientos de particiones de ejemplo para seleccionar de entre conjuntos de particiones predeterminadas y para comunicar la selección al descodificador.

En un aspecto, la presente solicitud describe un método de descodificar un flujo de bits de datos codificados para reconstruir un mapa significativo para una unidad de transformación como se reivindica en la reivindicación 1 o en la reivindicación 3 o en la reivindicación 5 o en la reivindicación 6.

En otro aspecto, la presente solicitud describe un método de codificar un mapa significativo para una unidad de transformación como se reivindica en la reivindicación 8 o en la reivindicación 9 o en la reivindicación 10 o en la reivindicación 11.

En un aspecto adicional, la presente solicitud describe codificadores y descodificadores configurados para llevar a cabo tales métodos de codificación y descodificación.

En otro aspecto adicional más, la presente solicitud describe medios interpretables por ordenador no transitorios que almacenan instrucciones de programa ejecutables por ordenador que, cuando se ejecutan, configuran un procesador para llevar a cabo los métodos descritos de codificación y/o descodificación.

Otros aspectos y características de la presente solicitud los entenderán aquellos expertos en la técnica al revisar la siguiente descripción de ejemplos en unión de las figuras que se adjuntan.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

Se hará referencia ahora, a modo de ejemplo, a los dibujos que se adjuntan que muestran realizaciones de ejemplo de la presente solicitud, y en los que:

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 descodificador para la descodificación de vídeo;
La figura 3 ilustra esquemáticamente una partición de un bloque de 4x4 en seis partes, en la que las posiciones de bit en cada parte se mapean a un contexto;
La figura 4 muestra un afinamiento de la partición de la figura 3, que queda en nueve partes;
La figura 5 ilustra esquemáticamente una partición de un bloque de 8x8 en cuatro partes, en la que las posiciones de bit en cada parte se mapean a un contexto;
La figura 6 muestra un afinamiento de la partición de la figura 5, que queda en doce partes;
La figura 7 muestra, en forma de diagrama de flujo, un método de ejemplo de descodificar datos codificados para reconstruir un mapa significativo;
La figura 8 muestra un gráfico que ilustra la eficacia relativa de las particiones en grueso y fino y su dependencia del tamaño del sector codificado;
La figura 9 muestra un diagrama de bloques simplificado de una realización de ejemplo de un codificador; y
La figura 10 muestra un diagrama de bloques simplificado de una realización de ejemplo de un descodificador.
Números de referencia similares se pueden usar en diferentes figuras para indicar componentes similares.

DESCRIPCIÓN DE REALIZACIONES DE EJEMPLOS

En la descripción que sigue, se describen algunas realizaciones de ejemplo con referencia a la norma H.264 para la codificación de vídeo y/o a la norma HEVC en desarrollo. Los expertos en la técnica comprenderán que la presente solicitud no se limita a las normas H.264/AVC o HEVC sino que puede ser aplicable a otras normas de codificación/descodificación de vídeo, incluyendo posibles futuras normas, normas de codificación multi-visión, normas de codificación de vídeo redimensionable y normas de codificación de vídeo reconfigurable.

En la descripción que sigue, cuando se hace referencia a las imágenes de vídeo o los términos cuadro, imagen, sector, mosaico y grupo de sectores rectangulares pueden usarse de forma intercambiable.

Los expertos en la técnica apreciarán que, en el caso de la norma H.264, un cuadro puede contener uno o más sectores. También se apreciará que se efectúan determinadas operaciones de codificación/descodificación sobre la base de cuadro a cuadro, realizándose algunas sobre la base de sector a sector, algunas imagen a imagen, algunas mosaico a mosaico, y realizándose algunas por grupo de sectores rectangulares, dependiendo de los requisitos particulares o de la terminología de la norma de codificación de vídeo o de imagen o aplicable. En cualquier realización en particular, la norma de codificación de vídeo o de imagen aplicable puede determinar si las operaciones descritas a continuación se llevan a cabo con respecto a cuadros y/o a sectores y/o a imágenes y/o a mosaicos y/o a grupos rectangulares de sectores, según cada caso. En consecuencia, los expertos en la técnica entenderán, a la luz de la presente descripción, si las operaciones o procesos particulares descritos en el presente

documento y las referencias en particular a los cuadros, sectores, imágenes, mosaicos, grupos rectangulares de sectores son aplicables a cuadros, sectores, imágenes, mosaicos, grupos rectangulares de sectores, o si se aplica alguno o todos para una realización dada. Esto también se aplica a las unidades de transformación, unidades de codificación, grupos de unidades de codificación, etc., como se hará evidente a la luz de la descripción que sigue a continuación.

Se hace referencia ahora 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 descodificador 50 para la descodificación de vídeo. Se apreciará que el codificador 10 y el descodificador 50 descritos en el presente documento puede cada uno realizarse en un dispositivo de ordenador de aplicación específica o de propósito general, conteniendo uno o más elementos de gestión y de memoria. Las operaciones realizadas por el codificador 10 o por el descodificador 50, según el caso, pueden ser realizadas por medio de un circuito integrado de aplicación específica, por ejemplo, o por medio de instrucciones de programa almacenadas, ejecutables por un procesador de propósito general. El dispositivo puede contener software adicional, incluyendo, por ejemplo, un sistema operativo para controlar las funciones básicas del dispositivo. La gama de dispositivos y plataformas dentro de los cuales se pueden realizar el codificador 10 o el descodificador 50 será apreciada por los expertos en la técnica teniendo en cuenta la siguiente descripción.

El codificador 10 recibe una fuente de vídeo 12 y produce un flujo de bits codificados 14. El descodificador 50 recibe el flujo de bits codificados 14 y proporciona a la salida un cuadro de vídeo descodificado 16. El codificador 10 y el descodificador 50 pueden ser configurados para operar en conformidad con diversas normas de compresión de vídeo. Por ejemplo, el codificador 10 y el descodificador 50 pueden ser compatibles con la norma H.264/AVC. En otras realizaciones, el codificador 10 y el descodificador 50 pueden ser conformes con otras normas de compresión de vídeo, incluyendo las evoluciones de la norma H.264/AVC, tal como la norma HEVC.

El codificador 10 incluye un predictor espacial 21, un selector 20 del modo de codificación, un procesador 22 de la transformación, un cuantificador 24 y un codificador entrópico 26. Como apreciarán los expertos en la técnica, el selector 20 del modo de codificación determina el modo apropiado de codificación para la fuente de vídeo, por ejemplo si el cuadro/sector objetivo es del tipo I, P o B y si las unidades particulares de codificación (es decir, macro bloques) dentro del cuadro/sector son inter o intra codificadas. El procesador 22 de la transformación realiza una transformación de los datos de dominio espacial. En particular, el procesador 22 de la transformación aplica una transformación basada en bloques para convertir los datos de dominio espacial en componentes espectrales. Por ejemplo, en muchas realizaciones se utiliza la transformada discreta del coseno (DCT). Se pueden utilizar en algunos casos otras transformadas, tales como la transformada discreta del seno u otras. La transformación basada en bloques se lleva a cabo sobre la base de macro bloque o sub-bloque, en función del tamaño de los macro bloques. En la norma H.264, por ejemplo, un macro bloque típico de 16x16 contiene dieciséis bloques transformados 4x4 y el proceso DCT se realiza en los bloques de 4x4. En algunos casos, los bloques transformados pueden ser de 8x8, es decir, que hay cuatro bloques transformados por macro bloque. En otros casos más, los bloques transformados pueden ser de otros tamaños. En algunos casos, un macro bloque de 16x16 puede incluir una combinación no yuxtapuesta de bloques transformados de 4x4 y de 8x8.

Aplicando la transformación basada en bloques a un bloque de datos de píxeles da como resultado un conjunto de coeficientes de dominio transformado. Un "conjunto" en este contexto es un conjunto ordenado en el que los coeficientes guardan posiciones de coeficientes. En algunos casos el conjunto de dominio de coeficientes transformados puede ser considerado como un "bloque" o matriz de coeficientes. En la descripción de este documento las frases un "conjunto de coeficientes de dominio transformado" o un "bloque de coeficientes de dominio transformado " se utilizan indistintamente y tratan de indicar un conjunto ordenado de coeficientes de dominio transformado.

El conjunto de coeficientes de dominio transformado se cuantifica por medio del cuantificador 24. Los coeficientes cuantificados y la información asociada los codifica a continuación el codificador entrópico 26.
Los cuadros/sectores intra-codificados (es decir, tipo I) están codificados sin referencia a otros cuadros/sectores. En otras palabras, no emplean predicción temporal. Sin embargo los cuadros intra-codificados no se basan en la predicción espacial dentro del cuadro/sector, tal como se ilustra en la figura 1 por medio del predictor espacial 21. Es decir, cuando se codifica un bloque en particular, los datos en el bloque pueden ser en comparados con los datos de los píxeles próximos dentro de los bloques ya codificados para ese cuadro/sector. Usando un algoritmo de predicción, los datos de origen del bloque pueden ser convertidos en datos residuales. El procesador 22 de transformación codifica luego los datos residuales. La norma H.264, por ejemplo, prescribe nueve modos de predicción espacial para bloques de transformación 4x4. En algunas realizaciones, cada uno de los nueve modos se puede utilizar para tratar de forma independiente un bloque, y luego se usa la optimización de la tasa de distorsión para seleccionar el mejor modo.

La norma H.264 también prescribe el uso de predicción/compensación de movimiento para aprovechar la predicción temporal. Por consiguiente, el codificador 10 tiene un bucle de realimentación que incluye un des-cuantificador 28, un procesador 30 de transformación inversa y un procesador 32 de desbloqueo. El procesador 32 de desbloqueo puede incluir un procesador de desbloqueo y un procesador de filtrado. Estos elementos reflejan el proceso de

descodificación aplicado por el descodificador 50 para reproducir el cuadro/sector. Un almacén 34 de cuadros se utiliza para almacenar los cuadros reproducidos. De esta manera, la predicción de movimientos se basa en lo que serán los cuadros reconstruidos en el descodificador 50 y no en los cuadros originales, que pueden diferir de los cuadros reconstruidos debido a la compresión con pérdida de datos involucrados en la codificación/descodificación. Un predictor 36 de movimiento utiliza los cuadros/sectores almacenados en el almacén 34 de cuadros como cuadros/sectores de origen para la comparación con un cuadro actual con el propósito de identificar bloques similares. De acuerdo con ello, para los macro bloques a los que se aplica la predicción de movimiento, los "datos de origen", que codifica el procesador 22 de la transformación son los datos residuales que se obtienen del proceso de predicción de movimiento. Por ejemplo, pueden incluir información acerca del cuadro de referencia, un desplazamiento espacial o "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 cuadro de referencia y/o al vector de movimiento puede no ser tratada por el procesador 22 de la transformación y/o por el cuantificador 24, pero en su lugar puede ser suministrada al codificador entrópico 26 para codificarla como parte del flujo de bits junto con los coeficientes cuantificados.

Los expertos en la técnica apreciarán los detalles y las posibles variaciones para la realización de codificadores según la norma H.264.

El descodificador 50 incluye un descodificador entrópico 52, un des-cuantificador 54, un procesador 56 para la transformación inversa, un compensador espacial 57, y un procesador de desbloqueo 60. El procesador de desbloqueo 60 puede incluir procesadores de desbloqueo y de filtrado. Una memoria intermedia 58 de cuadros suministra los cuadros reconstruidos para su uso por un compensador 62 de movimiento al aplicar la compensación de movimiento. El compensador espacial 57 representa la operación de recuperar los datos de vídeo para un bloque intra-codificado en particular de un bloque previamente descodificado.
El flujo 14 de bits es recibido y descodificado por el descodificador entrópico 52 para recuperar los coeficientes cuantificados. También se puede recuperar información lateral durante el proceso de descodificación entrópico, algo de la cual puede suministrarse al bucle de compensación de movimiento para su uso en la compensación de movimiento, si resulta de aplicación. Por ejemplo, el descodificador entrópico 52 puede recuperar vectores de movimiento y/o información sobre el cuadro de referencia para los macro bloques inter-codificados.

Los coeficientes cuantificados son entonces des-cuantificados por el des-cuantificador 54 para producir los coeficientes del dominio de la transformación, que luego son sometidos a una transformada inversa por el procesador 56 de transformación inversa para recrear los "datos de vídeo". Se apreciará que, en algunos casos, tal como con en un macro bloque intra-codificado, los "datos de vídeo" recreados son los datos residuales para su uso en la compensación espacial relativa a un bloque previamente descodificado dentro del cuadro. El compensador espacial 57 genera los datos de vídeo a partir de los datos residuales y los datos de píxeles a partir de un bloque descodificado previamente. En otros casos, tales como los macro bloques inter-codificados, los datos de vídeo "recreados" del procesador 56 de la transformación inversa son los datos residuales para su uso en la compensación de movimiento relativa a un bloque de referencia de un cuadro diferente. Ambas compensaciones, tanto la espacial como la de compensación de movimiento pueden ser referidas en este documento como "operaciones de predicción".

El compensador 62 de movimiento localiza un bloque de referencia específico dentro de la memoria intermedia 58 de cuadros para un macro bloque inter-codificado en particular. Lo hace basándose en el vector de movimiento e información del cuadro de referencia y el movimiento especificado para el macro bloque inter-codificado. Se aportan entonces los datos de píxeles del bloque de referencia para combinarlos con los datos residuales para llegar a los datos de vídeo reconstruido para ese macro bloque.

Un proceso de desbloqueo/filtrado puede entonces ser aplicado a un cuadro/sector reconstruido, tal como se indica, por el procesador 60 de desbloqueo. Después del desbloqueo/filtrado, el cuadro/sector queda como el cuadro de vídeo descodificado 16, por ejemplo, para su visualización en un dispositivo de visualización. Se entenderá que la máquina de reproducción de vídeo, tal como un ordenador, un descodificador, un reproductor de DVD o Blu-Ray, y/o un dispositivo de mano móvil, pueden almacenar temporalmente cuadros descodificados en una memoria antes de mostrarlos en un dispositivo de salida.

Se espera que los codificadores y descodificadores conformes a la norma HEVC tengan muchas de estas mismas o similares características.

Codificación del mapa significativo

Como se señaló anteriormente, la codificación entrópica de un bloque o conjunto de coeficientes de dominios transformados cuantificados incluye codificar el mapa significativo para ese bloque o un conjunto de coeficientes de dominios transformados cuantificados. El mapa significativo es un mapeado binario del bloque indicando en qué posiciones aparecen los coeficientes distintos de cero (excepto la última posición). El bloque puede tener ciertas características con las que se asocia. Por ejemplo, puede ser desde un sector intra-codificado o desde un sector inter-codificado. Puede ser un bloque de luminancia o un bloque de crominancia. El valor QP para el sector puede

variar de sector a sector. Todos estos factores pueden tener un impacto sobre la mejor manera en la que la entropía codifica el mapa significativo.

El mapa significativo se convierte en un vector de acuerdo con el orden de la exploración (que puede ser vertical, horizontal, diagonal, zig zag o en cualquier otro orden de exploración prescrito por la norma de codificación aplicable). Cada bit significativo es entonces codificado entrópicamente utilizando el esquema de codificación aplicable adaptable al contexto. Por ejemplo, en muchas aplicaciones se puede utilizar un esquema de codificación aritmética binaria adaptable al contexto (CABAC). Otras ejecuciones pueden utilizar otros códecs adaptables al contexto con binarización. Los ejemplos incluyen codificación aritmética binaria (BAC), codificación variable a variable (V2V) y codificación de longitud variable a fijo (V2F). A cada posición de bit, se le asigna un contexto. Al codificar el bit en esa posición de bit, el contexto asignado y el historial del contexto en ese punto, determinan la probabilidad estimada de un símbolo menos probable (LPS) (o en algunas realizaciones un símbolo más probable (MPS)).

En los codificadores de vídeo existentes, la asignación del contexto se predetermina tanto para el codificador como para el descodificador. Por ejemplo, con un bloque de luminancia de 4x4, el borrador actual de la norma HEVC prescribe que cada posición de bit en el mapa significativo de 4x4 tenga un contexto único. Excluyendo la última posición, significa que se rastrean 15 contextos para codificar los mapas significativos de luminancia 4x4. Para cada posición de bit, el contexto asignado a esa posición determina la probabilidad estimada asociada con un LPS en esa posición. El valor real del bit se codifica entonces usando esa probabilidad estimada. Finalmente, el contexto asignado a esa posición se actualiza basándose en el valor real del bit. En el descodificador, los datos codificados se descodifican usando el mismo modelo de contexto. Se rastrea y se usa un contexto para cada posición de bit para determinar la probabilidad estimada para los datos descodificados para recuperar bits para esa posición.

La asignación del contexto se puede considerar como particionar el bloque de datos y mapear un contexto distinto para cada parte. Matemáticamente, el mapeo puede ser definido utilizando P: {0, ..., n-1} X {0, ..., N-1} {0, ..., m-1} como un conjunto de particiones. Las posiciones se indexan como {0, ..., n-1} x {0, ..., n-1}. Los números 0, ..., m-1 identifican diferentes particiones. Cada partición tiene un contexto designado asociado con ella. Este contexto puede ser utilizado exclusivamente para esa partición (en algunos casos, un contexto puede ser utilizado tanto para bloques de tipo luminancia como de crominancia).

Para cualesquiera dos conjuntos de particiones P y Q, si hay un mapeo T tal que T(P(i, j)) = Q(i, j) para todo i y j, entonces decimos que Q es un subconjunto de P o P es un afinamiento de Q.

La codificación funciona de la siguiente manera: a la TU de tamaño nxn se le asigna un conjunto de particiones P. El mapa significativo puede ser considerado como una matriz M(i,j). La matriz M leída en orden de exploración horizontal puede quedar como M(0, 0), M(0, 1), M(0, n-1), M(1, 0), M(1, 1), .. ., M(1, n-1), ... m(n-1, n-1). El orden de exploración define un mapeo uno a uno de la representación matricial a una representación vectorial. En forma vectorial, el orden de exploración corresponde a una permutación de los números 0, 1, ..., n^2-2. En ejecuciones prácticas, el indexado puede estar basado en un solo indexado vectorial de valor único o en un indexado doble de estilo matricial, lo que más convenga. M(i, j) se codifica en el contexto BAC correspondiente a P(i, j), y ese contexto se actualiza utilizando M(i, j). La descodificación se deriva del procedimiento de codificación de manera sencilla.

Se puede usar esta estructura para describir el esquema de codificación del mapa significativo actualmente propuesto para la norma HEVC. Cada uno de las TUs de 4x4 y de 8x8 se asocia a un conjunto de particiones por separado, denominado P4 y P8, respectivamente. Esto viene dado por:

P4(i, j) = 4*i + j i, j = 0, 1, 2, 3 [15 contextos en total]

P8(i, j) = 4* [i/2] + [j/2] i, j = 0, 1, 2, 3, 4, 5, 6, 7 [16 contextos en total]

Las mismas asignaciones se utilizan para luminancia y crominancia, pero los contextos para luminancia y crominancia están separados. Por lo tanto, el número total de contextos usados para estas TUs es 15 + 15 + 16 + 16 = 62.

Cabe señalar que la partición de los mapas significativos está uniformemente distribuida. Es decir, exactamente hay tantos contextos asignados a las posiciones de bits del cuadrante inferior derecho, como los asignados al cuadrante superior izquierdo. Una distribución uniforme de contextos puede no ser óptima para muchas realizaciones. Los contextos asociados con el cuadrante superior izquierdo se usan mucho más que los contextos con el cuadrante inferior derecho (ya que los mapas significativos a menudo finalizan antes de llegar a estas posiciones de bits inferior derecha). En consecuencia, hay menos datos disponibles para estos contextos, que los hace menos adaptativos rápidamente y, generalmente, menos eficaces.

Como se describirá a continuación, la partición y el mapeo mejorados encontrarán un mejor equilibrio entre los objetivos de precisión (que tiende hacia menor número de posiciones de bit por contexto) y la adaptabilidad (que tiende hacia más posiciones de bit por contexto a fin de proporcionar más datos y converger más rápidamente en

una estimación óptima de probabilidad). Un buen conjunto de particiones pondrá equilibrio entre el rendimiento de la compresión y el número de particiones m. Al optimizar conjuntos de particiones bajo estas dos restricciones, en teoría deberían ser evaluados todos los posibles casos de P para un tamaño TU dado.

Para entender la complejidad de esta tarea, se puede calcular el número de conjuntos de particiones esencialmente únicas para cualquier tamaño dado de TU nxn y de la partición m. Se observará que la disposición de matriz de las particiones es arbitraria y es posible una representación equivalente en forma vectorial, utilizando, por ejemplo, un orden de exploración horizontal. Obsérvese el mapeo resultante por P_v: {0, N-1} → {0, ..., m-1}, donde $N = n^2 - 1$ (es decir, excluye la posición del bit inferior derecha). Sea C(N, m) el número de tales mapeos suprayectivos, es decir, que el rango de P_v es {0, ..., m-1}, omitiendo aquellos mapeos que son simples permutaciones de mapeos ya contados (es decir, las particiones que se pueden re-etiquetar para dar lugar a otro, mapeo ya contado). Téngase en cuenta que C(N, 1) = 1 y C(N, N) = 1 para cualquier N ≥ 1. Para m > 1 todos los mapeos P_v se pueden separar en dos clases. En la primera clase, sea P_v (0) ≠ {Pv(1), ..., P_v (N-1) }; ya que los valores 1, ..., N-1 están ahora mapeados en {1,…, P_v (0) -1, P_v (0) +1,…, m-1}, el número de tales mapeos es C(N-1, m-1). En la segunda clase $P_v(0) \varepsilon \{P_v(1),\ldots, P_v(N-1)\}$; los valores 1, ..., N-1 se mapean en 0,…., m-1, que se puede hacer de m*C (N-1, m) maneras, y P_v (0) puede ser insertado en cualquiera de las particiones m, lo que resulta en m*C (N-1, m) posibilidades. Así, se obtiene la recurrencia C(N, m) = C(N-1, M-1) + m*C(N-1, m). Téngase en cuenta que de ese modo los números C(N, m) coinciden con los números de Stirling de segunda clase.

Utilizando esta fórmula, puede calcularse que el número total de conjuntos de particiones para TUs de 4x4, es decir, 15 coeficientes o posiciones de bits, es 1.382,958.545; el número de conjuntos de particiones que tienen exactamente 5 partes es 210,766.920, y los que tienen exactamente 10 partes son 12,662.650 Los números correspondientes para TUs de 8x8 (63 coeficientes) se expresan mejor en forma exponencial: el número total de diferentes conjuntos de particiones es $8{,}2507717*10^{63}$, el número de conjuntos que tienen no más de 16 partes es $3{,}5599620*10^{62}$, el número de conjuntos que tienen exactamente 5 partes es $9{,}0349827*10^{41}$ y los que tienen exactamente 10 partes son $2{,}7197285*10^{56}$. Desde cualquiera de estos conjuntos de particiones legítimamente formados para la compresión de vídeo, la selección de los mejores de tantos candidatos es una tarea importante y difícil.

Ejemplo de conjuntos de particiones

A través de pruebas y análisis empíricos, el ejemplo siguiente de conjuntos de particiones y los mapeos de contexto parecen dar como resultado un equilibrio ventajoso de la velocidad de cálculo y el rendimiento de compresión.

Se hace referencia ahora a la figura 3, que ilustra esquemáticamente una partición de un bloque de 4x4 en seis partes, etiquetadas individualmente P1, P2, ..., P6. Esto puede ser utilizado, por ejemplo, para los mapas significativos en el caso de bloques de 4x4. El contexto (C_0, C_1, ..., C_5) asociado con cada posición de bit se muestra en el bloque 100. Las posiciones de los bits dentro de la misma parte comparten todas el mismo contexto. Debe observarse que la parte P4 incluye dos áreas no contiguas. Las cuatro posiciones de bits en la parte P4 están asignadas cada una al contexto C3. La partición mostrada en la figura 3 puede ser indicada como P4-6, para indicar que la partición se refiere a un bloque de 4x4 y cuenta con 6 partes.

La figura 4 muestra esquemáticamente un afinamiento de P4-6, en el que una partición adicional divide la parte P2 en tres partes individuales; dichas partes individuales se etiquetan con P2, P5 y P6. También debe observarse que la parte P4 se ha dividido por la mitad de tal manera que las dos áreas no contiguas son ahora partes separadas, etiquetadas con P4 y P9 en esta ilustración de ejemplo. Esta estructura de partición se puede indicar con P4-9 que quiere decir que asigna 9 contextos a las 9 partes distintas del bloque 4x4.

La figura 5 ilustra una partición de un bloque de 8x8 en 4 partes separadas, etiquetadas P1 a P4. Cada uno de los contextos C0 a C3 se asigna respectivamente a cada una de las partes, como se muestra. Esta partición puede ser indicada con P8-4.

La figura 6 ilustra un afinamiento de P8-4 como P8-12. En este caso, la partición de P8-4 se subdivide adicionalmente de manera que las cuatro partes se subdividen en un total de 12 partes, como se ilustra en el diagrama. Por lo tanto, hay 12 contextos Co,…, C11 en esta partición.

En todos los ejemplos anteriores, se observará que la partición, y por lo tanto la posición/asignación de los contextos, no se distribuye uniformemente a través del bloque. Es decir, las partes más pequeñas en la partición tienden a agruparse hacia el cuadrante superior izquierdo y las partes más grandes en la partición tienden a situarse hacia el fondo y el lado derecho del bloque. Como resultado, los contextos asignados al cuadrante superior izquierdo tienden a tener menor número de posiciones de bits asociados con ellos (en general, aunque no siempre), y el(los) contexto(s) asignado(s) al fondo o al lado derecho tienden a tener más posiciones de bits asociados con ellos. Con el tiempo, esto tenderá a dar lugar a un uso más uniforme de los contextos. Es decir, esta posición espacial no uniforme tiende hacia una posición más uniforme de los bits a cada contexto.

También se debe observar que la partición P4-6 es un subconjunto de la partición P4-9 y la partición P8-4 es un subconjunto de la partición P8-12. Esta característica tiene relevancia para algunos procesos de selección de conjuntos de particiones, como se explicará a continuación.

En una aplicación, la derivada del índice del contexto para los conjuntos de particiones de 4x4 y de 8x8 se puede obtener consultando una tabla. En otra aplicación, el índice del contexto puede ser determinado por medio de operaciones lógicas. Por ejemplo, para el conjunto P4-6 la derivada del índice del contexto podría obtenerse como:

```
(X & 2)? ((y & 2)? 5: x):
    ((y & 2)? (y & 1? 3: 4):
        (x|y);
```

Se apreciará que los cuatro conjuntos de particiones de ejemplo descritos anteriormente son ejemplos. Otros conjuntos de particiones (o adicionales) pueden ser utilizados en los procesos de selección descritos a continuación.

Selección de conjuntos de particiones- asignación estática

La presente solicitud detalla cuatro procesos de selección de ejemplo. El primer proceso de selección de ejemplo es la asignación estática. En este proceso de ejemplo, el codificador y el descodificador se pre-configuran para utilizar un conjunto de particiones en particular para los mapas significativos que tienen características particulares. Por ejemplo, la asignación puede basarse en el tamaño de la TU, el tipo de texto (luminancia o crominancia), y/o en el valor de QP. Esta asignación puede ser especificada por el codificador en un encabezamiento que precede a los datos de vídeo, o puede estar pre configurada tanto en el codificador como en el descodificador.

En algunas ejecuciones, la asignación puede estar (en parte) basada en submuestreo de crominancia. Para los submuestreos 4:02:0 y 4:1:1, los componentes de crominancia contienen considerablemente menos información que el componente de luminancia, lo que sugiere el uso de conjuntos de particiones más gruesas para crominancia que para luminancia. Por ejemplo, P4-9 puede ser usada para luminancia de 4x4, P4-6 para crominancia de 4x4, P8-12 para luminancia de 8x8 y P8-4 para crominancia de 8x8. Esto daría lugar a 31 contextos.

Para el caso de submuestreo 4:4:4, los valores de crominancia tienen comparativamente elevada importancia, lo que motiva el uso de un conjunto de particiones más afinadas para crominancia. De acuerdo con ello, en un ejemplo se puede usar P4-9 para luminancia y crominancia de 4x4 y P8-12 para luminancia y crominancia de 8x8. Esto daría lugar a 42 contextos.

Obsérvese, sin embargo, que en algunas ejecuciones los contextos pueden estar compartidos entre tipos de texto. Por ejemplo, luminancia de 4x4 y crominancia de 4x4 pueden ambos usar un conjunto de particiones P4-9, pero los contextos en ese conjunto se usan tanto para luminancia como para crominancia. En otra realización, tanto luminancia de 4x4 como crominancia de 4x4 pueden utilizar un conjunto de particiones P4-9, pero pueden utilizar contextos distintos.

Se hace referencia ahora a la figura 7, que muestra, en forma de diagrama de flujo, una método 100 de ejemplo para descodificar un flujo de bits de datos codificados para reconstruir un mapa significativo. El método 100 comienza con la determinación del tamaño del mapa significativo en la operación 102. Esta determinación se basa en el último coeficiente significativo, que se especifica en el flujo de bits. El último coeficiente significativo puede, en algunas realizaciones, estar señalizado en binario usando una cadena de ceros para todas las posiciones de bits (en el orden de exploración) antes del último coeficiente significativo y un uno en la posición del bit del último coeficiente significativo. Alternativamente se puede señalizar usando un par de índices (x, y) que indican la posición del bit. En otra realización se puede señalizar usando un índice simple que indica la posición del bit en el orden de exploración. También se pueden utilizar otros mecanismos de señalización del último coeficiente significativo. En cualquier caso, el último coeficiente significativo informa al descodificador del tamaño del mapa significativo.

Las operaciones 104, 106 y 108 se realizan para cada posición del bit en el mapa significativo en el mismo orden en el que el codificador los habría codificado. En algunas realizaciones, esto se puede hacer en el orden de exploración. En algunas realizaciones, esto se puede hacer en el orden de exploración inversa. Siempre que el codificador y el descodificador utilicen el mismo orden, se puede hacer en cualquier orden arbitrario.

En la operación 104, el contexto para la posición actual del bit se determina a partir de un conjunto de particiones almacenadas. En el caso de este procedimiento de ejemplo, se puede utilizar la asignación estática de un conjunto de particiones. Por consiguiente, el tipo de texto y el tamaño de la unidad transformada determinan el conjunto de particiones almacenadas que especifica el contexto asignado para cada posición de bit. Como ejemplo, los conjuntos de particiones almacenadas pueden ser conjuntos de particiones P4-6, P4-9, P8-4 y P8-12 descritos en el presente documento.

En la operación 106, los datos codificados se descodifican para reconstruir un valor del bit para esa posición del bit basándose en el contexto determinado. Por ejemplo, el contexto puede proporcionar una probabilidad estimada de un LPS, de la que el motor CABAC produce un valor de bit a partir de los datos codificados. En la operación 108, el contexto determinado se actualiza a continuación, basándose en el valor del bit.

En la operación 110, el descodificador evalúa si posiciones adicionales de los bits permanecen en el mapa significativo y, si es así, repite las operaciones 104, 106 y 108 para la siguiente posición del bit.

Selección del conjunto de particiones - asignación específica de la secuencia

El segundo procedimiento de selección de ejemplo es la asignación específica de la secuencia. En este procedimiento de ejemplo, el codificador determina qué conjunto de particiones utilizar para determinadas categorías de TUs basándose en, por ejemplo, el tamaño de la TU, tipo de texto, el valor de QP u otras características. Esta determinación se aplica a la totalidad de la secuencia de vídeo. Los conjuntos de particiones seleccionadas se especifican en el encabezamiento de la secuencia. En consecuencia, el descodificador lee el encabezamiento de la secuencia y a partir de entonces sabe qué conjuntos de particiones usar para descodificar los mapas significativos en circunstancias particulares. Si se usa el mismo conjunto de particiones con más de un tipo de texto (por ejemplo, para ambos, luminancia de 4x4 y crominancia de 4x4), entonces el codificador también puede especificar si los contextos son compartidos o si los dos tipos de texto utilizan contextos distintos.

En una sintaxis de ejemplo, el codificador puede enumerar un identificador para cada conjunto de particiones a utilizar, en el que el mismo conjunto de particiones se puede enumerar más de una vez si su estructura de partición se aplica en más de una situación y si los contextos para más de una situación deben ser distintos. El codificador asigna entonces uno de los conjuntos de partición enumerados para cada "categoría" de mapa significativo (por ejemplo, luminancia de 4x4, crominancia de 4x4, luminancia de 8x8, crominancia de 8x8, etc.) en un orden predeterminado. En algunas realizaciones, el valor de QP también puede ser un factor en la determinación de las "categorías" de los mapas significativos.

Para ilustrar este ejemplo de sintaxis, considérense cuatro conjuntos de particiones, tales como P4-6, P4-9, P8-4 y P8-12, ejemplos dados anteriormente. Los cuatro conjuntos pueden ser indexados usando cuatro bits, tales como 00, 01, 10, 11, correspondientes a P4-6, P4-9, P8-4, y P8-12, respectivamente.

Si el codificador determina que P4-9 debe ser utilizado tanto para luminancia de 4x4 como para crominancia de 4x4 con contextos separados y que P8-12 debe ser utilizado tanto para luminancia de 8x8 como para crominancia de 8x8 pero con contextos compartidos, entonces el codificador genera un encabezamiento secuencial que incluye el indicador binario: 01011100011010.

El descodificador, tras leer este indicador desde el encabezamiento secuencial, reconocerá que se van a utilizar los conjuntos P4-9 (01), P4-9 (01), y P8-12 (11). El descodificador también reconocerá que habiéndolos enumerado de este modo, ahora van a ser referidos como "00" para el primer conjunto P4-9, "01" para el segundo conjunto P4-9 y "10" para el conjunto P8-12.

El descodificador lee entonces "00011010", en el que cada porción de dos bits especifica el conjunto de particiones a usar para cada una de luminancia de 4x4, de crominancia de 4x4, de luminancia de 8x8 y de crominancia de 8x8. Los bits indexan el conjunto de particiones por su orden en la lista leída justo anteriormente. En consecuencia, el descodificador lee 00 y sabe que esto se refiere al primer conjunto P4-9. Si entonces lee 01, se refiere al segundo conjunto P4-9. Los últimos cuatro bits, "10" y "10", le dicen al descodificador que se tiene que utilizar el mismo conjunto P8-12 tanto para la luminancia de 8x8 como para la crominancia de 8x8, con contextos compartidos.

Se comprenderá que se puede emplear otra sintaxis para señalar la selección del conjunto de particiones en el encabezamiento de la secuencia, y que lo anterior no es más que un ejemplo de ejecución.

El codificador puede seleccionar los conjuntos de particiones usando una tabla, una función constante, u otro mecanismo. La función y/o tabla pueden tener en cuenta el tamaño de la TU, el tipo de texto (luminancia o crominancia), el valor de QP, el número de píxeles en el sector/secuencia u otros factores.

Selección del conjunto de particiones- asignación específica del sector

El tercer ejemplo del procedimiento de selección es la asignación específica del sector.

Se ha observado que el equilibrio entre la adaptabilidad y la precisión se inclina hacia particiones gruesas cuando el tamaño del sector codificado es relativamente pequeño y se inclina hacia particiones finas cuando el tamaño del sector codificado es relativamente grande. En consecuencia, el número de bits a ser codificados, o más particularmente, el número de bits codificados que resulten del procedimiento de codificación, puede ser un factor significativo en la determinación del conjunto de particiones más adecuado.

Se hace referencia ahora a la figura 8, que muestra un gráfico 200 de ejemplo de la eficacia relativa de la partición gruesa contra la partición fina para varios tamaños de sector codificados. Cada tamaño de sector codificado a lo

largo del eje horizontal muestra dos columnas, una para un conjunto de particiones gruesas 202 y una para un conjunto de particiones finas 204. La altura de la columna se basa en el número de veces que el conjunto de particiones dado resulta de mejor rendimiento de compresión que el conjunto alternativo para un sector de prueba, dividido por el número total de secciones de prueba. Se observará que el conjunto del particiones gruesas 202 supera al conjunto de particiones finas 204 para sectores de pequeño tamaño y que el conjunto del particiones finas 204 supera al conjunto de particiones gruesas 202 para sectores de mayor tamaño.

Por consiguiente, se pueden fijar uno o más valores de umbral para conmutar desde un conjunto de partición más gruesa al siguiente conjunto de partición más fina. Con los conjuntos de ejemplo descritos anteriormente, sólo hay dos conjuntos de particiones (uno de gruesa, uno de fina) para cada tamaño de TU, por lo que el umbral puede fijarse en o alrededor de 64k, por ejemplo. En el caso en que se predefinan más conjuntos de partición para un tamaño dado de TU se pueden establecer valores umbrales adicionales, o de otro tipo.

En el proceso de asignación específica de sectores, el codificador selecciona un conjunto de particiones para las TUs de cada sector. La selección puede ser comunicada al descodificador en el encabezamiento del sector. Se puede utilizar una sintaxis como la descrita anteriormente para comunicar los conjuntos de particiones seleccionados para determinadas categorías de TUs. De esta manera, el codificador puede adaptar la selección de los conjuntos de particiones a las características de un sector en particular. Sin embargo, para hacerlo, el codificador necesitaría codificar el sector usando una selección de partición por defecto, analizar las características del sector (como el tamaño del sector codificado), y luego volver a codificarlo con una nueva selección de particiones (si difiere de la de por defecto). En algunas ejecuciones, esta carga computacional extra en el codificador puede ser aceptable, tal como donde la codificación se produce una vez (es decir, en la codificación de un vídeo para su almacenamiento en medios de distribución tal como DVD/Blu-Ray) y la reproducción tiene lugar posteriormente no en tiempo real, posiblemente múltiples veces. En otras ejecuciones, como videoconferencia o la grabación de vídeo de mano, puede ser inaceptable la carga de la codificación en dos etapas en el codificador.

Una opción es basar la selección del conjunto de particiones en las estadísticas del sector previamente codificado que tiene el mismo valor de QP y mismo tipo de sector (intra o inter). Si tal sector anterior existe para el video, entonces el codificador puede asignar conjuntos de particiones a las TUs basándose en las estadísticas (por ejemplo, tamaño codificado) del sector anterior similar. Si no existe un sector anterior, entonces el codificador puede utilizar las selecciones del conjunto de particiones por defecto.

Selección del conjunto de particiones- asignación dinámica

El cuarto ejemplo del procedimiento de selección utiliza una secuencia de conjuntos de particiones para cada TU, en el que cada conjunto de particiones sucesivas en la secuencia es una versión más afinada que la de su predecesor. Cada TU comienza con el primer conjunto de particiones en su lista, entonces, en cada límite LCU comprueba si el tamaño codificado hasta el momento ha superado un determinado límite. Cuando eso sucede, el siguiente conjunto de particiones de esa lista se asigna a la TU. La decisión sobre cuándo cambiar se basa en el sector actual, por lo tanto, se puede determinar por el descodificador de la misma manera como fue hecho por el codificador, y ninguna otra información necesita ser especificada en la secuencia de vídeo.

En este procedimiento de ejemplo, el cambio de un conjunto de particiones Q a otro conjunto P hace uso del hecho de que Q es un subconjunto de P. El contexto BAC asociado con cada parte P (i, j) es inicializado a T (P (i, j)) desde Q; y la propiedad subconjunto afirma que esta inicialización está bien definida. Si para dos posiciones de bit (i_1, j_1) y (i_2, j_2), $P(i_1, j_1) \neq P(i_2, j_2)$ pero $T(P(i_1, j_1) = T(P(i_2, j_2))$, entonces las partes de (i_1, j_1) y (i_2, j_2) se inicializan al mismo estado BAC, pero desde ese punto en que los dos contextos que corresponden a estas dos particiones trabajan de forma independiente, y pueden divergir.

[0090] Para dar un ejemplo, supóngase que las particiones P4-6 y P4-9 son ambas utilizadas para codificar los mapas significativos de luminancia de 4x4 y las particiones P8-4 y P8-12 son ambas utilizadas para codificar los mapas significativos de luminancia de 8x8. Supóngase que las particiones de crominancia son fijas en este caso. Téngase en cuenta que P4-9 es un afinamiento de P4-6, y P8-12 es un afinamiento de P8-4. El criterio de conmutación es dos valores de umbral, uno para 4x4 y otro para 8x8, del número de contenedores que el codificador aritmético binario ha codificado hasta el momento en el sector actual. La partición P4-6 se inicializa y se utiliza para el mapa significativo de luminancia de 4x4 y se inicializa la partición P8-4 y se utiliza para el mapa significativo de luminancia de 8x8, respectivamente. Después de haber codificado cada LCU, el número de contenedores que ha codificado la BAC se comprueba y se compara con el umbral de 4x4 y de 8x8. Si se supera el umbral de 4x4, se usa el conjunto de particiones P4-9 para el mapa significativo de luminancia de 4x4, y de manera similar, si se supera el umbral de 8x8, se utiliza la partición P8-12 para el mapa significativo de luminancia de 8x8, para todos las siguientes LCUs. Los valores de inicialización de las particiones P4-9 (definidos como C4-9 [i], como se muestra a continuación) se podrían copiar de los valores de la partición P4-6 (definidos como C4-6 [i] como se muestra a continuación) como sigue:

C4-9: { C4-6 [0], C4-6 [1], C4-6 [2], C4-6 [3], C4-6 [1], C4-6 [1],

C4-6 [4], C4-6 [5], C4-6 [3] }

El valor de inicialización de las particiones P8-12 (definidas como C8-12 [i], como se muestra a continuación) se copiaría de los valores de la P8-4 (definida como C8-4 [i], como se muestra a continuación) de la siguiente manera:

C8-12: { [C8-4 [0], C8-4 [0], C8-4 [1], C8-4 [2], C8-4 [3], C8-4 [1],

C8-4 [1], C8-4 [2], C8-4 [2], C8-4 [2], C8-4 [3], C8-4 [3] }

Desde la siguiente LCU en adelante, cada partición/contexto en P4-9 y P8-12 opera y se actualiza con independencia de cualesquiera otros contextos.
Dado que el descodificador podría contar el número de contenedores descodificados de la misma manera, el proceso anterior se podría repetir en el lado del descodificador, sin señalización explícita del encabezamiento del sector codificado.

Inicialización de la partición
Dado que cada parte dentro de un conjunto de particiones corresponde a un estado BAC, que se utiliza para la codificación y la descodificación de los bits en esa partición, se tiene que determinar al comienzo de cada sector el valor inicial de ese estado. El valor inicial es un estado BAC, que en la actual terminología de la norma HEVC es un valor entero en el intervalo (1, ..., 126). El bit menos significativo de este valor especifica el MPS, y los 6 bits restantes identifican la probabilidad del LPS. El estado uniforme con MPS = 1 y p(LPS) = 0,5 se identifica por el valor 64.

Los conjuntos de particiones descritos anteriormente se han elegido de tal manera que la inicialización del estado se puede eximir en algunas realizaciones sin pérdida significativa en el rendimiento de la compresión. Por lo tanto, siempre que una partición necesite ser inicializada, se puede fijar al estado uniforme.

En otra realización, se pueden proporcionar valores de inicialización. En una ejecución, los valores de inicialización proporcionados son para inter sectores. Sin embargo, más que especificar una función lineal de QP para cada parte, tipo de sector (I, P, B) y tipo de texto (luminancia, crominancia), en una realización, la presente solicitud propone un único valor para cada partición.

Como ejemplo, se pueden usar los siguientes valores de inicialización para las particiones descritas anteriormente. Téngase en cuenta que para mayor claridad visual, éstos se muestran en notación matricial, en la que se muestra el valor de inicialización del contexto para cada posición en la que el contexto se utiliza en ese conjunto de particiones; sin embargo, en ejecuciones prácticas, podría ser más compacta una notación vectorial, en la que se muestre el valor de inicialización para cada contexto (en un orden conocido) en lugar de cada posición de bit.

Valores de inicialización intra para P4-9:

[77 71 66 61
71 67 66 61
66 66 65 65
61 61 65],

Valores de inicialización intra para P4-6:

[67 60 55 46
60 60 55 46
55 55 54 54
46 46 54],

Valores de inicialización intra para P8-12:

[71 67 59 59 53 53 45 45
67 67 59 59 53 53 45 45
59 59 55 55 51 51 45 45
59 59 55 55 51 51 45 45
53 53 51 51 55 51 42 42
53 53 51 51 51 42 42 42
45 45 45 45 42 42 42 42
45 45 45 45 42 42 42],

Valores de inicialización intra para P8-4:

[62 62 48 48 41 41 33 33

62 62 48 48 41 41 33 33
48 48 48 48 41 41 33 33
48 48 48 48 41 41 33 33
41 41 41 41 48 41 33 33
41 41 41 41 41 33 33 33
33 33 33 33 33 33 33 33
33 33 33 33 33 33 33],

Valores de inicialización inter (B) P4-9:

[61 56 52 51
56 54 52 51
52 52 55 55
51 51 55],

Valores de inicialización inter (B) para P4-6

[60 49 43 36
49 49 43 36
43 43 48 48
36 36 48],

Valores de inicialización inter (B) para P8-12:

[59 52 45 45 38 38 37 37
52 52 45 45 38 38 37 37
45 45 40 40 37 37 37 37
45 45 40 40 37 37 37 37
38 38 37 37 40 37 40 40
38 38 37 37 37 40 40 40
37 37 37 37 40 40 40 40
37 37 37 37 40 40 40],

Valores de inicialización inter (B) para P8-4:

[56 56 37 37 27 27 25 25
56 56 37 37 27 27 25 25
37 37 37 37 27 27 25 25
37 37 37 37 27 27 25 25
27 27 27 27 37 27 25 25
27 27 27 27 27 25 25 25
25 25 25 25 25 25 25 25
25 25 25 25 25 25 25],

Valores de inicialización (P) para P4-9:

[62 57 54 51
57 55 54 51
54 54 55 55
51 51 55],

Valores de inicialización inter (P) para P4-6:

[61 51 43 34
51 51 43 34
43 43 48 48
34 34 48],

Valores de inicialización inter (P) para P8-12:

[60 54 47 47 42 42 39 39
54 54 47 47 42 42 39 39
47 47 43 43 41 41 39 39
47 47 43 43 41 41 39 39
42 42 41 41 43 41 41 41
42 42 41 41 41 41 41 41
39 39 39 39 41 41 41 41

39 39 39 39 41 41 41],

Valores de inicialización inter (P) para P8-4:

[55 55 37 37 27 27 21 21
55 55 37 37 27 27 21 21
37 37 37 37 27 27 21 21
37 37 37 37 27 27 21 21
27 27 27 27 37 27 21 21
27 27 27 27 27 21 21 21
21 21 21 21 21 21 21 21
21 21 21 21 21 21 21],

Orden de la exploración

Como se explicó anteriormente, la posición del último coeficiente significativo (LSC) se determina utilizando un orden de exploración. Los órdenes de exploración definidos de ejemplo incluyen horizontal, vertical, diagonal y en zig-zag. La codificación y descodificación del mapa significativo prosigue en el orden de exploración inverso especificado, hacia atrás desde el LSC.

En alguna realización, por ejemplo, hecha mediante hardware, puede ser ventajoso minimizar el número de veces que el codificador o descodificador deben cargar un nuevo contexto. Dado que cada posición en una parte dada del conjunto de particiones usa el mismo contexto, esto significa que puede ser más eficiente tratar todas las posiciones en una parte antes de proseguir con la siguiente parte. Por consiguiente, se puede usar en algunas realizaciones un orden de exploración diferente para codificar el mapa significativo que se utilizó para la determinación del LSC.

En una TU nxn, el orden de exploración de la codificación es una permutación arbitraria de los números 0, 1, ..., n^2-2. La permutación se aplica a las posiciones de la matriz enumeradas en orden de exploración horizontal. Se puede utilizar cualquier permutación, tantas como el codificador y descodificador acuerden en la misma permutación para cada conjunto de particiones. La permutación puede estar diseñada, por ejemplo, de forma que minimice el número de cambios entre contextos.

Para usar un ejemplo, recordemos que el conjunto de particiones para P4-6 viene dado por:

0 1 2 3
1 1 2 3
4 4 5 5
3 3 5

Si utilizamos el escaneo diagonal, entonces la permutación viene dada por

0, 4, 1, 8, 5, 2, 12, 9, 6, 3, 13, 10, 7, 14, 11 (1)

donde los números 0, 1,... 14 se refieren a las posiciones de bits de 4x4 en orden horizontal. En esta permutación de exploración en diagonal, el contexto se utiliza de este modo en el siguiente orden:

0, 1, 1, 4, 1, 2, 3, 4, 2, 3, 3, 5, 3, 5, 5 (2)

Para la codificación y descodificación del mapa significativo, se utilizan estos contextos en un orden de lectura hacia atrás desde la posición anterior a la LSC. Esto da lugar a más cambios de contexto que el siguiente orden de exploración o permutación:

0, 4,1, 1, 5, 2, 6, 3, 7, 12, 13, 8, 9, 10, 11, 14 (3)

lo que queda en los contextos que se utilizan en el siguiente orden:

0, 1, 1, 1, 2, 2, 3, 3, 3, 3, 4, 4, 5, 5, 5 (4)

En consecuencia, el orden de exploración (3) puede ser predefinido para su uso con P4-6 en lugar de la exploración diagonal (1) al tratar el mapa significativo, lo que resulta en la secuencia de contexto (4) en lugar de (2), dando lugar a un menor número de cambios de contexto entre coeficientes.

El orden de exploración reordenado para el mapa significativo para minimizar los cambios de contexto puede ser ventajoso en algunas ejecuciones de hardware. Si bien es posible tratar los bits de dos contextos diferentes en un solo ciclo de reloj, es más fácil ejecutar el tratamiento de los bits desde el mismo contexto en un solo ciclo de reloj. Reordenando los contenedores para agruparlos por contexto, es más fácil tratar múltiples contenedores por ciclo de reloj. Si los dos contextos que están siendo tratados con un solo ciclo de reloj son diferentes, entonces el

codificador/descodificador debe leer dos contextos diferentes y actualizar dos contextos diferentes. Puede ser más fácil producir una ejecución de hardware que actualice un único contexto dos veces en un ciclo de reloj que leer y actualizar dos.

Ejemplo detallado de sintaxis- realización de asignación estática

Trabajando en la sintaxis en la norma actualmente en desarrollo HEVC, se pueden hacer las siguientes modificaciones y/o adiciones a la sintaxis en algunas realizaciones de ejemplo para facilitar el uso de la asignación estática. En los siguientes ejemplos, la sintaxis se basa en una ejecución en la que los cuatro conjuntos de ejemplo, las separaciones descritas anteriormente, P4-6, P4-9, P8-4 y P8-12, se almacenan y se asignan, respectivamente, para su uso con crominancia 4x4, luminancia 4x4, crominancia 8x8 y luminancia 8x8.

Las entradas a este proceso son el índice de componente de color CLDX, el actual coeficiente de posición de exploración (xC, yC), es decir, la posición del bit, y el tamaño de bloque de transformación log2TrafoSize. La salida de este proceso es ctxIdxInc. La variable sigCtx depende de la posición actual (xC, yC), del índice del componente de color cIdx, del tamaño de bloque de transformación y de los contenedores previamente descodificados del elemento de sintaxis significant_coeff_flag. Para la derivada de sigCtx, se aplica el siguiente procedimiento:

Si log2TrafoSize es igual a 2, sigCtx se deriva de la siguiente manera:

sigCtx = CTX_IND_MAP_4x4 [cldx] [(yC << 2) + xC]

De lo contrario, si log2TrafoSize es igual a 3, sigCtx se deriva de la siguiente manera:

sigCtx = CTX_IND_MAPA_8x8 [cldx] [(yC << 3) + xC]

Las constantes de CTX_IND_MAP_4x4 y CTX_IND_MAP_8x8 se pueden definir para luminancia y crominancia de la siguiente manera:

```
Constante estática UInt CTX_ IND_MAP_4x4 [2] [15] =
{
    / / Mapa de LUMINANCIA
    {
        0,    1,    2,    3,
        4,    5,    2,    3,
        6,    6,    7,    7,
        8,    8,    7,
    },

    / / Mapa de CROMINANCIA
    {
        0,    1,    2,    3,
        1,    1,    2,    3,
        4,    4,    5,    5,
        3,    3,    5
    },

constante estática  Ulnt CTX_IND_MA_8x8 [2] [63]   =
{
    / / Mapa de LUMINANCIA
    {
        0,    1,    2,    2,    3,    3,    4,    4,
        1,    1,    2,    2,    3,    3,    4,    4,
        5,    5,    6,    6,    7,    7,    4,    4,
        5,    5,    6,    6,    7.    7,    4,    4,
        8,    8,    9,    9,    6,    7,    10,   10,
        8,    8,    9,    9,    9,    10,   10,   10,
        11,   11,   11,   11,   10,   10,   10,   10,
        11,   11,   11,   11,   10,   10,   10,

    },

    / / Mapa de CROMINANCIA
    {

        0,    0,    1,    1,    2,    2,    3,    3,
        0,    0,    1,    1,    2,    2,    3,    3,
```

```
                1,   1,   1,   1,   2,   2,   3,   3,
                1,   1,   1,   1,   2,   2,   3,   3,
                2,   2,   2,   2,   1,   2,   3,   3,
                2,   2,   2,   2,   2,   3,   3,   3,
                3,   3,   3,   3,   3,   3,   3,   3,
                3,   3,   3,   3,   3,   3,   3,   3,
        },
```

El aumento del índice de contexto ctxIdxInc se deriva utilizando el índice de componente de color cIdx, el tamaño de bloque de transformación log2TrafoSize, sigCtx y los conjuntos de particiones, como se indica a continuación.
Los valores para ctxOffset[max(log2TrafoSize-2, 2)][cIdx] se definen en la siguiente tabla:

max(log2TrafoSize-2, 2)	cIdx=0	cIdx=1
0	0	0
1	num_partitions_luma4x4	num_partitions_chroma4x4
2	num_partitions_luma4x4 + num_partitions_luma8x8	num_partitions_chroma4x4 + num_partitions_chroma8x8

Por ejemplo, si el conjunto de particiones P4-9 se utiliza para bloques de luminancia 4x4, P4-6 para bloques de crominancia 4x4, P8-12 para bloques de luminancia 8x8 y P8-4 para bloques de crominancia 8x8, la tabla de referencia toma los siguientes valores:

max(log2TrafoSize-2, 2)	cIdx=0	cIdx=1
0	0	0
1	9	6
2	21	10

Se observa que ctxIdxInc se refiere a la posición inicial del bloque de 4x4 del componente cIdx. El valor de ctxIdxInc se deriva de:

ctxIdxInc = ctxOffset [max (log2TrafoSize-2, 2)] [cIdx] + sigCtx

En cuanto a la inicialización de las variables de contexto, la asociación entre ctxIdx y los elementos de sintaxis para cada tipo de sector pueden especificarse por:

	Elemento de sintaxis	ctxIdxTable	Tipo de Sector		
			I	P	B
residual_coding()	last_significant_coeff_x	...	...	...	...
	last_significant_coeff_y	...	...	...	...
	significant_coeff_flag	Tabla	0..56	0..56	0..56
	coeff_abs_level_greater1_flag	...	...	...	...
	coeff_abs_level_greater2_flag	...	...	...	...

Suponiendo una realización de inicialización uniforme, la Tabla ctxIdx referida anteriormente para el elemento de sintaxis significant_coeff_flag viene dada por:

Variables de inicialización	significant_coeff_flag ctxIdx															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64
	48	49	50	51	52	53	54	55	56							
m	0	0	0	0	0	0	0	0	0							
n	64	64	64	64	64	64	64	64	64							

Sin embargo, si se ejecuta una inicialización constante en lugar de una inicialización uniforme, entonces la asociación entre el elemento de sintaxis y ctxIdx puede modificarse como se muestra a continuación:

	Elemento de sintaxis	ctxIdxTable	Tipo de Sector		
			I	P	B
residual_coding()	last_significant_coeff_x	...	...	...	...
	last_significant_coeff_y	...	...	...	...
	significant_coeff_flag (I)	Tabla I	0..56		
	significant_coeff_flag (B)	Tabla B		0..56	
	significant_coeff_flag (P)	Tabla P			0..56
	coeff_abs_level_greater1_flag	...	...	...	...
	coeff_abs_level_greater2_flag	...	...	...	...

La Tabla I ctxIdxTable referida anteriormente para el elemento de sintaxis significant_coeff_flag(I) puede entonces venir dada por:

Variables de inicialización	significant_coeff_flag ctxIdx															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n	77	71	66	61	71	67	66	65	61	71	67	59	53	45	59	55
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
m	0	0	0	0	0	-15	-14	-15	-4	0	-2	-7	-15	-4	1	-4
n	51	53	51	42	45	119	104	106	49	62	72	88	112	28	54	72
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
m	-7	-10	0	0	0	0	0	0	0	0	0	0	15	7	5	14
n	82	96	67	60	55	46	55	54	62	48	41	33	59	56	57	11
	48	49	50	51	52	53	54	55	56							
m	10	7	5	11	-9	5	7	10	13							
n	45	53	61	59	38	46	49	48	47							

La Tabla B ctxIdx mencionada anteriormente para el elemento de sintaxis significant_coeff_flag (B) puede entonces venir dada por:

Variables de inicialización	significant_coeff_flag ctxIdx															
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n	61	56	52	51	56	54	52	55	51	59	52	45	38	37	45	40
	16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
m	0	0	0	0	0	0	0	-3	2	3	0	-3	-9	-4	4	1
n	37	38	37	40	37	78	66	68	31	53	65	74	93	20	44	57
	32	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**
m	0	-1	0	0	0	0	0	0	0	0	0	0	28	16	11	26
n	65	72	60	49	43	36	43	48	56	37	27	25	29	35	39	-18
	48	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**							
m	10	4	-2	-11	0	5	5	9	20							
n	44	58	71	94	0	45	49	45	32							

La Tabla P ctxIdx antes citada para el elemento de sintaxis significant_coeff_flag (P) puede venir entonces dada por:

Variables de inicialización	significant_coeff_flag ctxIdx															
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n	62	57	54	51	57	55	54	55	51	60	54	47	42	39	47	43
	16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
m	0	0	0	0	0	0	0	-3	2	3	0	-3	-9	-4	4	1
n	41	42	41	41	39	78	66	68	31	53	65	74	93	20	44	57
	32	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**
m	0	-1	0	0	0	0	0	0	0	0	0	0	28	16	11	26
n	65	72	61	51	43	34	43	48	55	37	27	21	29	35	39	-18
	48	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**							
m	10	4	-2	-11	0	5	5	9	20							
n	44	58	71	94	0	45	49	45	32							

Volviendo a la ejecución del software algorítmico, una constante adicional puede ser definida como el máximo número de particiones en cualquier conjunto 4x4:

constante UInt NUM_SIG_FLAG_CTX_4x4 = 9;

Usando esta constante y las constantes del conjunto de particiones definidas anteriormente, se pueden mostrar las modificaciones a la función TComTrQuant :: getSigCtxInc como:

```
getSigCtxInc(pcCoeff, uiPosX, uiPosY, uiLog2BlkSize, uiStride, eTType) {
    eTType = eTType == TEXT_LUMA ? TEXT_LUMA : eTType == TEXT_NONE ?
                TEXT_NONE : TEXT_CHROMA
    L_C = eTType != TEXT_LUMA
    uiScanIdx = ((uiLog2BlkSize - 2) << 1) | L_C
    if (uiLog2BlkSize == 2) {
        return CTX_IND_MAP_4x4[L_C][(uiPosY << 2) + uiPosX]
    }
    if (uiLog2BlkSize == 3) {
        return NUM_SIG_FLAG_CTX_4x4 + CTX_IND_MAP_8x8[L_C] [(uiPosY << 3) + uiPosX]
    }
    // The rest of the function is unchanged
}
```

Se hace referencia ahora a la figura 9, que muestra un diagrama de bloques simplificado de una realización de ejemplo de un codificador 900. El codificador 900 incluye un procesador 902, una memoria 904 y una aplicación 906

de codificación. La aplicación 906 de codificación puede incluir un programa de ordenador o una aplicación almacenada en la memoria 904 y que contiene instrucciones para configurar el procesador 902 para realizar los pasos u operaciones tales como los descritos en el presente documento. Por ejemplo, la aplicación 906 de codificación puede codificar y sacar flujos de bits codificados de acuerdo con el procedimiento de nivel de reconstrucción adaptativa descrito en el presente documento. Los puntos para entrada de datos pueden ser relativos a audio, imágenes, vídeo o cualesquiera otros datos que puedan ser objeto de un esquema de compresión con pérdida de datos. La aplicación 906 de codificación puede incluir un módulo 908 de cuantificación configurado para determinar un nivel de reconstrucción adaptativa para cada índice de una estructura de partición. La aplicación 906 de codificación puede incluir un codificador entrópico configurado para codificar entrópicamente los niveles de reconstrucción adaptativa o datos RSP y otros datos. Se comprenderá que la aplicación 906 de codificación puede ser almacenada en un medio interpretable por ordenador, tal como un disco compacto, un dispositivo de memoria flash, una memoria de acceso aleatorio, un disco duro, etc.

Se hace también ahora referencia a la figura 10, que muestra un diagrama de bloques simplificado de una realización de ejemplo de un descodificador 1000. El descodificador 1000 incluye un procesador 1002, una memoria 1004 y una aplicación 1006 de descodificación. La aplicación 1006 de descodificación puede incluir un programa de ordenador o aplicación almacenada en la memoria 1004 y que contiene instrucciones para configurar el procesador 1002 para realizar los pasos u operaciones tales como los descritos en el presente documento. La aplicación 1006 de descodificación puede incluir un descodificador entrópico y un módulo 1010 de des cuantificación configurado para obtener los datos RSP o niveles de reconstrucción adaptativa y utilizar esos datos obtenidos para reconstruir los coeficientes de dominio transformado o dichos otros puntos de datos. Se comprenderá que la aplicación 1006 de descodificación puede ser almacenada en un medio interpretable 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 descodificador y/o el codificador según la presente aplicación pueden ejecutarse en diversos dispositivos de computación, incluyendo, sin limitación, servidores, ordenadores de uso general programados adecuadamente, codificación de audio/vídeo y dispositivos de reproducción, descodificadores de televisión, equipos de transmisión de televisión y dispositivos móviles. El descodificador o codificador pueden ser ejecutados por medio de software que contiene instrucciones para configurar un procesador para llevar a cabo las funciones descritas en el presente documento. Las instrucciones de software pueden almacenarse en cualquier memoria interpretable por ordenador adecuada no transitoria, incluyendo CDs, RAM, ROM, memoria flash, etc.

Se comprenderá que el codificador descrito en este documento y que el módulo, la rutina, el proceso, el paso u otro componente de software que lleva a cabo el método/procedimiento descrito para configurar el codificador se pueden realizar usando técnicas y lenguajes de programación normalizadas de ordenador. La presente solicitud no se limita a procesadores en particular, lenguajes de programación, convenciones de programación de ordenadores, estructuras de datos, otros detalles de ejecución. Los expertos en la técnica reconocerán que los procedimientos descritos pueden ser ejecutados como parte del código ejecutable por ordenador almacenado en memoria volátil o no volátil, como parte de un chip integrado de aplicación específica (ASIC), etc.

Se pueden realizar ciertas adaptaciones y modificaciones de las realizaciones descritas. Por lo tanto, las realizaciones anteriormente descritas se consideran como ilustrativas y no restrictivas.

REIVINDICACIONES

1. Un método de descodificar un flujo de bits de datos codificados para reconstruir un mapa significativo para una unidad de transformación, comprendiendo el método:

para cada posición del bit en el mapa significativo,
determinar un contexto para esa posición del bit basándose en un conjunto de particiones,
descodificar los datos codificados basándose en el contexto determinado para reconstruir un valor del bit, en el que cada valor del bit es un señalizador que indica si una posición correspondiente en la unidad de transformación contiene un coeficiente distinto de cero, y
actualizar el contexto basándose en ese valor del bit reconstruido,
en el que los valores del bit reconstruido forman el mapa significativo descodificado, y en el que el conjunto de particiones asigna contextos a las posiciones del bit,
caracterizado porque
la unidad de transformación tiene un tamaño de 4x4 y porque el conjunto de particiones asigna contextos a las posiciones del bit de acuerdo con un mapeo basado en bloques dado por:

0, 1, 2, 3,
4, 5, 2, 3,
6, 6, 7, 7,
8, 8, 7,

y en el que los números enteros anteriores representan los contextos asignados a las posiciones del bit de un mapa significativo de bloques de 4x4.

2. El método reivindicado de acuerdo con la reivindicación 1, en el que la determinación incluye seleccionar el conjunto de particiones de entre una diversidad de conjuntos de particiones basándose en el tipo de texto, y en el que el tipo de texto es de luminancia.

3. Un método de descodificar un flujo de bits de datos codificados para reconstruir un mapa significativo para una unidad de transformación, comprendiendo el método:

para cada posición del bit en el mapa significativo,
determinar un contexto para esa posición del bit basándose en un conjunto de particiones,
descodificar los datos codificados basándose en el contexto determinado para reconstruir un valor del bit, en el que cada valor del bit es un indicador que señaliza si una posición correspondiente en la unidad de transformación contiene un coeficiente distinto de cero, y
actualizar el contexto basándose en que el valor del bit reconstruido,
en el que los valores del bit reconstruido forman el mapa significativo descodificado, y en el que el conjunto de particiones asigna contextos a las posiciones del bit,
caracterizado porque
la unidad de transformación tiene un tamaño de 4x4 y porque el conjunto de particiones asigna contextos a las posiciones del bit de acuerdo con un mapeo basado en bloques dado por:

0, 1 2, 3,
1, 1, 2, 3,
4, 4, 5, 5,
3, 3, 5

y en el que los números enteros anteriores representan los contextos asignados a las posiciones del bit de un mapa significativo de bloque 4x4.

4. Un descodificador para descodificar un flujo de bits de datos codificados para reconstruir un mapa significativo para una unidad de transformación, comprendiendo el descodificador:

un procesador;
una memoria; y
una aplicación de descodificación almacenada en la memoria y que contiene instrucciones para configurar la realización del método reivindicado en cualquiera de las reivindicaciones 1 a 3.

5. Un método para codificar un mapa significativo para una unidad de transformación, comprendiendo el método:

para cada posición del bit en el mapa significativo,
determinar un contexto para esa posición del bit basándose en un conjunto de particiones,

codificar un valor del bit en esa posición del bit basándose en el contexto determinado para generar datos codificados, en el que cada valor del bit es un indicador que señaliza si una posición correspondiente en la unidad de transformación contiene un coeficiente distinto de cero, y
actualizar el contexto basándose en ese valor del bit,
en el que los datos codificados forman un mapa significativo codificado, y en el que el conjunto de particiones asigna contextos a las posiciones del bit,
caracterizado porque
la unidad de transformación tiene un tamaño de 4x4 y porque el conjunto de particiones asigna contextos a las posiciones del bit de acuerdo con un mapeo basado en bloques dado por:

0, 1, 2, 3,
4, 5, 2, 3,
6, 6 7, 7,
8, 8, 7,

y en el que los números enteros anteriores representan los contextos asignados a las posiciones del bit de un mapa significativo de bloque 4x4.

6. Un método para codificar un mapa significativo para una unidad de transformación, comprendiendo el método:

para cada posición del bit en el mapa significativo,
determinar un contexto para esa posición del bit basándose en un conjunto de particiones,
codificar un valor del bit en esa posición del bit basándose en el contexto determinado para generar datos codificados, en el que cada valor del bit es un indicador que señaliza si una posición correspondiente en la unidad de transformación contiene un coeficiente distinto de cero, y
actualizar el contexto basándose en ese valor del bit,
en el que los datos codificados forman un mapa significativo codificado, y en el que el conjunto de particiones asigna contextos a las posiciones del bit,
caracterizado porque
la unidad de transformación tiene un tamaño de 4x4 y porque el conjunto de particiones asigna contextos a las posiciones del bit de acuerdo con un mapeo basado en bloques dado por:

0, 1, 2, 3,
1, 1, 2, 3,
4, 4, 5, 5,
3, 3, 5

y en el que los números enteros anteriores representan los contextos asignados a las posiciones del bit de un mapa significativo de bloques de 4x4.

7. Un codificador para codificar un mapa significativo para una unidad de transformación, comprendiendo el codificador:

un procesador;
una memoria que almacena el mapa significativo; y
una aplicación de codificación almacenada en la memoria y que contiene instrucciones para configurar el procesador para llevar a cabo el método de acuerdo con la reivindicación 5 o con la reivindicación 6.

8. Un medio no transitorio interpretable por ordenador para almacenar instrucciones ejecutables por el procesador que, cuando se ejecuta, configura uno o más procesadores para realizar el método reivindicado en una cualquiera de las reivindicaciones 1 a 3, 5, y 6.

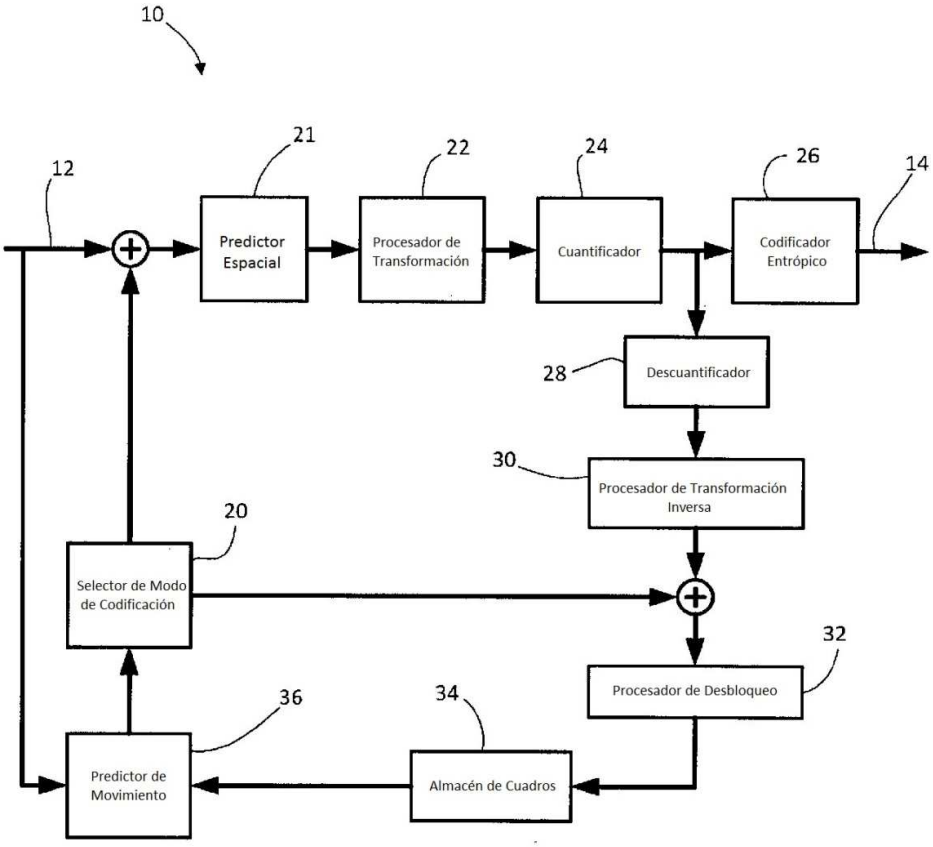
10
12
21
Predictor Espacial
22
Procesador de Transformación
24
Cuantificador
26
Codificador Entrópico
14
28
Descuantificador
30
Procesador de Transformación Inversa
20
Selector de Modo de Codificación
32
Procesador de Desbloqueo
36
Predictor de Movimiento
34
Almacén de Cuadros

FIG. 1

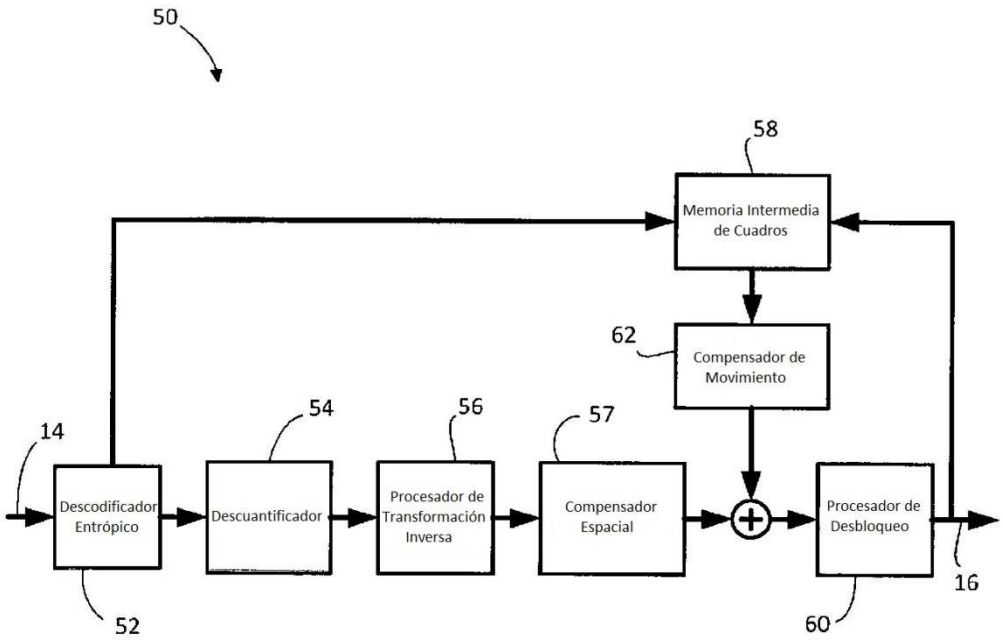
50
58
Memoria Intermedia de Cuadros
62
Compensador de Movimiento
14
54
56
57
Descodificador Entrópico
Descuantificador
Procesador de Transformación Inversa
Compensador Espacial
Procesador de Desbloqueo
16
52
60

FIG. 2

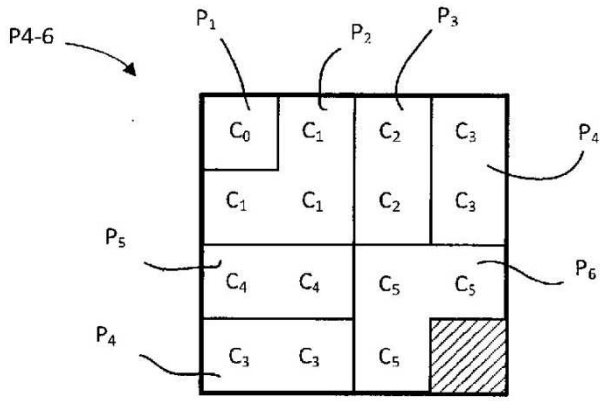
P4-6
P1
P2
P3
P4
P5
P6
P4
C0
C1
C2
C3
C1
C1
C2
C3
C4
C4
C5
C5
C3
C3
C5

FIG. 3

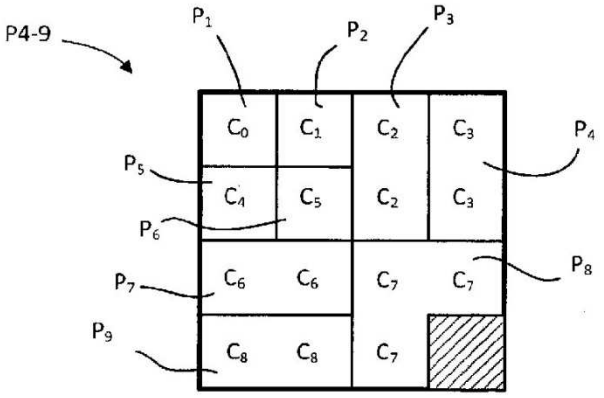
P4-9
P1
P2
P3
P4
P5
P6
P7
P8
P9
C0
C1
C2
C3
C4
C5
C2
C3
C6
C6
C7
C7
C8
C8
C7

FIG. 4

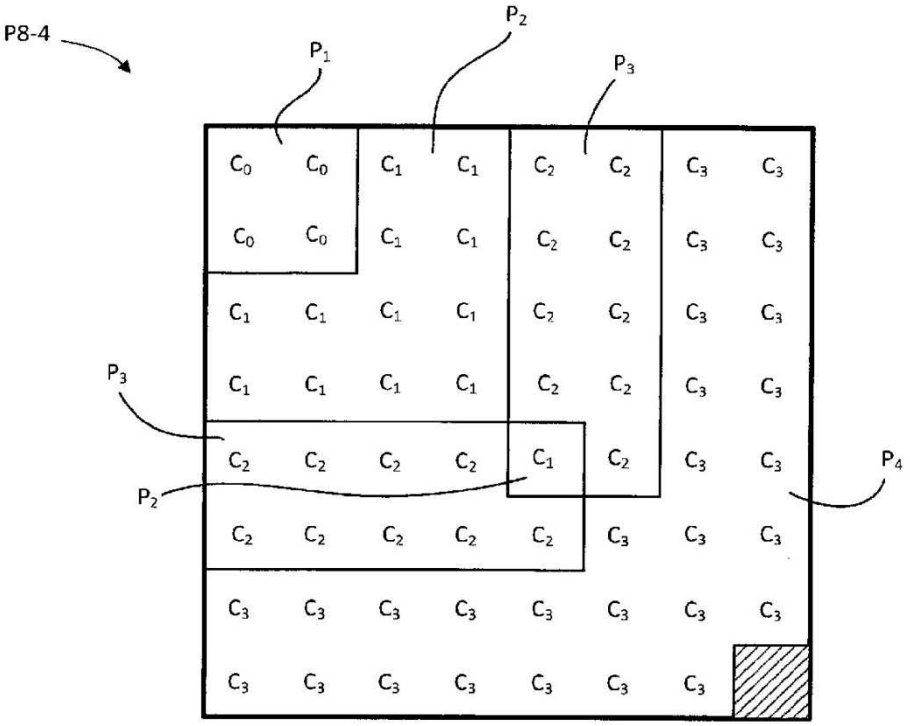
P8-4
P1
P2
P3
P3
P2
P4
C0 C0 C1 C1 C2 C2 C3 C3
C0 C0 C1 C1 C2 C2 C3 C3
C1 C1 C1 C1 C2 C2 C3 C3
C1 C1 C1 C1 C2 C2 C3 C3
C2 C2 C2 C2 C1 C2 C3 C3
C2 C2 C2 C2 C2 C3 C3 C3
C3 C3 C3 C3 C3 C3 C3 C3
C3 C3 C3 C3 C3 C3 C3

FIG. 5

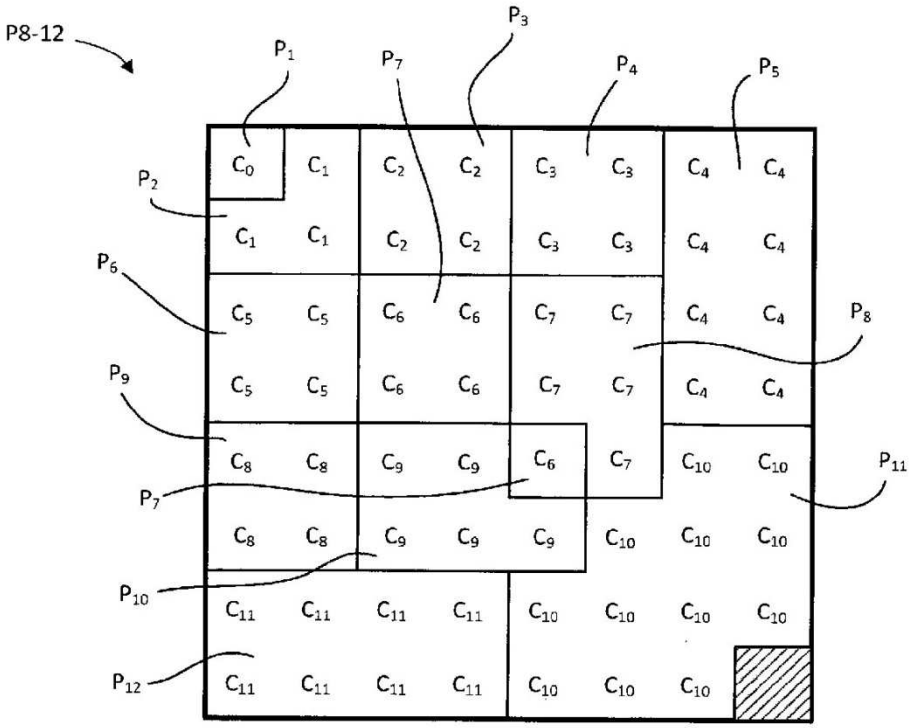
P8-12
P1
P7
P3
P4
P5
P2
P6
P8
P9
P7
P11
P10
P12
C0 C1 C2 C2 C3 C3 C4 C4
C1 C1 C2 C2 C3 C3 C4 C4
C5 C5 C6 C6 C7 C7 C4 C4
C5 C5 C6 C6 C7 C7 C4 C4
C8 C8 C9 C9 C6 C7 C10 C10
C8 C8 C9 C9 C9 C10 C10 C10
C11 C11 C11 C11 C10 C10 C10 C10
C11 C11 C11 C11 C10 C10 C10

FIG. 6

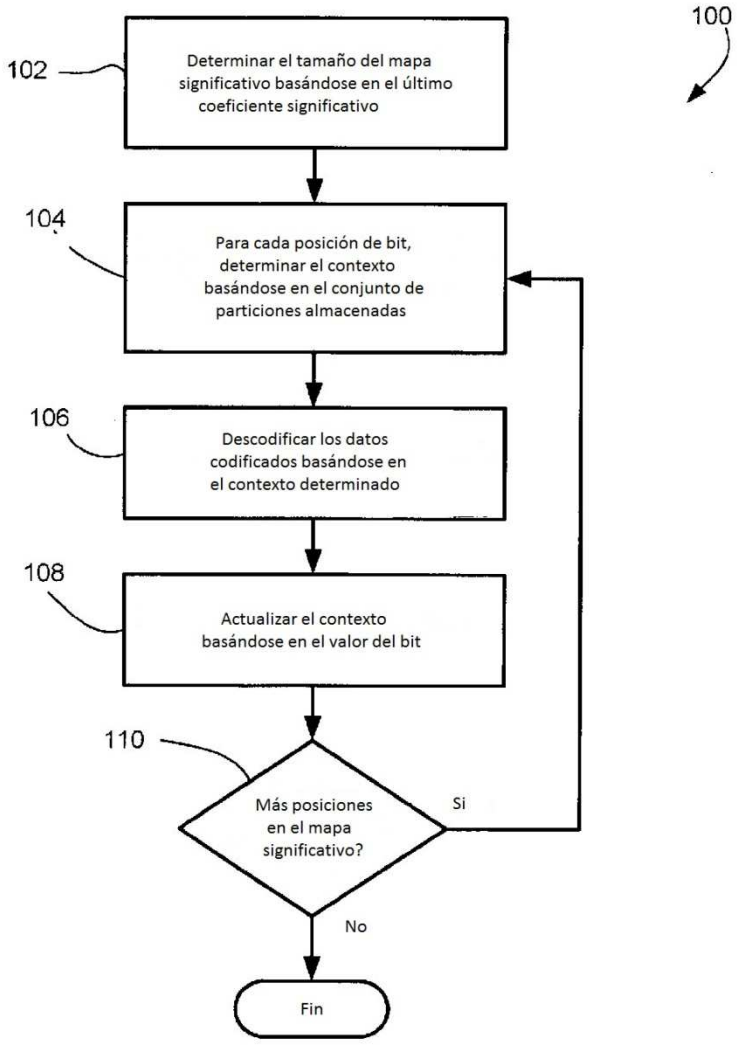
100
102
Determinar el tamaño del mapa significativo basándose en el último coeficiente significativo
104
Para cada posición de bit, determinar el contexto basándose en el conjunto de particiones almacenadas
106
Descodificar los datos codificados basándose en el contexto determinado
108
Actualizar el contexto basándose en el valor del bit
110
Más posiciones en el mapa significativo?
Si
No
Fin

FIG. 7

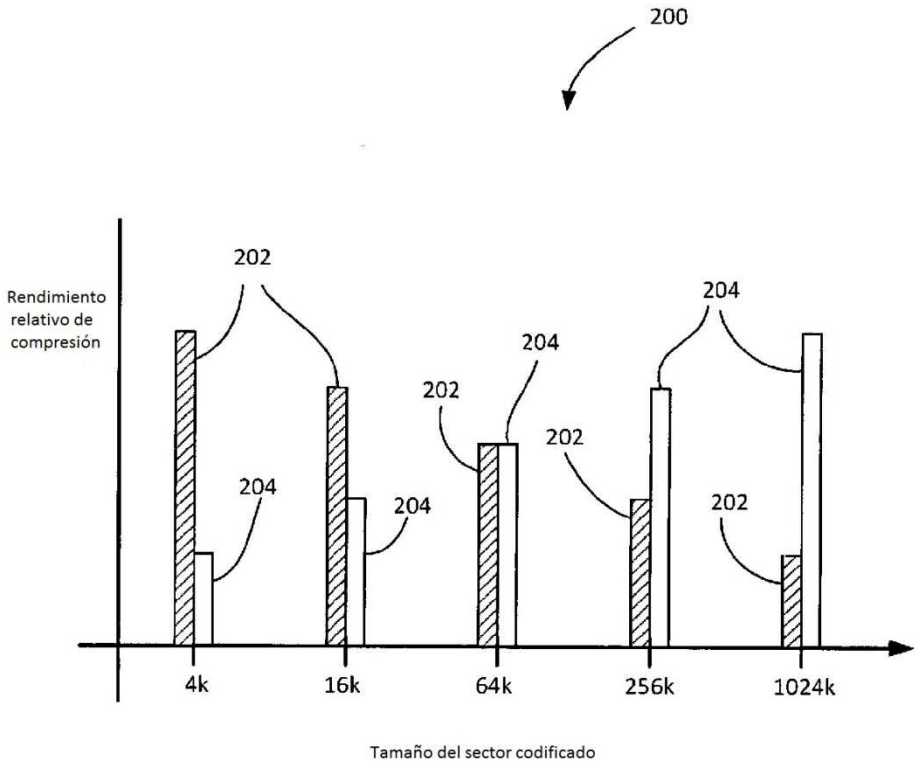
200
Rendimiento relativo de compresión
202
204
202
204
202
204
202
204
4k
16k
64k
256k
1024k
Tamaño del sector codificado

FIG. 8

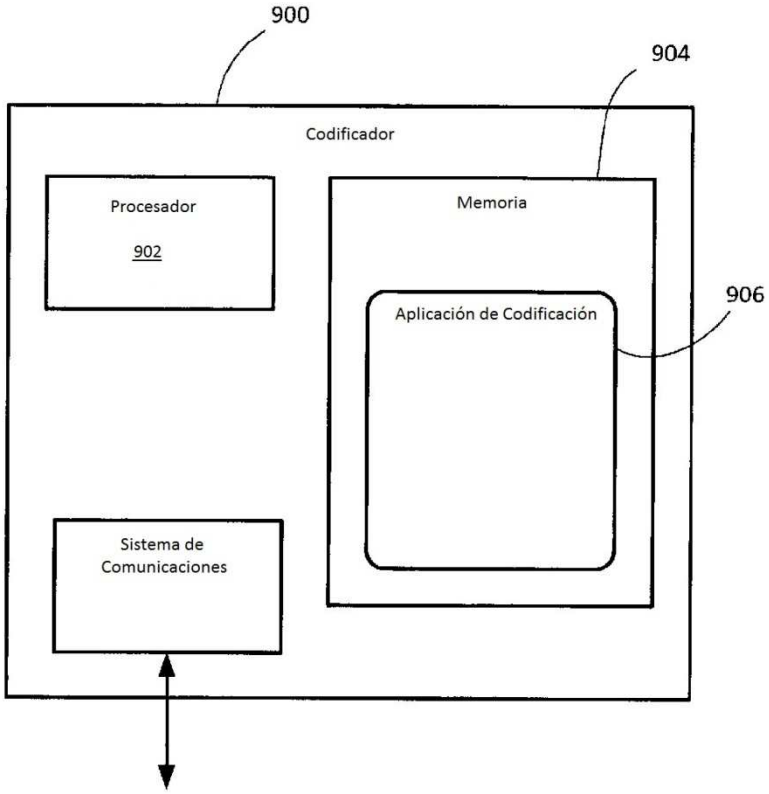
900
904
Codificador
Procesador
902
Memoria
906
Aplicación de Codificación
Sistema de
Comunicaciones

FIG. 9

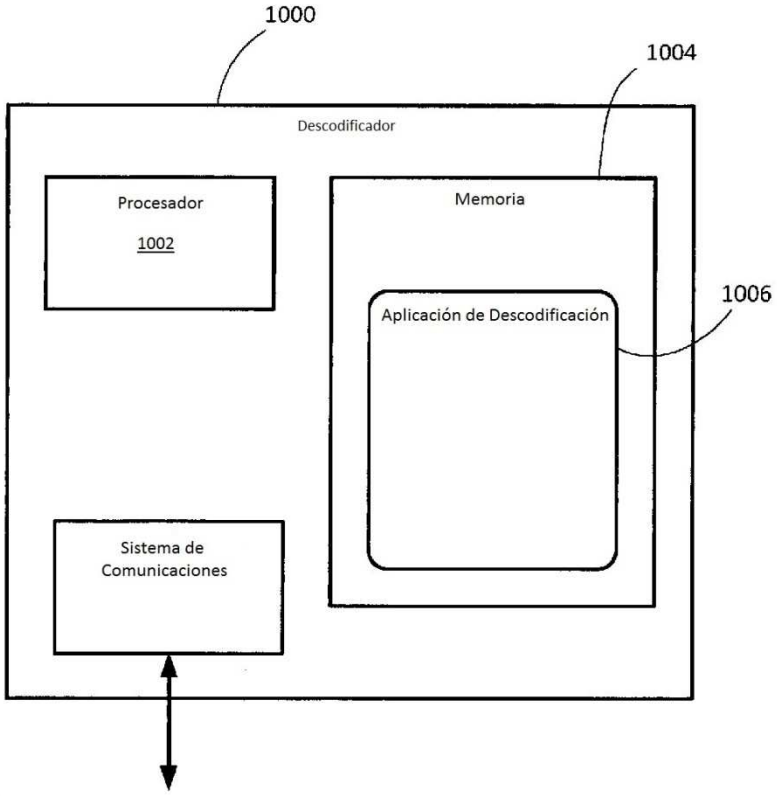
1000
1004
Descodificador
Procesador
1002
Memoria
1006
Aplicación de Descodificación
Sistema de
Comunicaciones

FIG. 10